<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Record Detail Page Actions</td>
<td>74</td>
</tr>
<tr>
<td>Get Record Edit Page Actions</td>
<td>76</td>
</tr>
<tr>
<td>Get Related List Actions</td>
<td>78</td>
</tr>
<tr>
<td>Get Related List Record Actions</td>
<td>80</td>
</tr>
<tr>
<td>Get List View Header Actions</td>
<td>82</td>
</tr>
<tr>
<td>Get List View Record Actions</td>
<td>85</td>
</tr>
<tr>
<td>Get List View Chart Actions</td>
<td>88</td>
</tr>
<tr>
<td>Get Lightning Page Actions</td>
<td>90</td>
</tr>
<tr>
<td>Get Lookup Field Actions</td>
<td>92</td>
</tr>
<tr>
<td>Get MRU List View Actions</td>
<td>94</td>
</tr>
<tr>
<td>Get Photo Actions</td>
<td>96</td>
</tr>
<tr>
<td>Favorites</td>
<td>98</td>
</tr>
<tr>
<td>Create a Favorite</td>
<td>99</td>
</tr>
<tr>
<td>Get a Favorite</td>
<td>100</td>
</tr>
<tr>
<td>Get Favorites</td>
<td>101</td>
</tr>
<tr>
<td>Update a Favorite</td>
<td>101</td>
</tr>
<tr>
<td>Update a Batch of Favorites</td>
<td>103</td>
</tr>
<tr>
<td>Delete a Favorite</td>
<td>103</td>
</tr>
<tr>
<td>Update Usage of a Favorite</td>
<td>104</td>
</tr>
<tr>
<td>List Views</td>
<td>104</td>
</tr>
<tr>
<td>Get List Views for an Object</td>
<td>105</td>
</tr>
<tr>
<td>Get List View Records</td>
<td>106</td>
</tr>
<tr>
<td>Get List View Metadata</td>
<td>110</td>
</tr>
<tr>
<td>Get List View Records and Metadata</td>
<td>112</td>
</tr>
<tr>
<td>Get Most Recently Used List View Records</td>
<td>116</td>
</tr>
<tr>
<td>Get Most Recently Used List View Metadata</td>
<td>119</td>
</tr>
<tr>
<td>Get Most Recently Used List View Records and Metadata</td>
<td>120</td>
</tr>
<tr>
<td>Apps</td>
<td>124</td>
</tr>
<tr>
<td>Get Apps</td>
<td>124</td>
</tr>
<tr>
<td>Get an App</td>
<td>127</td>
</tr>
<tr>
<td>Update Last Selected App</td>
<td>129</td>
</tr>
<tr>
<td>Get Last Selected App</td>
<td>131</td>
</tr>
<tr>
<td>Get Personalized Navigation Items</td>
<td>133</td>
</tr>
<tr>
<td>Get All Navigation Items</td>
<td>135</td>
</tr>
<tr>
<td>Update Personalized Navigation Items</td>
<td>137</td>
</tr>
<tr>
<td>Get Active Theme</td>
<td>140</td>
</tr>
<tr>
<td>Chapter 4: User Interface API Request Bodies</td>
<td>141</td>
</tr>
<tr>
<td>Favorite Batch Collection Input</td>
<td>141</td>
</tr>
<tr>
<td>Favorite Batch Input</td>
<td>142</td>
</tr>
<tr>
<td>Favorite Input</td>
<td>142</td>
</tr>
<tr>
<td>Navigation Item Input</td>
<td>143</td>
</tr>
<tr>
<td>Page Reference Input</td>
<td>144</td>
</tr>
<tr>
<td>Record Input</td>
<td>145</td>
</tr>
</tbody>
</table>
Chapter 5: User Interface API Response Bodies

User Interface API Top-Level Response Bodies
- Action ......................................................... 149
- App ........................................................... 150
- Apps .......................................................... 151
- Batch Results .............................................. 152
- Favorite ...................................................... 152
- Favorite Collection ....................................... 153
- List Metadata ............................................... 153
- List Record Collection .................................. 154
- List UI ......................................................... 155
- List View Summary Collection ....................... 155
- Lookup Values ............................................. 155
- Most Recently Used List Metadata .................. 156
- Most Recently Used List Record Collection ....... 157
- Most Recently Used List UI ............................ 157
- Navigation Items .......................................... 158
- Object Info .................................................. 158
- Object Info Directory .................................... 161
- Picklist Values ............................................. 161
- Picklist Values Collection .............................. 162
- Record ....................................................... 162
- Record Collection ........................................ 163
- Record Defaults .......................................... 164
- Record Layout ............................................ 164
- Record UI .................................................... 165
- Theme ....................................................... 166

User Interface API Nested Response Bodies
- Batch Result Item ........................................... 169
- Canvas Layout Component .............................. 170
- Child Relationship ........................................ 172
- Custom Link Layout Component ........................ 173
- Duplicate Record Error .................................. 174
- Error Message ............................................ 174
- Error with Output ....................................... 174
- Field ......................................................... 175
- Field Layout Component ................................ 179
- Field Value ............................................... 179
- Filtered Lookup Info ..................................... 180
- Lead Status Picklist Value Attributes ............... 180
- List Column ............................................... 181
- List Filter by Info ........................................ 181
- List Order by Info ........................................ 182
- List Reference ............................................. 182
Build Salesforce UI for native mobile apps and custom web apps using the same API that Salesforce uses to build Lightning Experience and Salesforce for Android, iOS, and mobile web. Build user interfaces that let users work with records, list views, actions, favorites, and more. Not only do you get data and metadata in a single response, but the response matches metadata changes made to the org by Salesforce admins. You don’t have to worry about layouts, picklists, field-level security, or sharing—all you have to do is build an app that users love.

The User Interface API base URL is

https://{your_instance}.salesforce.com/services/data/v{api_version}/ui-api.

The Record UI resource shows the power of User Interface API. To display a record, your code makes this simple request:

```bash
GET /ui-api/record-ui/001R0000003GeJ1IAK
```

Behind the scenes, Salesforce does the heavy lifting:

- Checks field-level security settings, sharing settings, and perms.
- Makes SOQL queries to get record data.
- Gets object metadata and theme information.
- Gets layout information.

You are one happy developer, because Salesforce transforms the results into an easy-to-consume JSON response with all the metadata and data necessary to display the record.

In addition to building UI for Salesforce records, you can also use User Interface API to build UI for Salesforce actions, favorites, and list views.

User Interface API Quick Start
The best way to learn an API is to use it. Let’s go!

User Interface API Sample Apps
We’ve built two sample apps that use User Interface API to create, read, update, and delete records. The apps are open-source and available on GitHub.

Supported Objects
User Interface API supports all custom objects and many standard objects.

Authentication, Versioning, Limits, ETag, and More
Learn the high-level facts about how to use User Interface API. How do you authenticate? How is the API versioned? What are the rate limits? All the information you need, available at a glance.

Response Body Encoding
Responses are minimally HTML entity-encoded by default.

Use CORS to Access Salesforce Resources from Web Browsers
Cross-Origin Resource Sharing (CORS) enables web browsers to request resources from origins other than their own (cross-origin). For example, using CORS, JavaScript code at https://www.example.com could request a resource from https://www.salesforce.com. To access supported Salesforce APIs, Apex REST resources, and Lightning Out from JavaScript code in a web browser, add the origin serving the code to a Salesforce CORS whitelist.
Status Codes and Error Responses
The response header contains an HTTP status code. If the response isn't successful, the response body contains an error message, and if appropriate, information about the field or object where the error occurred.

User Interface API Quick Start

The best way to learn an API is to use it. Let's go!

Sign up for a Salesforce Developer Edition (also known as a DE org).

This quick start makes a request to the Record UI resource, /ui-api/record-ui/{recordIds}. This resource returns all the information necessary to display a record in an app: layout metadata, object metadata, and record data.

1. Go to the Workbench API utility.
2. Log in to your DE org.
3. Select utilities > Rest Explorer.
4. Make a GET request to /ui-api/object-info/.
   The response is a directory of all the objects that User Interface API supports and the context user has access to. Grab an object apiName from the response to use in the next request. This example uses the Account object.
5. Make a GET request to /query?q=SELECT Id FROM {objectApiName}.
   Here's an example request:
   /query?q=SELECT Id FROM Account
   The Record UI resource requires a record Id. Grab one from the response to use in the next request.
6. Make a GET request to /ui-api/record-ui/{recordIds}. Substitute the record ID from the previous step.
   Here's an example request:
   GET /ui-api/record-ui/001R0000003GeJ1IAK
   Use request parameters to specify the form factor (Large, Medium, Small), layout type (Compact, Full), and access mode (Create, Edit, View) of the record (or records). This example uses the default values, which are Large form factor, Full layout, and View mode.

   The response includes nested response bodies. Each response body is a map because you can pass a list of record IDs to the request (for example, /ui-api/record-ui/001R0000003GeJ1IAK, 001R0000003GKR0IAO). In our example, we passed only 1 record ID.

   - layoutUserStates—A map of layout user state IDs to user state information for each layout section
   - layouts—A map of object API names to user type layout information for each object
   - objectInfos—A map of object API names to metadata for each object
   - records—A map of record IDs to data for each record

   **Tip:** To get these response bodies individually, use GET /ui-api/layout/{objectApiName}, GET /ui-api/object-info/{objectApiName}, and GET /ui-api/records/{recordId}.

   ```json
   {
     "layoutUserStates": {
       "00hR0000000NurZIAS": {
         "id": "00hR0000000NurZIAS",
         "sectionUserStates": {
   ```
"01BR0000000UjAhMAK":{
  "collapsed":false,
  "id":"01BR0000000UjAhMAK"
},
"01BR0000000UjAmMAK":{
  "collapsed":false,
  "id":"01BR0000000UjAmMAK"
}

"layouts":{
  "Account":{
    "012000000000000AAA":{
      "Full":{
        "View":{
          "id":"00hR0000000NurZIAS",
          "layoutType":"Full",
          "mode":"View",
          "sections":[
            {
              "collapsible":false,
              "columns":2,
              "heading":"Account Information",
              "id":"01BR0000000UjAhMAK",
              "layoutRows":[
                {
                  "layoutItems":[
                    {
                      "editableForNew":false,
                      "editableForUpdate":false,
                      "label":"Account Owner",
                      "layoutComponents":[
                        {
                          "apiName":"OwnerId",
                          "componentType":"Field",
                          "label":"Owner ID"
                        }
                      ],
                      "lookupIdApiName":"OwnerId",
                      "required":false,
                      "sortable":false
                    },
                    {
                      "editableForNew":true,
                      "editableForUpdate":true,
                      "label":"Phone",
                      "layoutComponents":[
                        {
                          "apiName":"Phone",
                          "componentType":"Field",
                          "label":"Account Phone"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      }
    }
  }
}
"required":false,
"sortable":false
},
{
"editableForNew":true,
"editableForUpdate":true,
"label":"Website",
"layoutComponents":[
{
"apiName":"Website",
"componentType":"Field",
"label":"Website"
}
],
"lookupIdApiName":null,
"required":false,
"sortable":false
}
}
"rows":3,
"useHeading":false
},
... sections removed for space ...
{
"collapsible":true,
"columns":3,
"heading":"Custom Links",
"id":"01BR0000000UjAmMAK",
"layoutRows":{
"layoutItems":{
"editableForNew":false,
"editableForUpdate":false,
"label":"
"layoutComponents":{

"apiName":"GoogleSearch",
"behavior":"NewWindow",
"componentType":"CustomLink",
"customLinkUrl":"/servlet/servlet.Integration?lid=00bR0000000ICHb&eid=ENTITY_ID&ic=1",
"label":"Google Search"
}
],
"lookupIdApiName":null,
"required":false,
"sortable":false
}
Get Started with User Interface API
Get Started with User Interface API

User Interface API Quick Start

"rows": 2,
  "useHeading": true
}
]
]
}
}
"objectInfos": {
  "Account": {
    "apiName": "Account",
    "childRelationships": [
      {
        "childObjectApiName": "Account",
        "fieldName": "ParentId",
        "junctionIdListNames": [
        ],
        "junctionReferenceTo": [
        ],
        "relationshipName": "ChildAccounts"
      },
      ... some child relationships removed for space ...
      {
        "childObjectApiName": "WorkOrder",
        "fieldName": "AccountId",
        "junctionIdListNames": [
        ],
        "junctionReferenceTo": [
        ],
        "relationshipName": "WorkOrders"
      }
    ],
    "createable": true,
    "custom": false,
    "defaultRecordTypeId": null,
    "deleteable": true,
    "feedEnabled": true,
    "fields": {
      "AccountSource": {
        "apiName": "AccountSource",
        "calculated": false,
        "compound": false,
        "compoundComponentName": null,
        "compoundFieldName": null,
        "controllerName": null,
        "createable": true,
        "custom": false,
        "dataType": "Picklist",
        "extraTypeInfo": null,
        "extraTypeInfo": null
      }
    }
  }
}
Get Started with User Interface API

"filterable":true,
"filteredLookupInfo":null,
"highScaleNumber":false,
"htmlFormatted":false,
"inlineHelpText":null,
"label":"Account Source",
"length":40,
"nameField":false,
"polymorphicForeignKey":false,
"precision":0,
"reference":false,
"referenceTargetField":null,
"referenceToInfos":[
  
],
"relationshipName":null,
"required":false,
"scale":0,
"searchPrefilterable":false,
"sortable":true,
"unique":false,
"updateable":true
],

... some field info removed for space ...

"Website":{
  "apiName":"Website",
  "calculated":false,
  "compound":false,
  "compoundComponentName":null,
  "compoundFieldName":null,
  "controllerName":null,
  "createable":true,
  "custom":false,
  "dataType":"Url",
  "extraTypeInfo":null,
  "filterable":true,
  "filteredLookupInfo":null,
  "highScaleNumber":false,
  "htmlFormatted":false,
  "inlineHelpText":null,
  "label":"Website",
  "length":255,
  "nameField":false,
  "polymorphicForeignKey":false,
  "precision":0,
  "reference":false,
  "referenceTargetField":null,
  "referenceToInfos":[
  
],
  "relationshipName":null,
  "required":false,
  "scale":0,
  "searchPrefilterable":false,
"sortable":true,
"unique":false,
"updateable":true
],
"keyPrefix":"001",
"label":"Account",
"labelPlural":"Accounts",
"layoutable":true,
"mruEnabled":true,
"nameFields":[
  "Name"
],
"queryable":true,
"recordTypeInfos":{
  "012000000000000AAA":{
    "available":true,
    "defaultRecordTypeMapping":true,
    "master":true,
    "name":"Master",
    "recordTypeId":"012000000000000AAA"
  }
},
"searchable":true,
"themeInfo":{
  "color":"7F8DE1",
  "iconUrl":"https://na7.salesforce.com/img/icon/t4v35/standard/account_120.png"
},
"updateable":true
},
"User":{
  "apiName":"User",
  "childRelationships":[
  {
    "childObjectApiName":"AcceptedEventRelation",
    "fieldName":"RelationId",
    "junctionIdListNames":[
      
    ],
    "junctionReferenceTo":[
      
    ],
    "relationshipName":"AcceptedEventRelations"
  },
  ... some child relationships removed for space ...
  {
    "childObjectApiName":"WorkThanks",
    "fieldName":"GiverId",
    "junctionIdListNames":[
      
    ],
    "junctionReferenceTo":[
      
    ]}
Get Started with User Interface API

User Interface API Quick Start

... fields removed for space ...

"Username":{
  "apiName":"Username",
  "calculated":false,
  "compound":false,
  "compoundComponentName":null,
  "compoundFieldName":null,
  "controllerName":null,
  "createable":true,
  "custom":false,
  "dataType":null,
  "extraTypeInfo":null,
  "filterable":false,
  "filteredLookupInfo":null,
  "highScaleNumber":false,
  "htmlFormatted":false,
  "inlineHelpText":false,
  "label":"Username",
  "length":1000,
  "nameField":false,
  "polymorphicForeignKey":false,
  "precision":0,
  "reference":false,
  "referenceTargetField":null,
  "referenceToInfos":[]
},

"relationshipName":null,
"required":false,
"scale":0,
"searchPrefilterable":false,
"sortable":true,
"unique":false,
"updateable":true
},

... fields removed for space ...
Get Started with User Interface API
User Interface API Sample Apps

We’ve built two sample apps that use User Interface API to create, read, update, and delete records. The apps are open-source and available on GitHub.

Record Viewer is a Node.js app available at https://github.com/forcedotcom/RecordViewer.

Record Viewer Native is a React Native app available at https://github.com/forcedotcom/RecordViewerNative.

Supported Objects

User Interface API supports all custom objects and many standard objects.

To get a directory of objects that are supported by the API and accessible to the context user, use the /ui-api/object-info resource.

To understand how the API operates on supported objects, let’s look at some examples.

This resource gets object metadata. It operates on any supported object.

GET /ui-api/object-info/{objectApiName}

This example gets Account metadata.

GET /ui-api/object-info/Account

This resource gets record data and metadata. It operates on any supported object record.

GET /ui-api/record-ui/{recordIds}

This example gets data and metadata to display an Opportunity record.

GET /ui-api/record-ui/006RM000002Xcd1YAG

Supported Objects

Most User Interface API resources support these objects. The list views and most recently used (MRU) list views resources support a smaller set of objects.

Supported objects:
Get Started with User Interface API

- Account
- AccountContactRelation
- AccountPartner
- AccountTeamMember
- ActionCadence
- Asset
- AssetRelationship
- AssignedResource
- AttachedContentNote
- BusinessAccount
- BusinessMilestone
- BusinessProfile
- Campaign
- CampaignMember
- CareBarrier
- CareBarrierType
- CareProgram
- CareProgramEnrollee
- Case
- Claim
- ClaimCase
- ClaimItem
- ClaimParticipant
- Contact
- ContactRequest
- ContentDocument
- ContentNote
- ContentVersion
- ContentWorkspace
- Contract
- ContractContactRole
- ContractLineItem
- CoverageType
- CustomObject
- CustomerProperty
- DataStreamDefinition
- ElectronicMediaGroup
- Entitlement
- EntityArchivingSetup
- EnvironmentHubMember
Get Started with User Interface API

- Image
- InsuranceClaimAsset
- InsurancePolicy
- InsurancePolicyAsset
- InsurancePolicyCoverage
- InsurancePolicyMemberAsset
- InsurancePolicyParticipant
- InsuranceProfile
- Lead
- LicensingRequest
- LoanApplicant
- LoanApplicationLiability
- MaintenanceAsset
- MaintenancePlan
- MarketingAction
- MarketingResource
- Note
- OperatingHours
- Opportunity
- OpportunityLineItem
- OpportunityPartner
- OpportunityTeamMember
- Order
- OrderItem
- OrderItemSummaryChange
- OrderSummary
- OrgMetric
- OrgMetricScanSummary
- OrgMetricScanResult
- Partner
- PersonAccount
- PersonLifeEvent
- PriceAdjustmentSchedule
- Pricebook2
- PricebookEntry
- Producer
- ProducerPolicyAssignment
- Product2
- ProductCoverage
- ProductMedia
Get Started with User Interface API

- ProductMedia
- Quote
- QuoteDocument
- QuoteLineItem
- RecordType
- ResourceAbsence
- ResourcePreference
- RetailVisitKpi
- RetailVisitWorkTask
- RetailVisitWorkTaskOrder
- SecuritiesHolding
- ServiceAppointment
- ServiceContract
- ServiceCrew
- ServiceCrewMember
- ServiceResource
- ServiceResourceCapacity
- ServiceResourceSkill
- Service Territory
- Service Territory Location
- Service Territory Member
- Shift
- Shipment
- SkillRequirement
- SocialPost
- Survey Invitation
- Survey Response
- Survey Subject
- Tenant
- TimeSheet
- TimeSheetEntry
- TimeSlot
- UsageEntitlement
- UsageEntitlementPeriod
- User
- Visit
- WebStoreSearchProdSettings
- WorkerCompCoverageClass
- WorkOrder
- WorkOrderLineItem
Supported Objects for List Views
The list views resources support a subset of objects.

List views resources:

- `/ui-api/list-ui/${objectApiName}`
- `/ui-api/list-info/${listViewId}` and
  `/ui-api/list-info/${objectApiName}/${listViewApiName}`
- `/ui-api/list-records/${listViewId}` and
  `/ui-api/list-records/${objectApiName}/${listViewApiName}`
- `/ui-api/list-ui/${listViewId}` and `/ui-api/list-ui/${objectApiName}/${listViewApiName}

Supported objects for list views resources:

- Account
- ActionCadence
- Asset
- BusinessMilestone
- BusinessProfile
- Campaign
- Case
- Claim
- ClaimCase
- ClaimItem
- ClaimParticipant
- Contact
- ContactRequest
- ContentVersion
- Contract
- CoverageType
- CustomerProperty
- Custom Object
- DataStreamDefinition
- Entitlement
- EntityMilestone
- EnvironmentHubMember
- Image
- InsuranceClaimAsset
- InsurancePolicy
- InsurancePolicyAsset
- InsurancePolicyCoverage
- InsurancePolicyMemberAsset
- InsurancePolicyParticipant
- InsuranceProfile
- Lead
- LoanApplicant
- LoanApplicationLiability
- MaintenancePlan
- OperatingHours
- Opportunity
- Order
- OrderSummary
- PersonLifeEvent
- Pricebook2
- Producer
- ProducerPolicyAssignment
- Product2
- ProductCoverage
- Quote
- ResourcePreference
- RetailVisitKpi
- RetailVisitWorkTask
- RetailVisitWorkTaskOrder
- ReturnOrder
- SecuritiesHolding
- ServiceAppointment
- ServiceContract
- ServiceCrew
- ServiceCrewMember
- ServiceResource
- ServiceResourceSkill
- ServiceTerritory
- ServiceTerritoryMember
- Shift
- Shipment
- SkillRequirement
- SocialPost
- SurveyInvitation
- SurveyResponse
- SurveySubject
- Tenant
- TimeSheet
- TimeSheetEntry
Supported Objects for MRU List Views

The MRU list views resources support a subset of objects.

MRU list views resources:

- `/ui-api/mru-list-info/${objectApiName}`
- `/ui-api/mru-list-records/${objectApiName}`
- `/ui-api/mru-list-ui/${objectApiName}`

Supported objects for MRU list views resources:

- Account
- ActionCadence
- Asset
- BusinessMilestone
- BusinessProfile
- Campaign
- Case
- Claim
- ClaimCase
- ClaimItem
- ClaimParticipant
- Contact
- ContactRequest
- Contract
- CoverageType
- CustomerProperty
- DataStreamDefinition
- Entitlement
- EnvironmentHubMember
- Image
- InsuranceClaimAsset
- InsurancePolicy
- InsurancePolicyAsset
- InsurancePolicyCoverage
- InsurancePolicyMemberAsset
- InsurancePolicyParticipant
- InsuranceProfile
- Lead
• LoanApplicant
• LoanApplicationLiability
• MaintenancePlan
• OperatingHours
• Opportunity
• Order
• OrderSummary
• PersonLifeEvent
• Pricebook2
• Producer
• ProducerPolicyAssignment
• Product2
• ProductCoverage
• Quote
• ResourcePreference
• RetailVisitKpi
• RetailVisitWorkTask
• RetailVisitWorkTaskOrder
• ReturnOrder
• Shift
• SecuritiesHolding
• ServiceAppointment
• ServiceResource
• ServiceTerritory
• ServiceTerritoryMember
• ServiceResourceSkill
• ServiceCrew
• ServiceCrewMember
• ServiceContract
• Shipment
• SkillRequirement
• SocialPost
• SurveyInvitation
• SurveyResponse
• SurveySubject
• TimeSheet
• TimeSheetEntry
• Visit
• WorkerCompCoverageClass
• WorkType
Authentication, Versioning, Limits, ETag, and More

Learn the high-level facts about how to use User Interface API. How do you authenticate? How is the API versioned? What are the rate limits? All the information you need, available at a glance.

Authentication

Like other Salesforce REST APIs, User Interface API uses OAuth 2.0. To specify OAuth settings, create a connected app in Salesforce. You can specify the api or full OAuth scopes.

Case Sensitivity

When writing a User Interface API client, assume case sensitivity. Some parts of the API aren’t case-sensitive, but some are, so it’s best to assume case sensitivity.

Communities User Access

Users with a Customer Community or Partner Community User License can access User Interface API.

Dates

Dates are represented in ISO 8601 format.

HTTP Caching with ETag

To reduce network traffic and increase your app’s response time, make conditional HTTP requests that take advantage of browser caching. If the browser cache holds the latest version of a representation, the web server returns only HTTP headers.

To take advantage of caching, User Interface API responses include an ETag header field. The ETag (or Entity Tag) contains a hash that identifies a representation: ETag: "b9a5cd4fbcf1b65b03d95a4d9ffa8f4--gzip". When a representation changes, the web server generates a new ETag.

GET /ui-api/object-info/account

HTTP/1.1 200 OK
Date: Tue, 07 Nov 2017 04:27:43 GMT
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: upgrade-insecure-requests
X-Robots-Tag: none
Cache-Control: no-cache,must-revalidate,max-age=0,no-store,private
Set-Cookie: BrowserId=GHrCS0saRl64DgVv-NrLjA;Path=/;Domain=.salesforce.com;Expires=Sat, 06-Jan-2018 04:27:45 GMT;Max-Age=5184000
Expires: Thu, 01 Jan 1970 00:00:00 GMT
ETag: "b9a5cd4fbcf1b65b03d95a4d9ffa8f4--gzip"
Content-Type: application/json;charset=UTF-8
Vary: Accept-Encoding
Content-Encoding: gzip
Transfer-Encoding: chunked

{
The client saves the ETag and sends it in a header in subsequent requests for the resource: If-None-Match: "b9a5cd4fbcf1b65b03d95a4d9ffa8f4". The web server compares this ETag value with the ETag value of the current version of the representation. If the values are the same, the browser cache holds the latest version and the web server returns an HTTP status of 304 Not Modified.

If-Modified-Since

PATCH requests support the If-Modified-Since HTTP header.

JSON Support

The JavaScript Object Notation (JSON) format is supported with UTF-8.

Limits

The User Interface API uses the Salesforce API limits. When you exceed the rate limit, all User Interface API resources return a 503 Service Unavailable error code.

Versioning

User Interface API is versioned. Specify the API version number in each request.

GET https://{your_salesforce_instance}.salesforce.com/services/data/v47.0/ui-api

SEE ALSO:

Digging Deeper into OAuth 2.0 on Salesforce

Response Body Encoding

Responses are minimally HTML entity-encoded by default.

These characters are escaped in all strings:

<table>
<thead>
<tr>
<th>Character</th>
<th>Escaped as</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>&lt;</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>&amp;</td>
<td>&amp;</td>
</tr>
</tbody>
</table>
URL values included in response payloads have special encoding. The main part of the URL is URL-encoded as per RFC2396, and the query string is HTML-form encoded. This encoding cannot be turned off.

⚠️ **Warning:** User-submitted content may not be filtered at input and can come from many sources including third-party mobile and web applications. Process API output for the context in which your app uses the output. Failing to process output can expose applications and users to Cross Site Scripting (XSS) exploits and other problems. These problems can result in data loss, application failure, and exposure of company information.

API output can be used in many contexts. Don’t assume that the default entity encoding is appropriate for all contexts. Using API output inside HTML attribute values, inside URLs, with JavaScript, inside `<script>` tags, and inside CSS all require different encoding and whitelisting. See the Open Web Application Security Project for information on how to handle API output in different contexts.

Clients can request raw (unencoded) output. Set the `X-Chatter-Entity-Encoding` HTTP header in a request to `false`.

# Use CORS to Access Salesforce Resources from Web Browsers

Cross-Origin Resource Sharing (CORS) enables web browsers to request resources from origins other than their own (cross-origin). For example, using CORS, JavaScript code at `https://www.example.com` could request a resource from `https://www.salesforce.com`. To access supported Salesforce APIs, Apex REST resources, and Lightning Out from JavaScript code in a web browser, add the origin serving the code to a Salesforce CORS whitelist.

These Salesforce technologies support CORS:

- Analytics REST API
- Bulk API
- Chatter REST API
- Salesforce IoT REST API
- Lightning Out
- REST API
- User Interface API
- Apex REST

In Salesforce, add the origin serving the code to a CORS whitelist. If a browser that supports CORS makes a request to an origin in the whitelist, Salesforce returns the origin in the `Access-Control-Allow-Origin` HTTP header, along with any additional CORS HTTP headers. If the origin is not included in the whitelist, Salesforce returns HTTP status code 403.

1. From Setup, enter CORS in the Quick Find box, then select CORS.
2. Select New.
3. Enter an origin URL pattern.

   The origin URL pattern must include the HTTPS protocol (unless you’re using your localhost) and a domain name and can include a port. The wildcard character (*) is supported and must be in front of a second-level domain name. For example, `https://*.example.com` adds all subdomains of `example.com` to the whitelist.

   The origin URL pattern can be an IP address. However, an IP address and a domain that resolve to the same address are not the same origin, and you must add them to the CORS whitelist as separate entries.

   The origin URL pattern might not match the URL that appears in the address bar in your browser. Make sure that you’re whitelisting the origin in the request header.
Important: CORS does not support requests for unauthenticated resources, including OAuth endpoints. You must pass an OAuth token with requests that require it.

Status Codes and Error Responses

The response header contains an HTTP status code. If the response isn’t successful, the response body contains an error message, and if appropriate, information about the field or object where the error occurred.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Success with GET, PATCH, or HEAD request</td>
</tr>
<tr>
<td>201</td>
<td>Success with POST request</td>
</tr>
<tr>
<td>204</td>
<td>Success with DELETE</td>
</tr>
<tr>
<td>400</td>
<td>The request could not be understood, usually because the ID is not valid for the particular resource.</td>
</tr>
<tr>
<td>401</td>
<td>The session ID or OAuth token has expired or is invalid. Or, if the request is made by a guest user, the resource isn’t accessible to guest users. The response body contains the message and errorCode.</td>
</tr>
<tr>
<td>403</td>
<td>The request has been refused. Verify that the context user has the appropriate permissions to access the requested data, or that the context user is not an external user.</td>
</tr>
<tr>
<td>404</td>
<td>Either the specified resource was not found, or the resource has been deleted.</td>
</tr>
<tr>
<td>409</td>
<td>A conflict has occurred. For example, an attempt was made to update a request to join a group, but that request had already been approved or rejected.</td>
</tr>
<tr>
<td>412</td>
<td>A precondition has failed. For example, in a batch request, if haltOnError is true and a subrequest fails, subsequent subrequests return 412.</td>
</tr>
<tr>
<td>500</td>
<td>An error has occurred within Lightning Platform, so the request could not be completed. Contact Salesforce Customer Support.</td>
</tr>
<tr>
<td>503</td>
<td>Too many requests in an hour or the server is down for maintenance.</td>
</tr>
</tbody>
</table>

Example

This request requires a 15 or 18 character record ID, but this record ID doesn’t contain enough characters:

GET /ui-api/record-ui/003R000000

The response body is:

```
HTTP/1.1 400 Bad Request
Date: Tue, 29 Nov 2016 00:16:36 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: reflected-xss block;report-uri /_/ContentDomainCSPNoAuth?type=xss
Content-Security-Policy: referrer origin-when-cross-origin
Set-Cookie: BrowserId=q27CK9tSSCeI3k1uAYMo6A;Path=/;Domain=.salesforce.com;Expires=Sat, 28-Jan-2017 00:16:36 GMT
Expires: Thu, 01 Jan 1970 00:00:00 GMT
```
[ {
  "errorCode" : "ILLEGAL_QUERY_PARAMETER_VALUE",
  "message" : "recordId should be a valid record ID: 003R000000"
} ]
CHAPTER 2  User Interface API Features

Learn to use User Interface API to work with various Salesforce features.
Build Salesforce UI for records, actions, and favorites.

Work with Records
Use User Interface API to build UI that lets users create, read, update, and delete Salesforce records. Let’s look at two of the trickier aspects of the API: how to work with child relationships, and how to upload binary files to records.

Work with Actions
Use the Actions resources to get data and metadata about actions displayed in the Salesforce UI. Examples include standard and custom buttons, quick actions, and productivity actions.

Manage Favorites
Favorites let users access frequently used pages in Salesforce. Use User Interface API Favorites resources to manage favorites.

Work with List Views
Use the List View resources to get record data and metadata about list views displayed in the Salesforce UI.

Manage Apps
An app is a collection of items that work together to serve a particular function. Salesforce apps come in two flavors: Classic apps and Lightning apps. Classic apps are created and managed in Salesforce Classic. Lightning apps are created and managed in Lightning Experience. Admins customize both types of apps to match the way users work.

Work with Records

Use User Interface API to build UI that lets users create, read, update, and delete Salesforce records. Let’s look at two of the trickier aspects of the API: how to work with child relationships, and how to upload binary files to records.

Note: User Interface API resources operate on all custom objects and supported standard objects on page 12.

Get Child Records
User Interface API has several resources that include child records in their responses. If a record has many child relationships, a response can be huge and have a negative effect on performance. To get only the records you need, request records for specific child relationships and set the page size of the results.

Build UI for Picklists
User Interface API exposes properties and resources that make it easy for a client to build picklists and dependent picklists.

Upload Binary Files
If a record has a Base64 field, you can upload binary data to the field when you create or update a record.
Get Child Records

User Interface API has several resources that include child records in their responses. If a record has many child relationships, a response can be huge and have a negative effect on performance. To get only the records you need, request records for specific child relationships and set the page size of the results.

There are multiple resources that return child records. All of them share some common functionality.

Specify a Child Relationship

Specify child relationships in the format `ObjectApiName.ChildRelationshipName`. For example, to specify the Contacts relationship on an account, use `Account.Contacts`.

Specify spanning relationships in the format `ObjectApiName.ChildRelationshipName.FieldApiName`. For example, to specify the account billing address for an opportunity, use `Opportunity.Account.BillingAddress`.

Paginated Responses

The child relationships records in the response are paginated and include URLs to get the next page and previous pages of child records. Use the `pageSize` parameter to specify the number of records per page. The default is 5.

Responses Include One Level of Child Relationships

User Interface API gets the first set of child relationships that it finds, it doesn’t get the next level of child relationships. For example, this request gets the Opportunities on an Account record. If an opportunity in the response has a child relationship with another object, the API doesn’t return that information.

```
GET /ui-api/record-ui/001R0000003IG0vIAG?childRelationships=Account.Opportunities
```

In this shortened response body, you see the first set of child relationships on the Account record, which in this case is three Opportunity records. The records in the next level of child relationships—on the Opportunity object—don’t appear because it would make the response body too large.

...response shortened...

```
"records" : { 
   "001RM0000003TXRbYAO" : { 
      "apiName" : "Account", 
      "childRelationships" : { 
         "Opportunities" : { 
            "count" : 3, 
            "currentPageUrl" : 
            "/services/data/v41.0/ui-api/records/001RM0000003TXRbYAO/child-relationships/Opportunities?

            fields=5BOpportunity.Name%2C%20Opportunity.Id%5D&page=1&pageSize=5",
            "nextPageUrl" : null, 
            "previousPageUrl" : null, 
            "records" : [ 
               "apiName" : "Opportunity", 
               "childRelationships" : { },
               "fields" : { 
                  "AccountId" : { 
                     "displayValue" : null, 
                     "value" : "001RM0000003TXRbYAO"
                  },
                  "Id" : { 
                     "displayValue" : null, 
                     "value" : "006RM0000002XCGxYAO"
                  },
                  "Name" : 
```

26
Resources That Return Child Records

Get Child Records for a Specific Record and Child Relationship

This example returns all the Contacts related to the Account record 001R0000003I6CoIAK.

```
/ui-api/records/001R0000003I6CoIAK/child-relationships/Contacts
```

The response is paginated. Use the `pageSize` parameter to specify the number of records per page. The default is 5.

```
{
   "count" : 5,
   "currentPageUrl" : "/services/data/v41.0/ui-api/records/001R0000003I6CoIAK/child-relationships/Contacts?page=1&pageSize=5",
   "nextPageUrl" : "/services/data/v41.0/ui-api/records/001R0000003I6CoIAK/child-relationships/Contacts?page=2&pageSize=5",
   "previousPageUrl" : null,
   "records" : [ {
      "apiName" : "Contact",
      "childRelationships" : { },
      "fields" : {
         "AccountId" : { 
            "displayValue" : null,
            "value" : "001R0000003I6CoIAK"
         },
         "Id" : {
            "displayValue" : null,
            "value" : "003R0000007qXwQIAU"
         },
         "Name" : { 
            "displayValue" : null,
            "value" : "John Smith"
         }
      },
      "id" : "003R0000007qXwQIAU",
      "recordTypeInfo" : null
   }, ...
```

27
Get a Record and Include Child Records
This example gets the data for fields in the compact layout for an opportunity record with ID 006R0000001rboIIAQ. The response includes OpportunityCompetitors and Partners child records.


Get a Batch of Records and Include Child Records
This example gets the Account.Name field and Account.Contacts child records for two accounts.

GET /ui-api/records/batch/001R0000003I6CeIAK,001R0000003I6CgIAK?fields=Account.Name&childRelationships=Account.Contacts

Get Record Data and Metadata and Include Child Records
The response for this request includes layout information, metadata, and record data. The record data includes child relationship data for Contacts and Opportunities.


SEE ALSO:
User Interface API Sample Apps
Get Child Records
Get a Record
Get a Batch of Records
Get Record Data and Object Metadata

Build UI for Picklists
User Interface API exposes properties and resources that make it easy for a client to build picklists and dependent picklists.

Picklists are complicated UI elements, and dependent picklists are especially tricky. The values in a dependent picklist field are filtered based on the selection in another field, called a controlling field. A controlling field can be a picklist field or a checkbox field. For example, imagine a Continents picklist that controls a Countries picklist that controls a Cities picklist and a Languages picklist field.

Dependent picklists exist in a field dependency tree that shows a hierarchy of controlling and dependent fields. Each node in the tree can have any number of child nodes. To build UI, you need to know the complete hierarchy.

```
Dependent picklist hierarchy:
- Continents
- Countries
- Cities
- Languages
```
An object can have any number of field dependency trees. User Interface API exposes the collection of field dependency trees in a single property on each object called `dependentFields`. Each dependent field has a `controllingFields` property that lists the fields that control it. User Interface API also provides a batch resource to get all picklist values for a record type:

 `/ui-api/object-info/{objectApiName}/picklist-values/{recordTypeId}`.

### Check for Picklists

To build UI to create, display, edit, and delete records, make a request to one of the User Interface API resources. The response from each of these resources includes an `Object Info` on page 158 response body, which contains the object metadata, including information about picklists.

- Get Record Data and Object Metadata
- Get Object Metadata
- Get Default Values to Create a Record
- Get Default Values to Clone a Record

1. Parse the response for each field’s `dataType`. If the `dataType` is `Picklist` or `MultiPicklist`, the field is a picklist.
2. If you find a picklist, grab the `recordTypeId`, you’ll use it in a request to get the picklist values.
3. Parse the response for dependent picklists. If the field is a picklist and if the `controllingFields` property is non-null, the field is a dependent picklist.
4. If the field is a dependent picklist, look at the `dependentFields` property on the object to see the tree structure of the object’s dependent and controlling fields. A dependent picklist can depend on (be controlled by) standard or and custom picklists and checkboxes.

```json
{
...
"dependentFields" : {
    "Continents__c" : {
      "Countries__c" : {
        "Cities__c" : { },
        "Languages__c" : { },
        
      },
    },
  },
  ...
"fields" : {
    "AccountSource" : {
      ...
    "controllingFields" : [ ],
    ...
    "dataType" : "Picklist",
    ...
  },
  "Cities__c" : {
    ...
    "controllingFields" : [ "Countries__c", "Continents__c" ],
    ...
    "dataType" : "Picklist",
    ...
  },
  "Continents__c" : {
    ...
  }
}
```
Here's a bit more info about the dependent and controlling fields properties.

- **(Object).dependentFields**—A map of the dependent fields tree structure. Each nested object is another Map<String, Object>. When the object is empty, it indicates a leaf of the tree, which is a field that doesn’t control other fields. An object can have multiple independent trees, which means this property can have multiple root objects.

  Imagine an object with the picklists Continents__c, Countries__c, and Cities__c. Continents__c is the root. Cities__c is a leaf. Continents__c and Countries__c are controlling fields. Countries__c and Cities__c are dependent picklists.

- **(Field).controllingFields**—If this field is a dependent picklist, this property is a collection of fields that control the values in the picklist. When there’s a hierarchy of controlling fields, the collection starts with the immediate parent and moves up the tree.

### Get Picklist Values

You know that the object has picklists, you know whether it has dependent picklists, and you know how the dependent picklists are controlled. It’s time to get the picklist values.

1. Do one of the following:
   - To retrieve the values for all picklists of a record type, make a request to GET `/ui-api/object-info/$objectApiName/picklist-values/$recordTypeId`. It’s handy to retrieve the values in a batch instead of requesting them for each field.
   - To get the values for a single picklist, make a request to GET `/ui-api/object-info/$objectApiName/picklist-values/$recordTypeId/$fieldApiName`. 
2. Get the values for independent picklists.
   a. To determine whether a picklist is independent, look at the `{Field}.controllingFields` property. If the property is empty, the picklist is independent.

   Tip: If the `controllerValues` property is empty, like it is for the `AccountSource` field in the following example, the field isn’t necessarily independent. If the controlling field is protected by field-level security (FLS), it doesn’t appear in the `controllerValues` property.

   b. To populate a picklist in the UI, use the field’s `label` and `value` properties.

3. Get the values for dependent picklists.
   a. If a picklist is dependent, its `controllerValues` property lists the values and indexes of its controlling field (except when the immediate controlling field is protected by FLS). The indexes are used in the `validFor` property to indicate which controlling field values map to which picklist value. For example, in `controllerValues`, Italy has index 5. In the `values.validFor` property, you see that Rome is valid for index 5.

   b. To populate a picklist in the UI, use the field’s `label` and `value` properties.

```json
{
  "picklistFieldValues" : {
    "AccountSource" : {
      "controllerValues" : {},
      "defaultValue" : null,
      "url" : "/services/data/v47.0/ui-api/object-info/account/picklist-values/012000000000000AAA/AccountSource",
      "values" : [ {
        "attributes" : null,
        "label" : "Advertisement",
        "validFor" : [],
        "value" : "Advertisement"
      }, {
        "attributes" : null,
        "label" : "Employee Referral",
        "validFor" : [],
        "value" : "Employee Referral"
      }, ...
    },
    "Cities__c" : {
      "controllerValues" : { "Australia" : 0,
        "Brazil" : 1,
        "China" : 2,
        "Colombia" : 3,
        "France" : 4,
        "Italy" : 5,
        "Mexico" : 6,
        "New Zealand" : 7,
        "Nigeria" : 8,
        "Senegal" : 9,
        "South Korea" : 10,
        "US" : 11
      },
    }
  }
}
```
Record Viewer Sample App

To view sample code that parses a response and displays dependent picklists, check out the Record Viewer sample app on GitHub. To contribute to the app, fork it and submit a pull request. We love to see what people do with User Interface API.

SEE ALSO:
- User Interface API Sample Apps
- Get Values for a Picklist Field
- Get Values for All Picklist Fields of a Record Type
Upload Binary Files

If a record has a Base64 field, you can upload binary data to the field when you create or update a record.

The field determines the binary data file size limit. For example, the `ContentNote.Content` field has a different limit than the `Attachment.Body` field. For information about fields, see Standard Objects in Object Reference for Salesforce and Lightning Platform.

There are two ways to upload a binary file in request.

- **Upload Binary Files in a Multipart/Form-Data Request**
- **Upload Binary Files in a JSON Request Body**

**Upload Binary Files in a Multipart/Form-Data Request**

To upload a binary file, create a multipart request with a part that contains the JSON request body required by the resource, and a part for each file you want to upload. Each file must have its own part. To upload binary data to multiple fields, add a part for each field. The part names must match the field names.

Let’s create a ContentNote record and upload a binary file to its `Content` field, which is the content or body of the note.

This curl example performs a multi-part request that contains the JSON request body and the binary file.

```bash
curl -v -H 'Authorization: Bearer 00Dxx0000001gER!ARoAQFFeB_55d8B3Sh....'
-F 'json={"apiName": "ContentNote","fields": {"Title": "My List of Things"}};type=application/json'
-F 'Content=@note.txt;type=application/octet-stream'
-X POST 'http://{instance-name}.salesforce.com/services/data/v47.0/ui-api/records'
```

The first part, starting with `-F`, contains the JSON request body that the `/ui-api/records` resource requires. The request body contains the API name of the record (ContentNote) and the title of the record (My List of Things). The part name must be `json`.

The second part, starting with the second `-F`, contains the binary file. The part name must match the name of the Base64 field to which the data is uploaded. The ContentNote Base64 field name is `Content`, and the plain text file we’re uploading is `note.txt`.

**Upload Binary Files in a JSON Request Body**

The JSON format doesn’t support binary data. To upload a binary file in a JSON payload, encode the file using Base64 encoding.

In this sample Record Input JSON request body, the `Content` property contains the file as a Base64 encoded string.

```json
{
  "apiName": "ContentNote",
  "fields": {
    "Content": "SGksIFRoaXMgaXMgYmFzZTY0",
    "Title": "Sample Title"
  }
}
```

This curl example creates a ContentNote record and uploads the file.

```bash
curl -X POST -H 'Authorization: Bearer 00Dxx0000001gER!ARoAQFFeB_55d8B3Sh....'
-H 'Content-Type: application/json' -H "Content-type: application/json"
-d '{"apiName": "ContentNote","fields": {"Content": "SGksIFRoaXMgaXMgYmFzZTY0","Title": "Sample Title"}}'}
```
Work with Actions

Use the Actions resources to get data and metadata about actions displayed in the Salesforce UI. Examples include standard and custom buttons, quick actions, and productivity actions.

You can use the actions endpoints to fetch lists of actions, but not to execute actions. It’s handy to get a list of actions if you want to use quick actions, if you have Visualforce overrides, or if you want to determine CRUD based on visible actions. For example, if a New action exists, then a user can create a record.

Get Actions for a Record Detail Page

As an example, let’s say we have an Account record detail page with a record ID of 001R0000003IDlwIAG. The page has some standard buttons and quick actions. To get information about all the actions on this page, make this request:

```
GET /services/data/v41.0/ui-api/actions/record/001R0000003IDlwIAG
```

The request returns a list of actions on the page and the URLs of the request and subrequests.

```
{
  "actions": {
    "001R0000003IDlwIAG": {
      "actions": [
        {
          "actionListContext": "Record",
          "actionTarget": null,
          "actionTargetType": "Invoke",
          "apiName": "Edit",
          "externalId": "00DR00000008n7M:001R0000003IDlwIAG::Record:Phone:StandardButton:Edit",
          "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/action/edit_120.png",
          "id": "0JVR0000000017OAA",
          "isMassAction": "false",
          "label": "Edit",
          "primaryColor": "1DCCBF",
          "relatedListRecordId": null,
          "relatedSourceObject": null,
          "section": "Page",
          "sourceObject": "001R0000003IDlwIAG",
          "subtype": null,
          "targetObject": "Account",
          "targetUrl": null,
          "type": "StandardButton"
        },
        {
          "actionListContext": "Record",
          "actionTarget": "/services/data/v41.0/quickActions/NewTask/describe",
```
Filter the Action Results

To filter the results returned by the Actions resources, use the following request parameters:

- **actionTypes**—The action type.
- **formFactor**—The layout display size.
- **sections**—The section of the user interface that the action resides in.

For example, to filter a record actions request so that it includes only standard button actions, set the `actionTypes` parameter to `StandardButton`.

```
GET /services/data/v41.0/ui-api/actions/record/001R0000003IDlwIAG?actionTypes=StandardButton
```

Manage Favorites

Favorites let users access frequently used pages in Salesforce. Use User Interface API Favorites resources to manage favorites.

You can create, get, delete, and update favorites, update all favorites at once, and update favorite usage information, for example, the number of times a favorite was clicked. Responses return all fields required to render favorites.
This request returns all the favorites of the context user.

GET /ui-api/favorites

```
{
  "favorites" : [ {
    "accessCount" : 1,
    "iconColor" : "FCB95B",
    "iconUrl" : "https://.../img/icon/t4v35/standard/opportunity_120.png",
    "id" : "0MVR00000004GGlOAM",
    "lastAccessDate" : "2017-08-07T18:51:54.000Z",
    "name" : "My Best Opportunities",
    "objectType" : "Opportunity",
    "sortOrder" : 1,
    "subtitle" : "Opportunities",
    "target" : "00BR0000000tTTwMAM",
    "targetType" : "ListView"
  }, {
    "accessCount" : 1,
    "iconColor" : "A094ED",
    "iconUrl" : "https://.../img/icon/t4v35/standard/contact_120.png",
    "id" : "0MVR00000004FOPOA2",
    "lastAccessDate" : "2017-07-21T21:35:27.000Z",
    "name" : "John Doe",
    "objectType" : "Contact",
    "sortOrder" : 2,
    "subtitle" : "Contact",
    "target" : "003R0000001bivAIAQ",
    "targetType" : "Record"
  }, {
    "accessCount" : 1,
    "iconColor" : "f6707b",
    "iconUrl" : "https://.../img/icon/t4v35/custom/custom84_120.png",
    "id" : "0MVR00000004D5OAI",
    "lastAccessDate" : "2017-08-11T17:24:01.000Z",
    "name" : "Trend Board",
    "objectType" : "TabDefinition",
    "sortOrder" : 3,
    "subtitle" : "Tab",
    "target" : "Trend_Board",
    "targetType" : "Tab"
  }, {
    "accessCount" : 1,
    "iconColor" : "F88962",
    "iconUrl" : "https://.../img/icon/t4v35/standard/lead_120.png",
    "id" : "0MVR00000004GOpOAM",
    "lastAccessDate" : "2017-08-11T17:24:24.000Z",
    "name" : "Leads to Follow Up on",
    "objectType" : "Lead",
    "sortOrder" : 4,
    "subtitle" : "Leads",
    "target" : "Lead",
    "targetType" : "ListView"
  }]
}
```
Adding a favorite to the top of the list is as easy as knowing what type of favorite it is and how to identify it. For example, to add an Account as a favorite, add it as a Record favorite using its ID. To insert it at the beginning of the list, explicitly set the sortOrder to 1, otherwise it’s inserted at the end of the list.

```json
POST /ui-api/favorites?target=001R0000002LBXU4A&targetType=Record&sortOrder=1
{
  "accessCount" : 1,
  "iconColor" : "7F8DE1",
  "iconUrl" : "https://.../img/icon/t4v35/standard/account_120.png",
  "id" : "0MVR000000004DFOAY",
  "lastAccessDate" : "2017-08-11T19:26:04.000Z",
  "name" : "Global Media",
  "objectType" : "Account",
  "sortOrder" : 1,
  "subtitle" : "Account",
  "target" : "001R0000002LBXU4A",
  "targetType" : "Record"
}
```

To track how often a user accesses a particular favorite, make a patch request using the favorite’s ID and see the accessCount increment and the lastAccessDate update.

```json
PATCH /ui-api/favorites/0MVR000000004DFOAY/usage
{
  "accessCount" : 2,
  "iconColor" : "7F8DE1",
  "iconUrl" : "https://.../img/icon/t4v35/standard/account_120.png",
  "id" : "0MVR000000004DFOAY",
  "lastAccessDate" : "2017-08-11T19:31:17.416Z",
  "name" : "Global Media",
  "objectType" : "Account",
  "sortOrder" : 1,
  "subtitle" : "Account",
  "target" : "001R0000002LBXU4A",
  "targetType" : "Record"
}
```

Removing a favorite is as easy and requires using the favorite’s ID again.

```json
DELETE /ui-api/favorites/0MVR000000004DFOAY
```

To reorder or rename your favorites, you have some options. The first option is to individually update each favorite one by one. When reordering this way, the server automatically updates the sort order of affected favorites in the list so the sort order stays consistent. The second and more useful option is to reorder, rename, and delete favorites in a single request. This option is useful for building an edit dialog where the user can rename some favorites, delete others, and reorder whichever they choose. All these changes can then be made in a single request.

```json
PUT /ui-api/favorites/batch
```

Here’s the input for the PUT request:

```json
{
  "favorites" : [
```
Here's the response for the PUT request:

```json
{
  "favorites" : [ {
    "accessCount" : 1,
    "iconColor" : "F88962",
    "iconUrl" : "https://.../img/icon/t4v35/standard/lead_120.png",
    "id" : "0MVR00000004GOpOAM",
    "lastAccessDate" : "2017-08-11T17:24:24.000Z",
    "name" : "Leads to Follow Up",
    "objectType" : "Lead",
    "sortOrder" : 1,
    "subtitle" : "Leads",
    "target" : "Lead",
    "targetType" : "ListView"
  }, {
    "accessCount" : 1,
    "iconColor" : "FCB95B",
    "iconUrl" : "https://.../img/icon/t4v35/standard/opportunity_120.png",
    "id" : "0MVR00000004GGlOAM",
    "lastAccessDate" : "2017-08-07T18:51:54.000Z",
    "name" : "Top Opportunities",
    "objectType" : "Opportunity",
    "sortOrder" : 2,
    "subtitle" : "Opportunities",
    "target" : "00BR0000000tTTwMAM",
    "targetType" : "ListView"
  }, {
    "accessCount" : 1,
    "iconColor" : "f6707b",
    "iconUrl" : "https://.../img/icon/t4v35/custom/custom84_120.png",
    "id" : "0MVR00000004D5OAI",
    "lastAccessDate" : "2017-08-11T17:24:01.000Z",
    "name" : "Trend Board",
    "objectType" : "TabDefinition",
    "sortOrder" : 3,
    "subtitle" : "Tab",
    "target" : "Trend_Board",
    "targetType" : "Tab"
  } ]
}
```
In this single request, we deleted our contact John Doe, moved Leads to Follow Up to the top of the list, and we renamed My Best Opportunities to Top Opportunities.

Work with List Views

Use the List View resources to get record data and metadata about list views displayed in the Salesforce UI.

Get Record Data and Metadata for a List View

As an example, let’s say we have a list view named All Accounts with a list view ID of 00BR0000000Wc0rMAC. To get information about all the records and metadata for this list view, make this request.

/services/data/v47.0/ui-api/list-ui/00BR0000000Wc0rMAC

You can also use the object API name and list view API name of the list view to make the same request.

/services/data/v47.0/ui-api/list-ui/Account/AllAccounts

The request returns a list of records and the metadata for the list view.

```json
{
    "eTag": "6fea2ca99f3740c72f1667ee522980a7",
    "info": {
        "cloneable": true,
        "createable": true,
        "deletable": true,
        "displayColumns": [
            {
                "fieldApiName": "Name",
                "label": "Account Name",
                "sortable": true
            },
            {
                "fieldApiName": "Site",
                "label": "Account Site",
                "sortable": true
            },
            {
                "fieldApiName": "BillingState",
                "label": "Billing State/Province",
                "sortable": true
            },
            {
                "fieldApiName": "Phone",
                "label": "Phone",
                "sortable": true
            },
            {
                "fieldApiName": "Type",
                "label": "Type",
                "sortable": true
            },
            {
                "fieldApiName": "Owner.Alias",
                "label": "Account Owner Alias",
                "sortable": true
            }
        ],
        "eTag": "f5f522aef88a0cb38be54b9c0095fe3c",
        "filterLogicString": null,
        "filteredByInfo": [],
        "label": "All Accounts"
    }
}
```
"listReference": {
  "id": "00BRM000002KPM42AO",
  "listViewApiName": "AllAccounts",
  "objectApiName": "Account",
  "type": "listView"
},
"orderByInfo": [{
  "fieldApiName": "Name",
  "isAscending": true,
  "label": "Account Name"
}],
"updateable": true,
"userPreferences": {
  "columnWidths": {
    "Site": -1,
    "Type": -1,
    "Owner.Alias": -1,
    "Phone": -1,
    "BillingState": -1,
    "Name": -1
  },
  "columnWrap": {
    "Site": false,
    "Type": false,
    "Owner.Alias": false,
    "Phone": false,
    "BillingState": false,
    "Name": false
  }
},
"visibility": "Public",
"visibilityEditable": true
},
"records": {
  "count": 12,
  "currentPageToken": "0",
  "currentPageUrl": "/services/data/v43.0/ui-api/list-records/Account/AllAccounts?pageSize=50&pageToken=0",
  "listInfoETag": "f5f522aef88a0cb38be54b9c0095fe3c",
  "nextPageToken": null,
  "nextPageUrl": null,
  "previousPageToken": null,
  "previousPageUrl": null,
  "records": [
    {
      "apiName": "Account",
      "childRelationships": {},
      "eTag": "d55bcd780f2145b724be5a563f37b9b2",
      "fields": {
        "BillingState": {
          "displayValue": null,
          "value": "NC"
        },
        "CreatedDate": {

"displayValue": null,
"value": "2018-04-11T00:37:56.000Z"
},
"Id": {
  "displayValue": null,
  "value": "001RM000003UNtxYAG"
},
"LastModifiedDate": {
  "displayValue": null,
  "value": "2018-04-11T00:37:56.000Z"
},
"Name": {
  "displayValue": null,
  "value": "Burlington Textiles Corp of America"
},
"Owner": {
  "displayValue": null,
  "value": {
    "apiName": "User",
    "childRelationships": { },
    "eTag": "671bc4877d13cd4f43d28a671636f873",
    "fields": {
      "Alias": {
        "displayValue": null,
        "value": "AUser"
      },
      "Id": {
        "displayValue": null,
        "value": "005RM000001cNuJYAU"
      }
    },
    "id": "005RM000001cNuJYAU",
    "recordTypeInfo": null
  }
},
"OwnerId": {
  "displayValue": null,
  "value": "005RM000001cNuJYAU"
},
"Phone": {
  "displayValue": null,
  "value": "(336) 222-7000"
},
"Site": {
  "displayValue": null,
  "value": null
},
"SystemModstamp": {
  "displayValue": null,
  "value": "2018-04-11T00:37:56.000Z"
},
"Type": {
  "displayValue": "Customer - Direct",
  "value": "Customer - Direct"}
Modify the List View Results

To modify the results returned by the List View resources, use the following request parameters:

- **fields**—Additional fields queried for the records returned, for display purposes. If a field is specified and the user doesn’t have access to it, an error occurs.
- **optionalFields**—Additional fields queried for the records returned, for display purposes. If a field is specified and the user doesn’t have access to it, no error occurs.
- **pageSize**—The number of list records viewed at one time.
- **pageToken**—A token that represents the page offset.
- **sortBy**—The API name of the field the list view is sorted by.

For example, to sort records by the date they were created, set the `sortBy` parameter to `CreatedDate`.

```
GET /services/data/v47.0/ui-api/list-ui/00BR0000000Wc0rMAC?sortBy=CreatedDate
```

**Note:** Depending on the list view object, some extra fields that were not requested in the `fields` or `optionalFields` parameters can be returned. For example, if person accounts are enabled for the org, a request to get list view records can return the `isPersonAccount` field, among others.

Manage Apps

An app is a collection of items that work together to serve a particular function. Salesforce apps come in two flavors: Classic apps and Lightning apps. Classic apps are created and managed in Salesforce Classic. Lightning apps are created and managed in Lightning Experience. Admins customize both types of apps to match the way users work.

Classic apps are a collection of standard and custom tabs, including most standard objects, custom objects, Visualforce tabs, Lightning component tabs, Canvas apps via Visualforce tabs, and web tabs. In Salesforce Classic, users use tabs to access objects, like accounts and leads. Tabs also serve up other features, such as the Home page or Chatter.

Lightning apps are a collection of items that include everything from the Classic apps list, plus Lightning page tabs, and utilities like Lightning Dialer. In Lightning apps, admins customize the app’s logo and enhance branding by customizing the color of the navigation bar. Navigation in Lightning Experience is similar to navigation in Salesforce Classic, but with some improvements. In Lightning Experience, users can complete actions and access recent records and lists directly from the navigation bar. Users can also switch between apps that have custom colors and branding.

Users can also personalize which items appear in their navigation bar in Lightning Experience or tabs in Salesforce Classic. Although users can’t completely override the navigation created by admins, they can add and reorder items or tabs from their view. In addition, in Lightning Experience, users can add Favorite items directly to their navigation bar.

Using these App resources, you can get a list of apps, items from an app, and the presentation data for a user’s navigation.
To return metadata for a list of all apps for the current user, use this resource.

GET /ui-api/apps

To return metadata about a specific app, use this resource.

GET /ui-api/apps/${appId}

To return metadata about a specific app, and save the app as the last accessed app for the user, use this resource.

PATCH /ui-api/apps/${appId}

To return metadata about the app the current user last selected, or sees by default, use this resource.

GET /ui-api/apps/selected

To return metadata about a user's personalized navigation, use this resource.

GET /ui-api/apps/${appId}/user-nav-items

To return metadata for a list of all navigation tabs for the current user, use this resource on page 135.

GET /ui-api/nav-items
CHAPTER 3  User Interface API Resources

This chapter contains User Interface API resources (also called endpoints).

Records
Use these resources to get metadata and data about objects and records. These resources make it easy to build Salesforce UI.

Actions
Use these resources to get data and metadata about actions displayed in the Salesforce UI. Examples include standard and custom buttons, quick actions, and productivity actions.

Favorites
Use these resources to create and update a single favorite or a collection of favorites, and to get favorites usage information. These resources make it easy to build UI for favorites.

List Views
Use these resources to get data and metadata about list views displayed in the Salesforce UI.

Apps
Use these resources to get data and metadata about apps displayed in the Salesforce UI.

Get Active Theme
Get a Salesforce org’s active theme. A theme uses colors, images, and banners to change the overall appearance of Salesforce. Administrators can define themes and switch themes to provide a different look. The User Interface API response matches the Admin’s selection.

Records
Use these resources to get metadata and data about objects and records. These resources make it easy to build Salesforce UI.

Get Record Data and Object Metadata
Get layout information, metadata, and data to build UI for a single record or for a collection of records.

Get Record Layout Metadata
Get metadata about page layouts for the specified object type.

Get a Directory of Supported Objects
Get a directory of objects that are supported by User Interface API and accessible to the context user. The directory includes a /ui-api/object-info/{objectApiName} resource for each supported object.

Get Object Metadata
Get metadata about a specific object. The response includes metadata describing fields, child relationships, record type, and theme.

Get Values for a Picklist Field
If a field’s dataType property is equal to Picklist or MultiPicklist, use this resource to get its values.
Get Values for All Picklist Fields of a Record Type
Use this resource to get the values for all the picklist fields of a specific record type. This resource is especially useful for getting dependent picklist values. For example, if an object has a tree of dependent picklists (Continents__c, Countries__c, Cities__c), use this resource to get all the values for each picklist in one request.

Get Child Records
Get child records for a specified record and child relationship name. Relationships are connections between records. On a record detail page, each record in a related list has a child relationship to the parent record.

Get a Record
Get a record’s data.

Get a Batch of Records
Get data for a batch of records.

Create a Record
Create a record. First, make a request to the Clone Record Default or Create Record Default resources to get the default metadata and data for the record.

Get Default Values to Clone a Record
Get the default layout information, object information, and data for cloning a record. After getting the default values, make a request to POST /ui-api/records to create the record.

Get Default Values to Create a Record
Get the default values for fields for a new record of a specified object and optional record type. After getting the default values, make a request to POST /ui-api/records to create the record.

Update a Record
Update a record’s data.

Delete a Record
Delete a record.

Get Lookup Field Suggestions
When a user edits a lookup field, use this resource to search for and display suggestions. You can search for most recently used matches, for matching names, or for any match in a searchable field. You can also specify lookup filter bindings for dependent lookups.

Get Lookup Field Suggestions for a Specified Object
When a user edits a lookup field, use this resource to search for and display suggestions for a specified object. You can search for most recently used matches, for matching names, or for any match in a searchable field. You can also specify lookup filter bindings for dependent lookups.

Get Record Data and Object Metadata
Get layout information, metadata, and data to build UI for a single record or for a collection of records.

The response contains layout information for whichever layout types are specified in the layoutTypes parameter.

It contains data for the records specified in the recordIds parameter.

The response contains object metadata for the object types of the records specified in the recordIds parameter, and for any nested objects. For example, a request to /ui-api/record-ui/001RM000003Rs0HYAO, which is an Account, returns object metadata for Account and User, because the OwnerId field on the Account object contains a reference to the User object.
**Resource**

/ui-api/record-ui/{recordIds}

- **recordIds**—A comma-delimited list of custom object records and supported object records.

**Available Version**

41.0

**HTTP Method**

GET

**Example**

The response for this request includes layout information, metadata, and record data. The record data includes child relationship data for Contacts and Opportunities.

GET


**Request Parameters**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>childRelationships</td>
<td>String[]</td>
<td>A collection of child relationship names. The records with those child relationship names are included in the response. Specify names in the format <code>ObjectName.ChildRelationshipName</code> or <code>ObjectName.ChildRelationshipName.FieldApiName</code>. For example, to specify the Contacts relationship on an Account, use <code>Account.Contacts</code>. You can get child relationships one level deep. To get a relationship name, look in the Object Info response body.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
</tbody>
</table>
| formFactor      | String     | The layout display size for the record. One of these values:
  - Large—(Default) Use this value to get a layout for desktop display size.
  - Medium—Use this value to get a layout for tablet display size.
  - Small—Use this value to get a layout for phone display size. | Optional             | 41.0              |
| layoutTypes     | String[]   | The layout type for the record. A collection of any of these values:
  - Compact—Use this value to get a layout that contains a record’s key fields. | Optional             | 41.0              |
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Available Version</th>
</tr>
</thead>
</table>
| modes          | String[]   | The access mode for the record. This value determines which fields to get from a layout. Layouts have different fields for create, edit, and view modes. For example, formula fields are rendered in view mode, but not in create mode because they're calculated at run time, like formulas in a spreadsheet. A collection of any of these values:   
|                |            | • Full—(Default) Use this value to get a full layout.                                                                                                                                                        | 41.0                          |
|                |            | • Create—Use this mode if you intend to build UI that lets a user create a record. This mode is used by the /ui-api/record-defaults/create/{apiName} resource.                                                  |                               |
|                |            | • Edit—Use this mode if you intend to build UI that lets a user edit a record. This mode is used by the /ui-api/record-defaults/clone/{recordId} resource.                                                          |                               |
|                |            | • View—(Default) Use this mode if you intend to build UI that displays a record.                                                                                                                                |                               |
| optionalFields | String[]   | A collection of optional field names. If a field is accessible to the context user, it’s included in the response. If a field isn’t accessible to the context user, it isn’t included in the response, but it doesn’t cause an error. Specify names in the format **ObjectApiName.Field**. You can get any field that has a named relationship to this record. There is no limit to the number of fields you can specify. | 41.0                          |
| pageSize       | Integer    | The maximum number of child records to return on a page.                                                                                                                                                     | 41.0                          |
| updateMru      | Boolean    | To add to the most recently used (MRU) list view, set to true. The default value is false.                                                                                                                   | 47.0                          |
Response Body

Record UI

SEE ALSO:
- User Interface API Quick Start
- Object Info
- Get Child Records

Get Record Layout Metadata

Get metadata about page layouts for the specified object type.

Resource

```plaintext
/ui-api/layout/{objectApiName}
```

- `objectApiName`—A supported object.

Available Version

41.0

HTTP Method

GET

Request Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size for the record. One of these values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—(Default) Use this value to get a layout for desktop display size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—Use this value to get a layout for tablet display size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—Use this value to get a layout for phone display size.</td>
</tr>
<tr>
<td>layoutType</td>
<td>String</td>
<td>The layout type for the record. One of these values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Compact—Use this value to get a layout that contains a record’s key fields.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Full—(Default) Use this value to get a full layout.</td>
</tr>
<tr>
<td>mode</td>
<td>String</td>
<td>The access mode for the record. This value determines which fields to get from a layout. Layouts have different fields for create, edit, and view modes. For example,</td>
</tr>
</tbody>
</table>
formula fields are rendered in view mode, but not in create mode because they’re calculated at run time, like formulas in a spreadsheet. One of these values:

- **Create**—Use this mode if you intend to build UI that lets a user create a record. This mode is used by the `/ui-api/record-defaults/create/{apiName}` resource.

- **Edit**—Use this mode if you intend to build UI that lets a user edit a record. This mode is used by the `/ui-api/record-defaults/clone/{recordId}` resource.

- **View**—(Default) Use this mode if you intend to build UI that displays a record.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordTypeId</td>
<td>Id</td>
<td>The ID of the record type (RecordType object) for the new record. If not provided, the default record type is used.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
</tbody>
</table>

**Response Body**

- **Record Layout**

**SEE ALSO:**
- User Interface API Quick Start

### Get a Directory of Supported Objects

Get a directory of objects that are supported by User Interface API and accessible to the context user. The directory includes a `/ui-api/object-info/{objectApiName}` resource for each supported object.

**Resource**

- `/ui-api/object-info/`

**Available Version**

- 42.0

**HTTP Method**

- GET
Get Object Metadata

Get metadata about a specific object. The response includes metadata describing fields, child relationships, record type, and theme.

Resource

```
/ui-api/object-info/{objectApiName}
```

- `objectApiName`—A supported object.

Available Version

41.0

HTTP Method

GET

Response Body

Object Info

SEE ALSO:

- [User Interface API Quick Start](#)
- [Get a Directory of Supported Objects](#)

Get Values for a Picklist Field

If a field’s `dataType` property is equal to `Picklist` or `MultiPicklist`, use this resource to get its values.

Resource

```
/ui-api/object-info/{objectApiName}/picklist-values/{recordTypeId}/{fieldApiName}
```

- `objectApiName`—The API name of a supported object.
- `recordTypeId`—The ID of the record type.
- `fieldApiName`—The API name of the picklist field on the object.

Available Version

41.0

HTTP Method

GET
Get Values for All Picklist Fields of a Record Type

Use this resource to get the values for all the picklist fields of a specific record type. This resource is especially useful for getting dependent picklist values. For example, if an object has a tree of dependent picklists (Continents__c, Countries__c, Cities__c), use this resource to get all the values for each picklist in one request.

**Resource**

```
/ui-api/object-info/{objectApiName}/picklist-values/{recordTypeId}
```

- **objectApiName**—The API name of a supported object.
- **recordTypeId**—The ID of the record type.

**Available Version**

42.0

**HTTP Method**

GET

**Response Body**

Picklist Values Collection

**SEE ALSO:**

Build UI for Picklists
Get Values for All Picklist Fields of a Record Type
Get Record Data and Object Metadata
Get Object Metadata
User Interface API Quick Start

Get Child Records

Get child records for a specified record and child relationship name. Relationships are connections between records. On a record detail page, each record in a related list has a child relationship to the parent record.

**Resource**

```
/ui-api/records/${recordId}/child-relationships/${relationshipName}
```

- **recordId**—The ID of the record for which you want to get child relationships.
- **relationshipName**—The name of the child relationship. This value is returned in the Object Info response body and is usually the plural form of the object name.

**Available Version**

41.0
**HTTP Method**

GET

**Example**

This example returns all the Contacts related to the Account record 001R0000003I6CoIAK.

/\ui-api/records/001R0000003I6CoIAK/child-relationships/Contacts

The response is paginated. Use the pageSize parameter to specify the number of records per page. The default is 5.

```json
{
  "count" : 5,
  "currentPageUrl" : "/services/data/v41.0/ui-api/records/001R0000003I6CoIAK/child-relationships/Contacts?page=1&pageSize=5",
  "nextPageUrl" : "/services/data/v41.0/ui-api/records/001R0000003I6CoIAK/child-relationships/Contacts?page=2&pageSize=5",
  "previousPageUrl" : null,
  "records" : [ {
    "apiName" : "Contact",
    "childRelationships" : { },
    "fields" : {
      "AccountId" : {
        "displayValue" : null,
        "value" : "001R0000003I6CoIAK"
      },
      "Id" : {
        "displayValue" : null,
        "value" : "003R0000007qXwQIAU"
      },
      "Name" : {
        "displayValue" : null,
        "value" : "John Smith"
      },
      "id" : "003R0000007qXwQIAU",
      "recordTypeInfo" : null
    },
    "id" : "003R0000007qXwQIAU",
    "recordTypeInfo" : null
  }, ...
```

**Request Parameters**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>fields</td>
<td>String[]</td>
<td>Specifies the fields to return. If this property is specified, the response is a union of fields and optionalFields. If the context user doesn’t have access to a field, an error is returned.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Type</td>
<td>Description</td>
<td>Required or Optional</td>
<td>Available Version</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you’re not sure whether the context user has access to a field and you don’t want the request to fail if they don’t, use the <code>optionalFields</code> parameter.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify names in the format <code>ObjectApiName.FieldName</code>. You can get any field that has a named relationship to this record. There is no limit to the number of fields you can specify.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>optionalFields</td>
<td>String[]</td>
<td>A collection of optional field names. If a field is accessible to the context user, it's included in the response. If a field isn't accessible to the context user, it isn't included in the response, but it doesn't cause an error.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify names in the format <code>ObjectApiName.FieldName</code>. You can get any field that has a named relationship to this record. There is no limit to the number of fields you can specify.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>page</td>
<td>Int</td>
<td>The page offset from which to begin returning records. The default value is 0, which returns records from the first page. For example, for <code>page=2</code> and <code>pageSize=10</code>, the first record returned is the 21st record in the list.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>pageSize</td>
<td>Int</td>
<td>The maximum number of child records to return on a page. The default value is 5.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>pageToken</td>
<td>String</td>
<td>A token that represents the page offset.</td>
<td>Optional</td>
<td>44.0</td>
</tr>
</tbody>
</table>

**Response Body**

**Record Collection**

**SEE ALSO:**

- Get Child Records

**Get a Record**

Get a record's data.

```
GET /ui-api/records/{recordId}
```
recordId—A custom object record or a supported object record.

Available Version
41.0

Example for GET
This example gets the data for fields in the compact layout for an opportunity record with ID 006R0000001rboIIAQ. The response includes OpportunityCompetitors and Partners child records.

GET

Request Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>childRelationships</td>
<td>String[]</td>
<td>A collection of child relationship names. The records with those child relationship names are included in the response. Specify names in the format ObjectApiName. ChildRelationshipName or ObjectApiName. ChildRelationshipName.FieldName. For example, to specify the Contacts relationship on an Account, use Account.Contacts. You can get child relationships one level deep. To get a relationship name, look in the Object Info response body.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>fields</td>
<td>String[]</td>
<td>Specifies the fields to return. If this property is specified, the response is a union of fields and optionalFields. If the context user doesn’t have access to a field, an error is returned. If you’re not sure whether the context user has access to a field and you don’t want the request to fail if they don’t, use the optionalFields parameter. Specify names in the format ObjectApiName.FieldName. You can get any field that has a named relationship to this record. There is no limit to the number of fields you can specify. Child relationship records don’t have layouts. To get more fields for these records, specify them in the fields parameter.</td>
<td>In API versions 45.0 and later, either fields, optionalFields, or layoutTypes is required. In earlier API versions, either fields or layoutTypes is required. In all API versions, to specify both fields and layoutTypes, you must specify childRelationships.</td>
<td>41.0</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Type</td>
<td>Description</td>
<td>Required or Optional</td>
<td>Available Version</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>
| layoutTypes    | String[]        | Specifies the fields to return. If this property is specified, the response is a union of layoutTypes, modes, and optionalFields. A collection containing any of these values:  
  - Compact—Use this value to get a layout that contains a record's key fields.  
  - Full—(Default) Use this value to get a full layout.                                                                                                                                            | Optional if layoutTypes is specified. In earlier API versions, either fields or layoutTypes is required. In all API versions, to specify both fields and layoutTypes, you must specify childRelationships. | 41.0               |
| modes          | String[]        | The access mode for the record. This value determines which fields to get from a layout. Layouts have different fields for create, edit, and view modes. For example, formula fields are rendered in view mode, but not in create mode because they're calculated at run time, like formulas in a spreadsheet. A collection containing any of these values:  
  - Create—Use this mode if you intend to build UI that lets a user create a record. This mode is used by the /ui-api/record-defaults/create/{apiName} resource.  
  - Edit—Use this mode if you intend to build UI that lets a user edit a record. This mode is used by the /ui-api/record-defaults/clone/{recordId} resource.  
  - View—(Default) Use this mode if you intend to build UI that displays a record.                                                                                                                | Optional if layoutTypes is specified. If layoutTypes is not specified, this parameter is ignored. | 41.0               |
### Response Body

**Record**

### SEE ALSO:

- Get Child Records

## Get a Batch of Records

Get data for a batch of records.

### Resource

```plaintext
GET /ui-api/records/batch/{recordIds}
```

**recordIds**—A comma-delimited list of custom object and supported object records.

### Available Version

41.0

### Example for GET

This example gets the Account.Name field and Account.Contacts child records for two accounts.

```plaintext
GET /ui-api/records/batch/001R0000003I6CeIAK,001R0000003I6CgIAK?
fields=Account.Name&childRelationships=Account.Contacts
```

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>optionalFields</td>
<td>String[]</td>
<td>A collection of optional field names. If a field is accessible to the context user, it’s included in the response. If a field isn’t accessible to the context user, it isn’t included in the response, but it doesn’t cause an error. Specify names in the format <code>ObjectApiName.FieldName</code>. You can get any field that has a named relationship to this record. There is no limit to the number of fields you can specify. To return a field whose type is <code>base64</code>, specify the field in the <code>optionalFields</code> parameter.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>pageSize</td>
<td>Integer</td>
<td>The maximum number of child records to return on a page.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>updateMru</td>
<td>Boolean</td>
<td>To add to the most recently used (MRU) list view, set to <code>true</code>. The default value is <code>false</code>.</td>
<td>Optional</td>
<td>47.0</td>
</tr>
</tbody>
</table>
### Request Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>childRelationships</td>
<td>String[]</td>
<td>A collection of child relationship names. The records with those child relationship names are included in the response. Specify names in the format <code>ObjectName ChildRelationshipName</code> or <code>ObjectName ChildRelationshipName FieldName</code>. For example, to specify the Contacts relationship on an Account, use <code>Account.Contacts</code>. You can get child relationships one level deep. To get a relationship name, look in the Object Info response body.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>fields</td>
<td>String[]</td>
<td>Specifies the fields to return. If this property is specified, the response is a union of fields and optionalFields. If the context user doesn’t have access to a field, an error is returned. If you’re not sure whether the context user has access to a field and you don’t want the request to fail if they don’t, use the optionalFields parameter. Specify names in the format <code>ObjectName.FieldName</code>. You can get any field that has a named relationship to this record. There is no limit to the number of fields you can specify. Child relationship records don’t have layouts. To get more fields for these records, specify them in the <code>fields</code> parameter. To return a field whose type is base64, specify the field in the optionalFields parameter.</td>
<td>In API versions 45.0 and later, either fields, optionalFields, or layoutTypes is required. In earlier API versions, either fields or layoutTypes is required. In all API versions, to specify both fields and layoutTypes, you must specify childRelationships</td>
<td>41.0</td>
</tr>
<tr>
<td>layoutTypes</td>
<td>String[]</td>
<td>Specifies the fields to return. If this property is specified, the response is a union of layoutTypes, modes, and optionalFields. A collection containing any of these values:</td>
<td>In API versions 45.0 and later, either fields, optionalFields, or layoutTypes is required. In earlier API versions, either fields or layoutTypes is required. In all API versions, to specify both fields and layoutTypes, you must specify childRelationships</td>
<td>41.0</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Type</td>
<td>Description</td>
<td>Required or Optional</td>
<td>Available Version</td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>
| modes          | String[]   | The access mode for the record. This value determines which fields to get from a layout. Layouts have different fields for create, edit, and view modes. For example, formula fields are rendered in view mode, but not in create mode because they're calculated at run time, like formulas in a spreadsheet. A collection containing any of these values:  
  - Create—Use this mode if you intend to build UI that lets a user create a record. This mode is used by the /ui/api/record-defaults/create/{apiName} resource.  
  - Edit—Use this mode if you intend to build UI that lets a user edit a record. This mode is used by the /ui/api/record-defaults/clone/{recordId} resource.  
  - View—(Default) Use this mode if you intend to build UI that displays a record. | Optional if layoutTypes is specified. If layoutTypes is not specified, this parameter is ignored. | 41.0               |
| optionalFields | String[]   | A collection of optional field names. If a field is accessible to the context user, it's included in the response. If a field isn't accessible to the context user, it isn't included in the response, but it doesn't cause an error. | In API versions 45.0 and later, either fields, optionalFields, or layoutTypes is required. | 41.0               |
### Parameter Name

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
<td>Specify names in the format <code>ObjectName.FieldName</code>. You can get any field</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>that has a named relationship to this record. There is no limit to the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>number of fields you can specify. To return a field whose type is <code>base64</code>,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>specify the field in the <code>optionalFields</code> parameter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pageSize</td>
<td>Integer</td>
<td>The maximum number of child records to return on a page.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>updateMru</td>
<td>Boolean</td>
<td>To add to the most recently used (MRU) list view, set to <code>true</code>. The default</td>
<td>Optional</td>
<td>47.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value is <code>false</code>.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Response Body

```
Batch Results
```

### Create a Record

Create a record. First, make a request to the Clone Record Default or Create Record Default resources to get the default metadata and data for the record.

As of API version 43.0, if you pass read-only fields in a request body, the response is an `Error with Output`.

### Resource

```
/ui-api/records
```

### Available Version

41.0

### HTTP Method

POST
Example

POST /ui-api/records

```json
{
  "apiName": "Account",
  "fields": {
    "Name": "Universal Containers"
  }
}
```

To provide an address, provide the individual fields as separate inputs. The compound field is only available in the response.

```json
{
  "apiName": "Account",
  "fields": {
    "Name": "Local Boxes",
    "BillingState": "WA",
    "BillingStreet": "123 Main Street",
    "BillingCountry": "USA"
  }
}
```

Request body

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>allowSaveOnDuplicate</td>
<td>Boolean</td>
<td>Specifies whether to save a duplicate record (true), or not (false). The default value is false.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>apiName</td>
<td>String</td>
<td>To create a record, specify the API name of an Object from which to create the record. To update a record, use null or don’t pass this property.</td>
<td>Required</td>
<td>41.0</td>
</tr>
<tr>
<td>fields</td>
<td>Map&lt;String, Object&gt;</td>
<td>Map of field names to field values. Format data types according to these rules:</td>
<td>Required</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Address—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Base64—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Boolean—JSON Boolean, true or false</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Currency—JSON Float</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Date—JSON string in correct format</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DateTime—JSON string in ISO 8601 format</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Double—JSON Float</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Email—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• EncryptedString—JSON String</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To specify the master record type, either don’t specify the recordTypeId field, or set it to null.

Response Body

Record

SEE ALSO:
Upload Binary Files

Get Default Values to Clone a Record

Get the default layout information, object information, and data for cloning a record. After getting the default values, make a request to POST /ui-api/records to create the record.

The response contains the default field values for a record cloned from the record specified in {recordId}, optionally of the specified recordTypeId.

It also contains the corresponding layout information for edit mode. In the Salesforce user interface, an admin with “Customize Application” permission can mark a field as required in a layout. When you’re building UI, to determine which fields to mark as required in a layout for edit mode, use the RecordLayoutItem.required property.

The response contains object metadata for the object type of the record specified in {recordId} and for any nested objects. For example, /ui-api/record-defaults/clone/001d000000AtfRIAAZ is a request to clone an Account record. It returns object metadata for Account and User, because the OwnerId field on the Account object contains a reference to the User object.

Resource

/ui-api/record-defaults/clone/{recordId}

Available Version

41.0
HTTP Method
GET

Request Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size for the record. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—(Default) Use this value to get a layout for desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—Use this value to get a layout for tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—Use this value to get a layout for phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>optionalFields</td>
<td>String[]</td>
<td>A collection of optional field names. If a field is accessible to the context user, it’s included in the response. If a field isn’t accessible to the context user, it isn’t included in the response, but it doesn’t cause an error. Specify names in the format <code>ObjectName.FieldName</code>. You can get any field that has a named relationship to this record. There is no limit to the number of fields you can specify.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>recordTypeId</td>
<td>Id</td>
<td>The ID of the record type (RecordType object) for the new record. If not provided, the default record type is used.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
</tbody>
</table>

Response Body

Record Defaults

SEE ALSO:
User Interface API Quick Start

Get Default Values to Create a Record

Get the default values for fields for a new record of a specified object and optional record type. After getting the default values, make a request to `POST /ui-api/records` to create the record.

The response contains the default field values for the Full layout type for a new record of the object type specified in `{objectApiName}`.
It also contains the corresponding layout for create mode. In the Salesforce user interface, an admin with “Customize Application” permission can mark a field as required in a layout. When you’re building UI, to determine which fields to mark as required in a layout for create mode, use the `RecordLayoutItem.required` property.

The response contains object metadata for the object specified in `{objectApiName}` and for any nested objects. For example, a request to `/ui-api/record-defaults/create/Account` returns object metadata for Account and User, because the `OwnerId` field on the Account object contains a reference to the User object.

**Resource**

```
/ui-api/record-defaults/create/{objectApiName}
```

- `{objectApiName}`—The API name of the object type for the new record.

**Available Version**

41.0

**HTTP Method**

GET

**Request Parameters**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
</table>
| formFactor    | String   | The layout display size for the record. One of these values:  
  - Large—(Default) Use this value to get a layout for desktop display size.  
  - Medium—Use this value to get a layout for tablet display size.  
  - Small—Use this value to get a layout for phone display size. | Optional | 41.0 |
| optionalFields | String[] | A collection of optional field names. If a field is accessible to the context user, it’s included in the response. If a field isn’t accessible to the context user, it isn’t included in the response, but it doesn’t cause an error. Specify names in the format `{ObjectApiName}.FieldName`. You can get any field that has a named relationship to this record. There is no limit to the number of fields you can specify. | Optional | 41.0 |
| recordTypeId  | Id       | The ID of the record type (RecordType object) for the new record. If not provided, the default record type is used. | Optional | 41.0 |
Response Body

Record Defaults

SEE ALSO:

User Interface API Quick Start

Update a Record

Update a record’s data.

User Interface API enforces Salesforce validation rules. If a validation rule fails, the response is an Error with Output.

When you make a PATCH request to update a record, make sure that the record hasn’t changed since the user started editing it. To find out whether it’s safe to save a record, pass the `If-Modified-Since` HTTP header in the request.

As of API version 43.0, if you pass read-only fields in a request body, the response is an Error with Output.

```
/ui-api/records/{recordId}
```

Available Version

41.0

HTTP Method

PATCH

Example

This example passes a request body to update the name of an Account.

```
PATCH /ui-api/records/001R0000003Gjrt
{
  "fields": {
    "Name": "Universal Containers"
  }
}
```

To provide an address, provide the individual fields as separate inputs. The compound field is only available in the response.

```
{
  "fields": {
    "Name": "Local Boxes",
    "BillingState": "WA",
    "BillingStreet": "123 Main Street",
    "BillingCountry": "USA"
  }
}
```

Request Body

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>allowSaveOnDuplicate</td>
<td>Boolean</td>
<td>Specifies whether to save a duplicate record (true), or not (false). The default value is false.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
</tbody>
</table>
### Update a Record

#### Name: apiName
**Type:** String
**Description:**
To create a record, specify the API name of an Object from which to create the record. To update a record, use null or don't pass this property.

#### Name: fields
**Type:** Map<String, Object>
**Description:** Map of field names to field values.
**Required or Optional:** Required
**Available Version:** 41.0

### Response Body
- **Record**

SEE ALSO:
- Upload Binary Files
Delete a Record

Delete a record.

/ui-api/records/{recordId}

Available Version
41.0

HTTP Method
DELETE

Delete a record
Deleting a record returns a 200 HTTP status code.

DELETE /ui-api/records/001R0000003Gjrt

Get Lookup Field Suggestions

When a user edits a lookup field, use this resource to search for and display suggestions. You can search for most recently used matches, for matching names, or for any match in a searchable field. You can also specify lookup filter bindings for dependent lookups.

Resource

/ui-api/lookups/{objectApiName}/{fieldApiName}

- objectApiName—The API name of a source object.
- fieldApiName—The API name of a lookup field on the source object.

Available Version
41.0

HTTP Method
GET

Example: Get Most Recently Used Records
A lookup relationship field links a source object to a target object. To edit a lookup field, users click a lookup icon to search for and select a value from a popup list. The target object provides the values in the list.

For example, the Opportunity object has an AccountId lookup field. When a user edits an Opportunity record, the user selects an Account to associate with the Opportunity. To build UI that suggests the most recently used Account records to the user, use this resource. The default value of the searchType parameter is Recent, so the request doesn’t specify it.

GET /ui-api/lookups/Opportunity/AccountId

Example: Search for Suggestions
To search all searchable fields for suggestions, set searchType=Search. This example searches for strings that start with the characters ca (a wildcard is implied).

GET /ui-api/lookups/Opportunity/AccountId?searchType=Search&q=ca

Example: Search Record Names for Suggestions
To search record name fields for suggestions, set searchType=TypeAhead. This example searches record names for strings that start with the characters ta (a wildcard is implied).

GET /ui-api/lookups/Opportunity/AccountId?searchType=TypeAhead&q=ta
Example: Search Dependent Lookups for Suggestions

Some lookup fields have lookup filters that restrict their valid values by referencing fields on the source object. These fields are called dependent lookups. For example, you can filter the Case Contact field to show only contacts that are associated with the account selected in the Case Account Name field.

In the Salesforce UI, this lookup filter is Contact Name: Account Name ID EQUALS Case: Account Name ID. In User Interface API, lookup filter information appears in the filteredLookupInfo property of the Object Info response body, which is returned from several resources, including /ui-api/record-ui/{recordIds}. If the filteredLookupInfo property is non-null, the field it describes is a dependent lookup.

In this example, the Case object’s ContactId field has a filteredLookupInfo property containing a controllingFields property. The controlling field is AccountId. To restrict the lookup search, you must pass all controlling fields and their values to the lookups resource in the dependentFieldBindings parameter.

```
"objectInfos" : {
  ...
  "Case" : {
    "apiName" : "Case",
    ...
    "fields" : {
      ...
      "ContactId" : {
        "apiName" : "ContactId",
        ...
        "filteredLookupInfo" : {
          "controllingFields" : [ "AccountId" ],
          "dependent" : true,
          "optionalFilter" : false
        },
        ...
      }
    }
  }
  ...
}
```

To get the value for AccountId, look in the Record response body, which is returned from several resources, including /ui-api/record-ui/{recordIds}.

```
"records" : {
  "500R0000000bKpBIAU" : {
    "apiName" : "Case",
    "childRelationships" : { },
    "fields" : {
      ...
      "AccountId" : {
        "displayValue" : null,
        "value" : "001R0000003IG0MIAW"
      },
      "CaseNumber" : {"displayValue" : null,
      "value" : "00001005" },
      ...
  }
```

To get the value for AccountId, look in the Record response body, which is returned from several resources, including /ui-api/record-ui/{recordIds}.
To filter the suggestions for a dependent lookup, specify the names and values of the controlling fields in the `dependentFieldBindings` query parameter. (This example has one controlling field).

```
/ui-api/lookups/case/ContactId?dependentFieldBindings=AccountId=001R0000003IG0MIAW
```

### Request Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>page</td>
<td>Integer</td>
<td>The page number. The default value is 1.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>pageSize</td>
<td>Integer</td>
<td>The number of items per page. The default value is 25.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>q</td>
<td>String</td>
<td>The term the user is searching for. When <code>searchType=Search</code>, specify at least 2 characters. A wildcard at the end of the search term is implied. For example, <code>q=ca</code> returns Cat and Cats. When <code>searchType=TypeAhead</code>, specify at least 3 characters. A wildcard at the end of the search term is implied. You can’t use a <code>?</code>.</td>
<td>Required if <code>searchType</code> is <code>TypeAhead</code></td>
<td>41.0</td>
</tr>
</tbody>
</table>
| searchType     | String        | The type of search to perform. One of these values:  
  - `Recent`—Return most recently used matches. | Optional           | 41.0              |
| dependentFieldBindings | String[] | The dependent field bindings for dependent lookups. These field bindings represent the lookup filter that restricts the valid values for the field. Specify field bindings in a comma-separated list in the format `dependentFieldBindings={fieldApiName}={fieldValue},{field2ApiName}={field2Value}`. To know whether a field is a dependent lookup, check the `Object Info` response body for a non-null `filteredLookupInfo` property. Specify the name and a value for each field in the `controllingFields` property. Get the field values from the `Record` response body. Both responses are returned from the `/ui-api/record-ui/{recordIds}` resource. | Optional           | 41.0              |
### Parameter Name | Type | Description | Required or Optional | Available Version
--- | --- | --- | --- | ---

- **Search**—Search for records with searchable fields that match the query term.
- **TypeAhead**—Search for records whose names start with the query term.

The default value is **Recent**.

### Response Body

- **Lookup Values**

### SEE ALSO:

- Get Lookup Field Suggestions for a Specified Object
- Get Lookup Field Actions

### Get Lookup Field Suggestions for a Specified Object

When a user edits a lookup field, use this resource to search for and display suggestions for a specified object. You can search for most recently used matches, for matching names, or for any match in a searchable field. You can also specify lookup filter bindings for dependent lookups.

**Note:** This resource is designed for future API versions. This resource filters results by object for lookup fields that have a polymorphic target object. In API version 41.0, none of the supported standard objects have a lookup field with a polymorphic target object. However, you might want to use this resource to get a simpler response. You might also want to write a general lookup component that works with future versions of User Interface API, which is likely to support more standard objects.

### Resource

```
/ui-api/lookups/{objectApiName}/{fieldApiName}/{targetApiName}
```

- **objectApiName**—The API name of a source object.
- **fieldApiName**—The API name of a lookup field on the source object.
- **targetApiName**—The API name of the target (lookup) object.

### Available Version

41.0

### HTTP Method

GET

### Example

See Get Lookup Field Suggestions.
### Request Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>dependent FieldBindings</td>
<td>String[]</td>
<td>The dependent field bindings for dependent lookups. These field bindings represent the lookup filter that restricts the valid values for the field. Specify field bindings in a comma-separated list in the format dependentFieldBindings={fieldApiName}={fieldName}, {fieldApiName}={fieldName}. To know whether a field is a dependent lookup, check the Object Info response body for a non-null filteredLookupInfo property. Specify the name and a value for each field in the controllingFields property. Get the field values from the Record response body. Both responses are returned from the /ui-api/record-ui/{recordIds} resource.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>page</td>
<td>Integer</td>
<td>The page number. The default value is 1.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>pageSize</td>
<td>Integer</td>
<td>The number of items per page. The default value is 25.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>q</td>
<td>String</td>
<td>The term the user is searching for. When searchType=Search, specify at least 2 characters. A wildcard at the end of the search term is implied. For example, q=ca returns Cat and Cats. When searchType=TypeAhead, specify at least 3 characters. A wildcard at the end of the search term is implied. You can’t use a ?.</td>
<td>Required if searchType is TypeAhead</td>
<td>41.0</td>
</tr>
</tbody>
</table>
| searchType                 | String    | The type of search to perform. One of these values:  
  - Recent—Return most recently used matches.  
  - Search—Search for records with searchable fields that match the query term. | Optional             | 41.0              |
### Actions

Use these resources to get data and metadata about actions displayed in the Salesforce UI. Examples include standard and custom buttons, quick actions, and productivity actions.

- **Get Global Actions**
  Get the actions in the Global Actions menu.
- **Get Record Detail Page Actions**
  Get the actions on record detail pages.
- **Get Record Edit Page Actions**
  Get the actions on a record edit page.
- **Get Related List Actions**
  Get the actions on related lists on a record detail page.
- **Get Related List Record Actions**
  Get the actions on records in related lists.
- **Get List View Header Actions**
  Get the header actions on list views.
- **Get List View Record Actions**
  Get the record actions on list views.
- **Get List View Chart Actions**
  Get the chart actions on list views.
- **Get Lightning Page Actions**
  Get the actions on Lightning pages (FlexiPages).
- **Get Lookup Field Actions**
  Get the actions on lookup fields.

### Parameter Name | Type | Description |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Version</td>
<td>Required or Optional</td>
<td>Description</td>
</tr>
<tr>
<td>TypeAhead</td>
<td>Search for records whose names start with the query term.</td>
<td>The default value is Recent.</td>
</tr>
</tbody>
</table>

### Response Body

Record Collection

SEE ALSO:

- Get Lookup Field Suggestions
- Get Lookup Field Actions
Get MRU List View Actions
Get the header actions on the most recently used (MRU) list view for objects.

Get Photo Actions
Get the photo actions for pages. Currently, only group and user pages support photo actions.

Get Global Actions
Get the actions in the Global Actions menu.

Resource
/ui-api/actions/global

Available version
41.0

HTTP methods
GET

Example
As an example, let’s say we have a New Opportunity global action, among other actions. To get information about global actions, make this request:

GET /services/data/v41.0/ui-api/actions/global

The request returns a list of global actions, including New Opportunity, and the URLs of the request and subrequests.

```json
{
   "actions" : {
      "001R0000003IDlwIAG" : {
         "actionListContext" : "Global",
         "actionTarget" : "/services/data/v41.0/quickActions/NewOpportunity/describe",
         "actionTargetType" : "Describe",
         "apiName" : "NewOpportunity",
         "externalId" : "00DR00000008n7M:Global::Global:Phone:09DR00000000Sn1",
         "iconUrl" : "https://yourInstance.salesforce.com/img/icon/t4v35/action/new_opportunity_120.png",
         "id" : "0JVR00000000030OAA",
         "isMassAction" : "false",
         "label" : "New Opportunity",
         "primaryColor" : "FCB95B",
         "relatedListRecordId" : null,
         "relatedSourceObject" : null,
         "section" : "Page",
         "sourceObject" : "Global",
         "subtype" : "Create",
         "targetObject" : "Opportunity",
         "targetUrl" : null,
         "type" : "QuickAction"
      }
   }
   ...
   Additional actions removed for brevity ...
}
```


## Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>CustomButton</strong>—A button that opens a URL or Visualforce page or executes JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>ProductivityAction</strong>—A pre-defined Salesforce action, attached to a limited set of objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>QuickAction</strong>—A global or object-specific action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>StandardButton</strong>—A pre-defined Salesforce button, such as New, Edit, and Delete.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Large</strong>—(Default) Use this value to get a layout for desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Medium</strong>—Use this value to get a layout for tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Small</strong>—Use this value to get a layout for phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sections</td>
<td>String[]</td>
<td>The section of the user interface that the action resides in. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>ActivityComposer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>CollaborateComposer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Page</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>SingleActionLinks</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Response body for GET

**ActionRepresentation**
Get Record Detail Page Actions

Get the actions on record detail pages.

Resource

/ui-api/actions/record/${recordIds}

recordIds—A single record ID, or a comma-delimited list of record IDs.

Available version

41.0

HTTP methods

GET

Example

As an example, let’s say we have an Account record detail page with a record ID of 001R0000003IDlwIAG. The page has some standard buttons and quick actions. To get information about all the actions on this page, make this request:

GET /services/data/v41.0/ui-api/actions/record/001R0000003IDlwIAG

The request returns a list of actions on the page and the URLs of the request and subrequests.

```json
{
  "actions": {
    "001R0000003IDlwIAG": {
      "actions": [
        {
          "actionListContext": "Record",
          "actionTarget": null,
          "actionTargetType": "Invoke",
          "apiName": "Edit",
          "externalId": "00DR00000008n7M:001R0000003IDlwIAG::Record:Phone:StandardButton:Edit",
          "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/action/edit_120.png",
          "id": "0JVR00000000017OAA",
          "isMassAction": "false",
          "label": "Edit",
          "primaryColor": "1DCCBF",
          "relatedListRecordId": null,
          "relatedSourceObject": null,
          "section": "Page",
          "sourceObject": "001R0000003IDlwIAG",
          "subtype": null,
          "targetObject": "Account",
          "targetUrl": null,
          "type": "StandardButton"
        },
        {
          "actionListContext": "Record",
          "actionTarget": "/services/data/v41.0/quickActions/NewTask/describe",
          "actionTargetType": "Describe",
          "apiName": "NewTask",
          "externalId": "00DR00000000Sns:001R0000003IDlwIAG::Record:Phone:09DR0000000Sns",
          "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/action/new_task_120.png",
```
The action type. One of these values:

- **CustomButton**—A button that opens a URL or Visualforce page or executes JavaScript.
- **ProductivityAction**—A pre-defined Salesforce action, attached to a limited set of objects.
- **QuickAction**—A global or object-specific action.
- **StandardButton**—A pre-defined Salesforce button, such as **New**, **Edit**, and **Delete**.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
</tbody>
</table>

- Large—(Default) Use this value to get a layout for desktop display size.
### User Interface API Resources

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
<td>- Medium—Use this value to get a layout for tablet display size.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Small—Use this value to get a layout for phone display size.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>sections</td>
<td>String[]</td>
<td>The section of the user interface that the action resides in. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ActivityComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CollaborateComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SingleActionLinks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Response body for GET**

ActionRepresentation

### Get Record Edit Page Actions

Get the actions on a record edit page.

**Resource**

```
/ui-api/actions/record/${recordId}/record-edit
```

**Available version**

41.0

**HTTP methods**

GET

**Example**

As an example, let’s say we have an Account page with a record ID of 001R0000031DlwIAG. To get information about all the actions on the edit page for this record, make this request:

```
GET /services/data/v41.0/ui-api/actions/record/001R0000031DlwIAG/record-edit
```

The request returns a list of actions on the edit page and the URLs of the request and subrequests.

```
{
    "actions": {
        "001R0000031DlwIAG": {
            "actions": [
                {
                    "actionListContext": "RecordEdit",
                    "actionTarget": null,
                    "actionTargetType": "Invoke",
                    "apiName": "SaveEdit",
                    "externalId": "00DR0000008n7M:001R0000031DlwIAG::RecordEdit:Phone:StandardButton:SaveEdit",
                    "iconUrl":
```

76
Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CustomButton—A button that opens a URL or Visualforce page or executes JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ProductivityAction—A pre-defined Salesforce action, attached to a limited set of objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• QuickAction—A global or object-specific action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• StandardButton—A pre-defined Salesforce button, such as New, Edit, and Delete.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—(Default) Use this value to get a layout for desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—Use this value to get a layout for tablet display size.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Get Related List Actions

Get the actions on related lists on a record detail page.

Resources

`/ui-api/actions/record/${recordId}/related-list/${relatedListIds}`

- **relatedListIds** — A related list ID, or a comma-delimited list of related list IDs.

**Note:** ${relatedListId} is optional. The request returns all related list actions when ${relatedListIds} isn’t supplied.

**Available version**

41.0

**HTTP methods**

GET

**Example**

As an example, let’s say we have an Account record detail page with a record ID of 001R0000003IDlwIAG. The page has some related lists, including tasks and events. To get information about all the related list actions on this page, make this request:

GET /services/data/v41.0/ui-api/actions/record/001R0000003IDlwIAG/related-list

The request returns a list of related list actions on the page and the URLs of the request and subrequests.

```json
{
    "actions" : {
        "001R0000003IDlwIAG" : {
            "actions" : [ {
                "actionListContext" : "RelatedList",
                "actionTarget" : null,
                "actionTargetType" : "Invoke",
                "apiName" : "NewTask",
                "externalId" : "00DR0000008n7M:001R0000003IDlwIAG::RelatedList:Phone:StandardButton:NewTask",
```
### Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CustomButton—A button that opens a URL or Visualforce page or executes JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ProductivityAction—A pre-defined Salesforce action, attached to a limited set of objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• QuickAction—A global or object-specific action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• StandardButton—A pre-defined Salesforce button, such as New, Edit, and Delete.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—(Default) Use this value to get a layout for desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—Use this value to get a layout for tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—Use this value to get a layout for phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sections</td>
<td>String[]</td>
<td>The section of the user interface that the action resides in. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ActivityComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CollaborateComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SingleActionLinks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Response body for GET

**ActionRepresentation**

### Get Related List Record Actions

Get the actions on records in related lists.

**Resource**

`/ui-api/actions/record/${recordId}/related-list-record/${relatedListRecordIds}`

- relatedListRecordIds—A related list record ID, or a comma-delimited list of related list record IDs.
Available version
41.0

HTTP methods
GET

Example

As an example, let’s say we have an Account record detail page with a record ID of 001R0000003IDlwIAG. The page has a Contacts related list, and one of the Contacts has a record ID of 003R0000007qfC7. To get information about the actions for this contact, make this request:

GET /services/data/v41.0/ui-api/actions/record/001R0000003IDlwIAG/related-list-record/003R0000007qfC7

The request returns a list of actions for the contact and the URLs of the request and subrequests.

```json
{
    "actionListContext" : "RelatedListRecord",
    "actionTarget" : null,
    "actionTargetType" : "Invoke",
    "apiName" : "Delete",
    "externalId" : "00DR00000008n7M:001R0000003IDlwIAG::RelatedListRecord:Phone:StandardButton:Delete",
    "iconUrl" : "https://yourInstance.salesforce.com/img/icon/t4v35/action/delete_120.png",
    "id" : "0JVR00000000014OAA",
    "isMassAction" : "false",
    "label" : "Delete",
    "primaryColor" : "E6717C",
    "relatedListRecordId" : "003R0000007qfC7",
    "relatedSourceObject" : "Contacts",
    "section" : "Page",
    "sourceObject" : "001R0000003IDlwIAG",
    "subtype" : null,
    "targetObject" : "Contact",
    "targetUrl" : null,
    "type" : "StandardButton"
}

... Additional actions removed for brevity ...
```

```json
],

    "links" : [ ],

    "url" : "/services/data/v41.0/ui-api/actions/record/001R0000003IDlwIAG/related-list-record/003R0000007qfC7"
}
}

"url" : "/services/data/v41.0/ui-api/actions/record/001R0000003IDlwIAG/related-list-record/003R0000007qfC7"
```
Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CustomButton—A button that opens a URL or Visualforce page or executes JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ProductivityAction—A pre-defined Salesforce action, attached to a limited set of objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• QuickAction—A global or object-specific action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• StandardButton—A pre-defined Salesforce button, such as New, Edit, and Delete.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—(Default) Use this value to get a layout for desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—Use this value to get a layout for tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—Use this value to get a layout for phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sections</td>
<td>String[]</td>
<td>The section of the user interface that the action resides in. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ActivityComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CollaborateComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SingleActionLinks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Response body for GET

ActionRepresentation

Get List View Header Actions

Get the header actions on list views.

Resource

/ui-api/actions/list-view/${listViewIds}

listViewIds—A single list view ID, or a comma-delimited list of list view IDs.
Available version
41.0

HTTP methods
GET

Example
As an example, let’s say we have an Accounts list view with an ID of 00BR0000000WYY9MAO. This list view has Sort and New header actions, among other actions. To get information about the header actions on this list view, make this request:

GET /services/data/v41.0/ui-api/actions/list-view/00BR0000000WYY9MAO

The request returns a list of header actions on the list view and the URLs of the request and subrequests.

```json
{
  "actions": {
    "00BR0000000vL8yMAE": {
      "actions": [{
        "actionListContext": "ListView",
        "actionTarget": null,
        "actionTargetType": "Invoke",
        "apiName": "EditFilter",
        "externalId": "00DR0000000978H:00BR0000000vL8yMAE::ListView:Desktop:StandardButton:EditFilter",
        "iconUrl": "https://mobile1.t.salesforce.com/img/icon/t4v35/action/filter_120.png",
        "id": "0JVR00000021485OAA",
        "isMassAction": "false",
        "label": "Filter",
        "primaryColor": "54698d",
        "relatedListRecordId": null,
        "relatedSourceObject": null,
        "section": "Page",
        "sourceObject": "00BR0000000vL8yMAE",
        "subtype": null,
        "targetObject": "Account",
        "targetUrl": null,
        "type": "StandardButton"
      }, {
        "actionListContext": "ListView",
        "actionTarget": null,
        "actionTargetType": "Invoke",
        "apiName": "ListSort",
        "externalId": "00DR0000000978H:00BR0000000vL8yMAE::ListView:Desktop:StandardButton:ListSort",
        "iconUrl": "https://mobile1.t.salesforce.com/img/icon/t4v35/action/sort_120.png",
        "id": "0JVR00000021486OAA",
        "isMassAction": "false",
        "label": "Sort",
        "primaryColor": "FAB9A5",
        "relatedListRecordId": null,
        "relatedSourceObject": null,
        "section": "Page",
        "sourceObject": "00BR0000000vL8yMAE",
        "subtype": null
      }]
    }
  }
}
```
Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CustomButton—A button that opens a URL or Visualforce page or executes JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ProductivityAction—A pre-defined Salesforce action, attached to a limited set of objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• QuickAction—A global or object-specific action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• StandardButton—A pre-defined Salesforce button, such as New, Edit, and Delete.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—(Default) Use this value to get a layout for desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—Use this value to get a layout for tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—Use this value to get a layout for phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sections</td>
<td>String[]</td>
<td>The section of the user interface that the action resides in. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ActivityComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CollaborateComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SingleActionLinks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Response body for GET

**ActionRepresentation**

Get List View Record Actions

Get the record actions on list views.

**Resource**

```
/ui-api/actions/list-view-record/${recordIds}
```

`recordIds`—A single record ID, or a comma-delimited list of record IDs.
Available version
41.0

HTTP methods
GET

Example

As an example, let's say we have an Account lists view containing several Account records. Each Account record has an Edit and Delete record action. In Lightning, these actions display in a drop-down list for each record. To get information about the actions on a particular record with an ID of 001R0000003IDlwIAG, make this request:

GET /services/data/v41.0/ui-api/actions/list-view-record/001R0000003IDlwIAG

The request returns a list of the record actions for the record and the URLs of the request and subrequests.

```json
{
  "actions": {
    "001R0000003IDlwIAG": {
      "actions": [
        {
          "actionListContext": "ListViewRecord",
          "actionTarget": null,
          "actionTargetType": "Invoke",
          "apiName": "Edit",
          "externalId": "00DR00000008n7M:001R0000003IDlwIAG::ListViewRecord:Phone:StandardButton:Edit",
          "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/action/edit_120.png",
          "id": "0JVR0000000660AA",
          "isMassAction": "false",
          "label": "Edit",
          "primaryColor": "1DCCBF",
          "relatedListRecordId": null,
          "relatedSourceObject": null,
          "section": "Page",
          "sourceObject": "001R0000003IDlwIAG",
          "subtype": null,
          "targetObject": "Account",
          "targetUrl": null,
          "type": "StandardButton"
        },
        {
          "actionListContext": "ListViewRecord",
          "actionTarget": null,
          "actionTargetType": "Invoke",
          "apiName": "Delete",
          "externalId": "00DR00000008n7M:001R0000003IDlwIAG::ListViewRecord:Phone:StandardButton:Delete",
          "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/action/delete_120.png",
          "id": "0JVR0000000670AA",
          "isMassAction": "false",
          "label": "Delete",
          "primaryColor": "E6717C",
          "relatedListRecordId": null,
          "relatedSourceObject": null,
          "section": "Page"
        }
      ]
    }
  }
}```
... Additional actions removed for brevity ...

],
  "links" : [ ],
  "url" : "/services/data/v41.0/ui-api/actions/list-view-record/001R0000003IDlwIAG"
}]
],
  "url" : "/services/data/v41.0/ui-api/actions/list-view-record/001R0000003IDlwIAG"
}

### Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CustomButton—A button that opens a URL or Visualforce page or executes JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ProductivityAction—A pre-defined Salesforce action, attached to a limited set of objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• QuickAction—A global or object-specific action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• StandardButton—A pre-defined Salesforce button, such as New, Edit, and Delete.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—(Default) Use this value to get a layout for desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—Use this value to get a layout for tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—Use this value to get a layout for phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sections</td>
<td>String[]</td>
<td>The section of the user interface that the action resides in. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ActivityComposer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Available Version

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Available Version</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HTTP methods</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get List View Chart Actions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get the chart actions on list views.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ActionRepresentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response body for GET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: Currently, only ListViewChartInstance is valid for objectApiName.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Available version</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HTTP methods</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This example gets the chart actions on a list view.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GET /services/data/v41.0/ui-api/actions/list-view-chart/ListViewChartInstance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The request returns a list of chart actions on the list view and the URLs of the request and subrequests.</td>
</tr>
</tbody>
</table>

```json
{
  "actions": {
    "ListViewChartInstance": {
      "actions": [
        {
          "actionListContext": "ObjectHomeChart",
          "actionTarget": null,
          "actionTargetType": null,
          "apiName": "NewObjectHomeChartAction",
          "externalId": "00DR00000008n7M:ListViewChartInstance::ObjectHomeChart:Phone:StandardButton:NewObjectHomeChartAction",
          "iconUrl": "https://yourinstance.salesforce.com/img/icon/t4v35/action/new_custom19_120.png",
          "id": "0JVR00000000001OAA",
          "isMassAction": "false",
          "label": "New Chart",
          "primaryColor": null,
          "relatedListRecordId": null,
          "relatedSourceObject": null,
          "section": "Page"
        }
      ]
    }
  }
}
```
Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>CustomButton</strong>—A button that opens a URL or Visualforce page or executes JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>ProductivityAction</strong>—A pre-defined Salesforce action, attached to a limited set of objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>QuickAction</strong>—A global or object-specific action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>StandardButton</strong>—A pre-defined Salesforce button, such as <strong>New</strong>, <strong>Edit</strong>, and <strong>Delete</strong>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Large</strong>—(Default) Use this value to get a layout for desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Medium</strong>—Use this value to get a layout for tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Small</strong>—Use this value to get a layout for phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sections</td>
<td>String[]</td>
<td>The section of the user interface that the action resides in. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>ActivityComposer</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Parameter Name

- CollaborateComposer
- Page
- SingleActionLinks

<table>
<thead>
<tr>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>41.0</td>
</tr>
</tbody>
</table>

### Response body for GET

ActionRepresentation

### Get Lightning Page Actions

Get the actions on Lightning pages (FlexiPages).

#### Resource

```
/ui-api/actions/flexipage/${flexipageNames}
```

*flexipageNames*—A Lightning page name, or a comma-delimited list of Lightning page names.

#### Available version

41.0

#### HTTP methods

GET

#### Example

As an example, let’s say we have a Lightning page named *MyNewLightningPage*. The page has quick actions for logging a call and creating an account. To get information about all the actions on this page, make this request:

```
GET /services/data/v41.0/ui-api/actions/flexipage/MyNewLightningPage
```

The request returns a list of actions on the Lightning page and the URLs of the request and subrequests.

```json
{
   "actions" : {
      "MyNewLightningPage" : {
         "actions" : [ {
            "actionListContext" : "Flexipage",
            "actionTarget" : "/services/data/v41.0/quickActions/LogACall/describe",
            "actionTargetType" : "Describe",
            "apiName" : "LogACall",
            "externalId" : "00DR00000008n7M:MyNewLightningPage::Flexipage:Desktop:09DR00000000Snt",
            "iconUrl" : "https://yourInstance.salesforce.com/img/icon/t4v35/action/log_a_call_120.png",
            "id" : "0JVR00000000009OAA",
            "isMassAction" : "false",
            "label" : "Log a Call",
            "primaryColor" : "48C3CC",
            "relatedListRecordId" : null,
            "relatedSourceObject" : null,
            "section" : null,
```

90
Get Lightning Page Actions

Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>CustomButton</strong>—A button that opens a URL or Visualforce page or executes JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>ProductivityAction</strong>—A pre-defined Salesforce action, attached to a limited set of objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>QuickAction</strong>—A global or object-specific action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>StandardButton</strong>—A pre-defined Salesforce button, such as New, Edit, and Delete.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Get Lookup Field Actions

Get the actions on lookup fields.

**Resource**

```
/ui-api/actions/lookup/${objectApiNames}
```

`objectApiNames`—A lookup field name, or a comma-delimited list of lookup field names.

**Available version**

41.0

**HTTP methods**

GET

**Example**

As an example, let’s say we have a lookup field on a New Account dialog box. To get information about the actions on this lookup field, make this request:

```
GET /services/data/v41.0/ui-api/actions/lookup/Account
```

The request returns a list of actions on the lookup field and the URLs of the request and subrequests.

```json
{
    "actions" : {
        "Account" : {
            "actions" : [ {
```
Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CustomButton—A button that opens a URL or Visualforce page or executes JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ProductivityAction—A pre-defined Salesforce action, attached to a limited set of objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• QuickAction—A global or object-specific action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• StandardButton—A pre-defined Salesforce button, such as New, Edit, and Delete.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—(Default) Use this value to get a layout for desktop display size.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Get MRU List View Actions

Get the header actions on the most recently used (MRU) list view for objects.

**Resource**

```
/ui-api/actions/mru-list/${objectApiNames}
```

- `objectApiNames`—An object name, or a comma-delimited list of object names.

**Available version**

41.0

**HTTP methods**

GET

**Example**

As an example, let’s say we have a most recently used Accounts list view. To get information about the header actions on this list view, make this request:

```
GET /services/data/v41.0/ui-api/actions/mru-list/Account
```

The request returns a list of header actions on the list view and the URLs of the request and subrequests.

```json
{
    "actions" : {
        "Account" : {
            "actions" : [ { ...
```
Get MRU List View Actions

```
"actionListContext" : "MruList",
"actionTarget" : null,
"actionTargetType" : "Invoke",
"apiName" : "New",
"externalId" : "00DR00000008n7M:Account::MruList:Phone:StandardButton:New",
"iconUrl" : "https://yourInstance.salesforce.com/img/icon/t4v35/action/new_120.png",
"id" : "0JVR00000000102OAA",
"isMassAction" : "false",
"label" : "New",
"primaryColor" : "33BCE7",
"relatedListRecordId" : null,
"relatedSourceObject" : null,
"section" : "Page",
"sourceObject" : "Account",
"subtype" : null,
"targetObject" : "Account",
"targetUrl" : null,
"type" : "StandardButton",
},

"actionListContext" : "MruList",
"actionTarget" : null,
"actionTargetType" : null,
"apiName" : "MruListSort",
"externalId" : "00DR00000008n7M:Account::MruList:Phone:StandardButton:MruListSort",
"iconUrl" : "https://yourInstance.salesforce.com/img/icon/t4v35/action/sort_120.png",
"id" : "0JVR00000000103OAA",
"isMassAction" : "false",
"label" : "Sort",
"primaryColor" : "FAB9A5",
"relatedListRecordId" : null,
"relatedSourceObject" : null,
"section" : "Page",
"sourceObject" : "Account",
"subtype" : null,
"targetObject" : "Account",
"targetUrl" : null,
"type" : "StandardButton"
}

... Additional actions removed for brevity ...

],
"links" : [ ],
"uri" : "/services/data/v41.0/ui-api/actions/mru-list/Account"
}
`,
"uri" : "/services/data/v41.0/ui-api/actions/mru-list/Account"
}
### Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
</table>
| actionTypes    | String[] | The action type. One of these values:  
  - CustomButton—A button that opens a URL or Visualforce page or executes JavaScript.  
  - ProductivityAction—A pre-defined Salesforce action, attached to a limited set of objects.  
  - QuickAction—A global or object-specific action.  
  - StandardButton—A pre-defined Salesforce button, such as New, Edit, and Delete. | Optional | 41.0 |
| formFactor     | String   | The layout display size. One of these values:  
  - Large—(Default) Use this value to get a layout for desktop display size.  
  - Medium—Use this value to get a layout for tablet display size.  
  - Small—Use this value to get a layout for phone display size. | Optional | 41.0 |
| sections       | String[] | The section of the user interface that the action resides in. One of these values:  
  - ActivityComposer  
  - CollaborateComposer  
  - Page  
  - SingleActionLinks | Optional | 41.0 |

### Response body for GET

ActionRepresentation

### Get Photo Actions

Get the photo actions for pages. Currently, only group and user pages support photo actions.

### Resource

`/ui-api/actions/photo/{recordIds}

recordIds—A single record ID, or a comma-delimited list of record IDs.`
Available version
41.0

HTTP methods
GET

Example
As an example, let’s say we have a Group tab with a record ID of 0F9R00000000182KAA. To get information about the photo actions on the tab, make this request:

```
GET /services/data/v41.0/ui-api/actions/photo/0F9R00000000182KAA
```

The request returns a list of photo actions on the tab and the URLs of the request and subrequests.

```json
{
  "actions": {
    "0F9R00000000182KAA": {
      "actions": [
        {
          "actionListContext": "Photo",
          "actionTarget": null,
          "actionTargetType": "Invoke",
          "apiName": "ViewPhotoAction",
          "externalId": "00DR00000008n7M:0F9R00000000182KAA::Photo:Phone:StandardButton:ViewPhotoAction",
          "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/action/email_120.png",
          "id": "0JVR00000000035OAA",
          "isMassAction": "false",
          "label": "View Photo",
          "primaryColor": "95AEC5",
          "relatedListRecordId": null,
          "relatedSourceObject": null,
          "section": "Page",
          "sourceObject": "0F9R00000000182KAA",
          "subtype": null,
          "targetObject": null,
          "targetUrl": null,
          "type": "StandardButton"
        }
      ],
      "links": [],
      "url": "/services/data/v41.0/ui-api/actions/photo/0F9R00000000182KAA"
    }
  }
}
```

... Additional actions removed for brevity ...

```json
}
```

**Request parameters for GET**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionTypes</td>
<td>String[]</td>
<td>The action type. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CustomButton—A button that opens a URL or Visualforce page or executes JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ProductivityAction—A pre-defined Salesforce action, attached to a limited set of objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• QuickAction—A global or object-specific action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• StandardButton—A pre-defined Salesforce button, such as New, Edit, and Delete.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The layout display size. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—(Default) Use this value to get a layout for desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—Use this value to get a layout for tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—Use this value to get a layout for phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sections</td>
<td>String[]</td>
<td>The section of the user interface that the action resides in. One of these values:</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ActivityComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CollaborateComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SingleActionLinks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Response body for GET**

*ActionRepresentation*

**Favorites**

Use these resources to create and update a single favorite or a collection of favorites, and to get favorites usage information. These resources make it easy to build UI for favorites.
Create a Favorite
Create a favorite.

Get a Favorite
Get a favorite.

Get Favorites
Get all of a user’s favorites.

Update a Favorite
Update a favorite.

Update a Batch of Favorites
Update all favorites at once. The sort order is updated to the given relative ordering. Any favorites missing from the request body are deleted.

Delete a Favorite
Delete a favorite.

Update Usage of a Favorite
Update the usage of an individual favorite, for example, the last time and number of times the favorite was clicked.

Create a Favorite
Create a favorite.

Resource
/ui-api/favorites

Available Version
41.0

HTTP Method
POST

Request Body

JSON example

```json
{
  name: "Most Important Accounts",
  sortOrder: 1,
  target: "00BR0000000tTTwMAM",
  targetType: "ListView"
}
```

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the favorite.</td>
<td>You must specify either name or sortOrder. You may specify both.</td>
<td>41.0</td>
</tr>
</tbody>
</table>
**Available Version**

**Required or Optional**

**Description**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>sortOrder</td>
<td>Integer</td>
<td>The sort order of the favorite, from 1 to N.</td>
<td>41.0</td>
</tr>
<tr>
<td>target</td>
<td>String</td>
<td>The record, API name, or content being favorited.</td>
<td>41.0</td>
</tr>
<tr>
<td>targetType</td>
<td>String</td>
<td>The type of favorite. One of these values:</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ListView—A favorited list view.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ObjectHome—A favorited object home.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Record—A favorited record.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tab—A favorited tab.</td>
<td></td>
</tr>
</tbody>
</table>

**Request Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the favorite.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>sortOrder</td>
<td>Integer</td>
<td>The sort order of the favorite, from 1 to N.</td>
<td>Optional</td>
<td>41.0</td>
</tr>
<tr>
<td>target</td>
<td>String</td>
<td>The record, API name, or content being favorited.</td>
<td>Required</td>
<td>41.0</td>
</tr>
<tr>
<td>targetType</td>
<td>String</td>
<td>The type of favorite. One of these values:</td>
<td>Required</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ListView—A favorited list view.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ObjectHome—A favorited object home.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Record—A favorited record.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tab—A favorited tab.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Response Body for POST**

**Favorite**

**Get a Favorite**

Get a favorite.
Resource

/ui-api/favorites/${favoriteId}

favoriteId—The ID of a favorite.

Available Version

41.0

HTTP Method

GET

Response Body

FavoriteRepresentation on page 152

Get Favorites

Get all of a user’s favorites.

Resource

/ui-api/favorites

Available Version

41.0

HTTP Method

GET

Response Body

Favorite Collection

Update a Favorite

Update a favorite.

Resource

/ui-api/favorites/${favoriteId}

favoriteId—The ID of a favorite.

Available Version

41.0

HTTP Method

PATCH

Example

This example uses a request body to update a favorite.

```json
{
  name: "Red Accounts",
  sortOrder: 1
}
```
## Request Body

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the favorite.</td>
<td>You must specify either name or sortOrder. You may specify both.</td>
<td>41.0</td>
</tr>
<tr>
<td>sortOrder</td>
<td>Integer</td>
<td>The sort order of the favorite, from 1 to N.</td>
<td>You must specify either name or sortOrder. You may specify both.</td>
<td>41.0</td>
</tr>
<tr>
<td>target</td>
<td>String</td>
<td>The record, API name, or content being favorited.</td>
<td>Required in a POST request. Do not specify in a PATCH request.</td>
<td>41.0</td>
</tr>
<tr>
<td>targetType</td>
<td>String</td>
<td>The type of favorite. One of these values:</td>
<td>Required in a POST request. Do not specify in a PATCH request.</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ListView—A favorited list view.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ObjectHome—A favorited object home.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Record—A favorited record.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tab—A favorited tab.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Request Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the favorite.</td>
<td>You must specify either name or sortOrder. You may specify both.</td>
<td>41.0</td>
</tr>
<tr>
<td>sortOrder</td>
<td>Integer</td>
<td>The sort order of the favorite, from 1 to N.</td>
<td>You must specify either name or sortOrder. You may specify both.</td>
<td>41.0</td>
</tr>
</tbody>
</table>

## Response Body

FavoriteRepresentation
Update a Batch of Favorites

Update all favorites at once. The sort order is updated to the given relative ordering. Any favorites missing from the request body are deleted.

Resource

```
/ui-api/favorites/batch
```

Available Version

41.0

HTTP Method

PUT

Request Body

**JSON example**

```
{
  favorites: [
    {
      id: "0MVR00000004DhnOAE",
      name: "Q4 Perf"
    },
    {
      id: "0MVR00000004DhsOAE"
    },
    {
      id: "0MVR00000004DiGOAU"
    },
    {
      id: "0MVR00000001e2OAA",
      name: "Office Group"
    },
    {
      id: "0MVR00000004GGlOAM"
    }
  ]
}
```

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>favorites</td>
<td>Favorite Batch Input[]</td>
<td>The list of favorites to keep and update.</td>
<td>Required</td>
<td>41.0</td>
</tr>
</tbody>
</table>

Response Body

Favorite Collection

Delete a Favorite

Delete a favorite.

Resource

```
/ui-api/favorites/${favoriteId}
```

**favoriteId**—The ID of a favorite.
Available Version
41.0
HTTP Method
DELETE
Response Body
Deleting a record returns a 200 HTTP status code.

Update Usage of a Favorite
Update the usage of an individual favorite, for example, the last time and number of times the favorite was clicked.

Resource
/get-api/favorites/${favoriteId}/usage

favoriteId—The ID of a favorite.

Available Version
41.0
HTTP Method
PATCH
Response Body
Favorite

List Views
Use these resources to get data and metadata about list views displayed in the Salesforce UI.

Get List Views for an Object
Returns a collection of list views associated with an object.

Get List View Records
Returns record data for a list view.

Get List View Metadata
Returns list view metadata.

Get List View Records and Metadata
Returns record data and metadata for a list view.

Get Most Recently Used List View Records
Returns record data for an object’s most recently used (MRU) list view.

Get Most Recently Used List View Metadata
Returns metadata for an object’s most recently viewed (MRU) list view.

Get Most Recently Used List View Records and Metadata
Returns record data and metadata for an object’s most recently used (MRU) list view.
Get List Views for an Object

Returns a collection of list views associated with an object.

Resource

```
/ui-api/list-ui/${objectApiName}
```

- `objectApiName`—A supported object, such as Account.

Available version

43.0

HTTP methods

GET

Example

To retrieve all list views associated with accounts, make this request:

GET /services/data/v47.0/ui-api/list-ui/Account

The request returns a list of all list views for the given object.

```json
{
  "count" : 6,
  "currentPageToken" : "0",
  "currentPageUrl" : "/services/data/v43.0/ui-api/list-ui/Account?pageSize=20&pageToken=0",
  "eTag" : "d33d58077e8560bcaec18bdc2647c642",
  "lists" : [ {
    "apiName" : "AllAccounts",
    "id" : "00BRM000002KPM42AO",
    "label" : "All Accounts",
    "listUiUrl" : "/services/data/v43.0/ui-api/list-ui/Account/AllAccounts"
  }, {
    "apiName" : "MyAccounts",
    "id" : "00BRM000002KPM2A4",
    "label" : "My Accounts",
    "listUiUrl" : "/services/data/v43.0/ui-api/list-ui/Account/MyAccounts"
  }, {
    "apiName" : "NewLastWeek",
    "id" : "00BRM000002KPMJ2A0",
    "label" : "New Last Week",
    "listUiUrl" : "/services/data/v43.0/ui-api/list-ui/Account/NewLastWeek"
  }, {
    "apiName" : "NewThisWeek",
    "id" : "00BRM000002KPMQ2A4",
    "label" : "New This Week",
    "listUiUrl" : "/services/data/v43.0/ui-api/list-ui/Account/NewThisWeek"
  }, {
    "apiName" : "PlatinumandGoldSLACustomers",
    "id" : "00BRM000002KPMg2AO",
    "label" : "Platinum and Gold SLA Customers",
    "listUiUrl" : "/services/data/v43.0/ui-api/list-ui/Account/PlatinumandGoldSLACustomers"
  } ],
  "nextPageToken" : "0",
```
"id" : "00BRM000002KPM22AO",
"label" : "Recently Viewed Accounts",
"listUiUrl" : "services/data/v43.0/ui-api/list-ui/Account/RecentlyViewedAccounts"
}
,

"nextPageToken" : null,
"nextPageUrl" : null,
"previousPageToken" : null,
"previousPageUrl" : null
}

Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>pageSize</td>
<td>String</td>
<td>The number of list records viewed at one time. The default value is 20. Value can be 1–2000.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>pageToken</td>
<td>String</td>
<td>A token that represents the page offset. To indicate where the page starts, use this value with the pageSize parameter. The maximum offset is 2000 and the default is 0.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>q</td>
<td>String</td>
<td>Query list views that contain certain key words.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>recentListsOnly</td>
<td>Boolean</td>
<td>Show only recently viewed lists.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
</tbody>
</table>

Response body for GET

List View Summary Collection

Get List View Records

Returns record data for a list view.

Resources

/ui-api/list-records/${listViewId}

/ui-api/list-records/${objectApiName}/${listViewApiName}

- listViewId—The ID of a list view, such as 00BR0000000Wc0rMAC.
- objectApiName—A supported object, such as Account.
- listViewApiName—The API name of a list view, such as AllAccounts.

Available version

42.0

HTTP methods

GET
Example

As an example, let's say we have a list view named All Accounts with a list view ID of 00BR0000000Wc0rMAC. To get information about all the records on this list view, make this request:

/services/data/v47.0/ui-api/list-records/00BR0000000Wc0rMAC

You can also use the object API name and list view API name of the list view to make the same request:

/services/data/v47.0/ui-api/list-records/Account/AllAccounts

The request returns a list of records on the list view.

```json
{
  "count" : 12,
  "currentPageToken" : "0",
  "currentPageUrl" : 
    "/services/data/v43.0/ui-api/list-records/Account/AllAccounts?pageSize=50&pageToken=0",
  "listInfoETag" : "f5f522ae3f88a0cb38be54b9c0095fe3c",
  "nextPageToken" : null,
  "nextPageUrl" : null,
  "previousPageToken" : null,
  "previousPageUrl" : null,
  "records" : [
    {
      "apiName" : "Account",
      "childRelationships" : { }
    },
    {
      "BillingState" : {
        "displayValue" : null,
        "value" : "NC"
      },
      "CreatedDate" : {
        "displayValue" : null,
        "value" : "2018-04-11T00:37:56.000Z"
      },
      "Id" : {
        "displayValue" : null,
        "value" : "001RM000003UNtxYAG"
      },
      "LastModifiedDate" : {
        "displayValue" : null,
        "value" : "2018-04-11T00:37:56.000Z"
      },
      "Name" : {
        "displayValue" : null,
        "value" : "Burlington Textiles Corp of America"
      },
      "Owner" : {
        "displayValue" : null,
        "value" : {
          "apiName" : "User",
          "childRelationships" : { }
        }
      }
    }
  ]
}
```
"displayValue": null,
"value": "2018-04-11T00:37:56.000Z"
},
"Name": {
"displayValue": null,
"value": "Dickenson plc"
},
"Owner": {
"displayValue": null,
"value": {
"apiName": "User",
"childRelationships": { },
"eTag": "671bc4877d13cd4f43d28a671636f873",
"fields": {
"Alias": {
"displayValue": null,
"value": "AUUser"
},
"Id": {
"displayValue": null,
"value": "005RM000001cNuJYAU"
}
},
"id": "005RM000001cNuJYAU",
"recordTypeInfo": null
}
},
"OwnerId": {
"displayValue": null,
"value": "005RM000001cNuJYAU"
},
"Phone": {
"displayValue": null,
"value": "(785) 241-6200"
},
"Site": {
"displayValue": null,
"value": null
},
"SystemModstamp": {
"displayValue": null,
"value": "2018-04-11T00:37:56.000Z"
},
"Type": {
"displayValue": "Customer - Channel",
"value": "Customer - Channel"
}
},
"id": "001RM000003UNtzYAG",
"recordTypeInfo": null
},
{ ... Additional records removed for brevity ... ]
Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>fields</td>
<td>String[]</td>
<td>Additional fields queried for the records returned. These fields don’t create visible columns. If the field is not available to the user, an error occurs.</td>
<td>Optional</td>
<td>42.0</td>
</tr>
<tr>
<td>optionalFields</td>
<td>String[]</td>
<td>Additional fields queried for the records returned. These fields don’t create visible columns. If the field is not available to the user, an error occurs.</td>
<td>Optional</td>
<td>42.0</td>
</tr>
<tr>
<td>pageSize</td>
<td>String</td>
<td>The number of list records viewed at one time. The default value is 50. Value can be 1–2000.</td>
<td>Optional</td>
<td>42.0</td>
</tr>
<tr>
<td>pageToken</td>
<td>String</td>
<td>A token that represents the page offset. To indicate where the page starts, use this value with the pageSize parameter. The maximum offset is 2000 and the default is 0.</td>
<td>Optional</td>
<td>42.0</td>
</tr>
<tr>
<td>sortBy</td>
<td>String</td>
<td>The API name of the field the list view is sorted by. If the name is preceded with “-“, the sort order is descending. For example, &quot;Name&quot; sorts by name in ascending order. &quot;-CreatedDate&quot; sorts by created date in descending order.</td>
<td>Optional</td>
<td>42.0</td>
</tr>
</tbody>
</table>

Response body for GET

List Record Collection

Get List View Metadata

Returns list view metadata.

Resource

```
/ui-api/list-info/${listViewId}
```

```
/ui-api/list-info/${objectApiName}/${listViewApiName}
```

- **listViewId**—The ID of a list view, such as 00BR000000Wc0rMAC.
- **objectApiName**—A supported object, such as Account.
- **listViewApiName**—The API name of a list view, such as AllAccounts.
Available version
42.0

HTTP methods
GET

Example
As an example, let’s say we have a list view named All Accounts with a list view ID of 00BR0000000Wc0rMAC. To get information about the metadata for this list view, make this request:

/services/data/v47.0/ui-api/list-info/00BR0000000Wc0rMAC

You can also use the object API name and list view API name of the list view to make the same request:

/services/data/v47.0/ui-api/list-info/Account/AllAccounts

The request returns the metadata for the list view.

```json
{
   "cloneable": true,
   "createable": true,
   "deletable": true,
   "displayColumns": [
      {
         "fieldApiName": "Name",
         "label": "Account Name",
         "sortable": true
      },
      {
         "fieldApiName": "BillingState",
         "label": "Billing State/Province",
         "sortable": true
      },
      {
         "fieldApiName": "Phone",
         "label": "Phone",
         "sortable": true
      },
      {
         "fieldApiName": "Type",
         "label": "Type",
         "sortable": true
      },
      {
         "fieldApiName": "Owner.Alias",
         "label": "Account Owner Alias",
         "sortable": true
      }
   ],
   "eTag": "86d3a33edd84d9889b38b102c915abf",
   "filterLogicString": null,
   "filteredByInfo": [],
   "id": {
      "id": "00BR0000000Wc0rMAC",
      "objectApiName": "Account",
      "type": "listView"
   },
   "label": "All Accounts",
   "listViewApiName": "AllAccounts",
   "orderedByInfo": []
}
```
### Response body for GET

**List Metadata**

#### Get List View Records and Metadata

Returns record data and metadata for a list view.

**Resources**

- `/ui-api/list-ui/${listViewId}`
- `/ui-api/list-ui/${objectApiName}/${listViewApiName}`

- **listViewId**—The ID of a list view, such as `00BR0000000Wc0rMAC`.
- **objectApiName**—A [supported](https://example.com) object, such as `Account`.
- **listViewApiName**—The API name of a list view, such as `AllAccounts`.

**Available version**

42.0

**HTTP methods**

GET

**Example**

As an example, let's say we have a list view named All Accounts with a list view ID of `00BR0000000Wc0rMAC`. To get information about all the records and metadata for this list view, make this request.

```
/services/data/v47.0/ui-api/list-ui/00BR0000000Wc0rMAC
```

You can also use the object API name and list view API name of the list view to make the same request.

```
/services/data/v47.0/ui-api/list-ui/Account/AllAccounts
```
The request returns a list of records and the metadata for the list view.

```json
{
    "eTag": "6fea2ca99f3740c72f1667ee522980a7",
    "info": {
        "cloneable": true,
        "createable": true,
        "deletable": true,
        "displayColumns": [
            {
                "fieldApiName": "Name",
                "label": "Account Name",
                "sortable": true
            },
            {
                "fieldApiName": "Site",
                "label": "Account Site",
                "sortable": true
            },
            {
                "fieldApiName": "BillingState",
                "label": "Billing State/Province",
                "sortable": true
            },
            {
                "fieldApiName": "Phone",
                "label": "Phone",
                "sortable": true
            },
            {
                "fieldApiName": "Type",
                "label": "Type",
                "sortable": true
            },
            {
                "fieldApiName": "Owner.Alias",
                "label": "Account Owner Alias",
                "sortable": true
            }
        ],
        "eTag": "f5f522aef88a0cb38be54b9c0095fe3c",
        "filterLogicString": null,
        "filteredByInfo": [],
        "label": "All Accounts",
        "listReference": {
            "id": "00BRM000002KPM42AO",
            "listViewApiName": "AllAccounts",
            "objectApiName": "Account",
            "type": "listView"
        },
        "orderedByInfo": [
            {
                "fieldApiName": "Name",
                "isAscending": true,
                "label": "Account Name"
            }
        ],
        "updateable": true,
        "userPreferences": {
            "columnWidths": {
                "Site": -1,
                "Type": -1,
                "Owner.Alias": -1,
                "Phone": -1,
```
"BillingState" : -1,
"Name" : -1
},
"columnWrap" : {
"Site" : false,
"Type" : false,
"Owner.Alias" : false,
"Phone" : false,
"BillingState" : false,
"Name" : false
},
"visibility" : "Public",
"visibilityEditable" : true
},
"records" : {
"count" : 12,
"currentPageToken" : "0",
"currentPageUrl" : "/services/data/v43.0/ui-api/list-records/Account/AllAccounts?pageSize=50&pageToken=0",
"listInfoETag" : "f5f522afe88a0cb38be54b9c0095fe3c",
"nextPageToken" : null,
"nextPageUrl" : null,
"previousPageToken" : null,
"previousPageUrl" : null,
"records" : [ {
"apiName" : "Account",
"childRelationships" : { },
"eTag" : "d55bcd780f2145b724be5a563f37b9b2",
"fields" : {
"BillingState" : {
"displayValue" : null,
"value" : "NC"
},
"CreatedDate" : {
"displayValue" : null,
"value" : "2018-04-11T00:37:56.000Z"
},
"Id" : {
"displayValue" : null,
"value" : "001RM000003UNtxYAG"
},
"LastModifiedDate" : {
"displayValue" : null,
"value" : "2018-04-11T00:37:56.000Z"
},
"Name" : {
"displayValue" : null,
"value" : "Burlington Textiles Corp of America"
},
"Owner" : {
"displayValue" : null,
"value" : { }}
"apiName" : "User",
"childRelationships" : { },
"eTag" : "671bc4877d13cd4f43d28a671636f873",
"fields" : {
   "Alias" : {
      "displayValue" : null,
      "value" : "AUser"
   },
   "Id" : {
      "displayValue" : null,
      "value" : "005RM000001cNuJYAU"
   },
   "id" : "005RM000001cNuJYAU",
   "recordTypeInfo" : null
},
"OwnerId" : {
   "displayValue" : null,
   "value" : "005RM000001cNuJYAU"
},
"Phone" : {
   "displayValue" : null,
   "value" : "(336) 222-7000"
},
"Site" : {
   "displayValue" : null,
   "value" : null
},
"SystemModstamp" : {
   "displayValue" : null,
   "value" : "2018-04-11T00:37:56.000Z"
},
"Type" : {
   "displayValue" : "Customer - Direct",
   "value" : "Customer - Direct"
},
"id" : "001RM000003UNtxYAG",
"recordTypeInfo" : null
},

... Additional records removed for brevity ...

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>fields</td>
<td>String[]</td>
<td>Additional fields queried for the records returned. These fields don't create visible</td>
<td>Optional</td>
<td>42.0</td>
</tr>
</tbody>
</table>
### Get Most Recently Used List View Records

Returns record data for an object’s most recently used (MRU) list view.

**Resource**

```
/ui-api/mru-list-records/${objectApiName}
```

- `objectApiName`—A supported object, such as Account.

**Available version**

43.0

**HTTP methods**

GET

**Example**

To get information about the records in the most recently used account list views, make this request:

```
GET /services/data/v43.0/ui-api/mru-list-records/Account
```
The request returns a list of records in the most recently used account list view.

```json
{
   "count" : 6,
   "currentPageToken" : "0",
   "currentPageUrl" : "/services/data/v43.0/ui-api/mru-list-records/Account?pageSize=50&pageToken=0",
   "listInfoETag" : "c4f11fa9015eeb2387d301daeebb8b43",
   "nextPageToken" : null,
   "nextPageUrl" : null,
   "previousPageToken" : null,
   "previousPageUrl" : null,
   "records" : [ {
      "apiName" : "Account",
      "childRelationships" : { },
      "eTag" : "e4475909d999e4c05a79afcf25b74947",
      "fields" : {
         "createdDate" : { 
            "displayValue" : null,
            "value" : "2018-04-11T00:37:56.000Z"
         },
         "id" : { 
            "displayValue" : null,
            "value" : "001RM000003UNu0YAG"
         },
         "lastModifiedDate" : { 
            "displayValue" : null,
            "value" : "2018-04-11T00:37:56.000Z"
         },
         "name" : { 
            "displayValue" : null,
            "value" : "Grand Hotels & Resorts Ltd"
         },
         "owner" : { 
            "displayValue" : null,
            "value" : { 
               "apiName" : "User",
               "childRelationships" : { },
               "eTag" : "671bc4877d13cd4f3d28a671636f873",
               "fields" : {
                  "alias" : { 
                     "displayValue" : null,
                     "value" : "AUser"
                  },
                  "id" : { 
                     "displayValue" : null,
                     "value" : "005RM000001cNuJYAU"
                  }
               }
            }
         },
         "ownerId" : { 
            "displayValue" : null,
            "value" : "005RM000001cNuJYAU"
         }
      }
   }
}
```
### Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>fields</td>
<td>String[]</td>
<td>Additional fields queried for the records returned. These fields don’t create visible columns. If the field is not available to the user, an error occurs.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>optionalFields</td>
<td>String[]</td>
<td>Additional fields queried for the records returned. These fields don’t create visible columns. If the field is not available to the user, no error occurs and the field isn’t included in the records.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>pageSize</td>
<td>String</td>
<td>The number of list records viewed at one time. The default value is 50. Value can be 1–2000.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>pageToken</td>
<td>String</td>
<td>A token that represents the page offset. To indicate where the page starts, use this value with the <code>pageSize</code> parameter. The maximum offset is 2000 and the default is 0.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>sortBy</td>
<td>String[]</td>
<td>The API name of the field the list view is sorted by. If the name is preceded with “-“, the sort order is descending. For example, &quot;Name&quot; sorts by name in ascending order.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
</tbody>
</table>
Response body for GET

Most Recently Used List Record Collection

Get Most Recently Used List View Metadata

Returns metadata for an object’s most recently viewed (MRU) list view.

Resource

/ui-api/mru-list-info/${objectApiName}

- objectApiName—A supported object, such as Account.

Available version

43.0

HTTP methods

GET

Example

To get information about the metadata for the most recently used account list views, make this request:

GET /services/data/v43.0/ui-api/mru-list-info/Account

The request returns the metadata for the most recently used account list view.

```json
{
  "cloneable" : false,
  "createable" : true,
  "deletable" : false,
  "displayColumns" : [ {
    "fieldApiName" : "Name",
    "label" : "Account Name",
    "sortable" : true
  }, {
    "fieldApiName" : "Site",
    "label" : "Account Site",
    "sortable" : true
  }, {
    "fieldApiName" : "Phone",
    "label" : "Phone",
    "sortable" : true
  }, {
    "fieldApiName" : "Owner.Alias",
    "label" : "Account Owner Alias",
    "sortable" : true
  } ],
  "eTag" : "c4f11fa9015eeb2387d301daeebb8b43",
  "filterLogicString" : null,
  "name" : "Most Recently Used View",
  "objectApiName" : "Account",
  "showMoreOptions" : true,
  "showMoreOptionsLabel" : "Show more options",
  "showMoreOptionsValue" : null,
  "showQuickFilterInputLabel" : false,
  "showQuickFilterInputValue" : null,
  "showQuickFilterInputWarning" : false,
  "showQuickFilterOptionLabel" : false,
  "showQuickFilterOptionValue" : null,
  "showQuickFilterOptionWarning" : false,
  "showQuickFilterOptionText" : null,
  "showQuickFilterOptionTextWarning" : false,
  "showQuickFilterOptionWarningMessage" : null,
  "sortable" : true,
  "sortDirection" : "asc",
  "sortField" : "-CreatedDate",
  "useQuickFilter" : false
}
```
Get Most Recently Used List View Records and Metadata

Returns record data and metadata for an object’s most recently used (MRU) list view.

Resource

```
/ui-api/mru-list-ui/${objectApiName}
```

- `objectApiName` — A supported object, such as Account.

Available version

43.0

HTTP methods

GET

Example

To get information about the record data and metadata for the most recently used account list views, make this request.

```
GET /services/data/v43.0/ui-api/mru-list-ui/Account
```

The request returns the record data and metadata for the most recently used account list view.

```
{
  "eTag" : "70eb4acb889c6dc5954f2e4ad25f428c",
```

Response body for GET

Most Recently Used List Metadata
"info" : {
"cloneable" : false,
"createable" : true,
"deletable" : false,
"displayColumns" : [ {
"fieldApiName" : "Name",
"label" : "Account Name",
"sortable" : true
}, {
"fieldApiName" : "Site",
"label" : "Account Site",
"sortable" : true
}, {
"fieldApiName" : "Phone",
"label" : "Phone",
"sortable" : true
}, {
"fieldApiName" : "Owner.Alias",
"label" : "Account Owner Alias",
"sortable" : true
}
],
"eTag" : "c4f11fa9015eeb2387d301daebb8b43",
"filterLogicString" : null,
"filteredByInfo" : [ ],
"label" : "Recently Viewed",
"listReference" : {
"id" : null,
"listViewApiName" : null,
"objectApiName" : "Account",
"type" : "mru"
},
"orderedByInfo" : [ ],
"updateable" : false,
"userPreferences" : {
"columnWidths" : {
"Site" : -1,
"Owner.Alias" : -1,
"Phone" : -1,
"Name" : -1
},
"columnWrap" : {
"Site" : false,
"Owner.Alias" : false,
"Phone" : false,
"Name" : false
}
},
"visibility" : "Public",
"visibilityEditable" : false
},
"records" : {
"count" : 6,
"currentPageToken" : "0",
"currentPageUrl" :
Get Most Recently Used List View Records and Metadata

"/services/data/v43.0/ui-api/mru-list-records/Account?pageSize=50&pageToken=0",
"listInfoETag" : "c4f11fa9015eeb2387d301daeebb8b43",
"nextPageToken" : null,
"nextPageUrl" : null,
"previousPageToken" : null,
"previousPageUrl" : null,
"records" : [ {
  "apiName" : "Account",
  "childRelationships" : { },
  "eTag" : "e4475909d999e4c05a79afcf25b74947",
  "fields" : { 
    "apiName" : "Account",
    "childRelationships" : { },
    "Id" : { 
      "displayValue" : null,
      "value" : "001RM000003UNu0YAG"
    },
    "LastModifiedDate" : { 
      "displayValue" : null,
      "value" : "2018-04-11T00:37:56.000Z"
    },
    "Name" : { 
      "displayValue" : null,
      "value" : "Grand Hotels & Resorts Ltd"
    },
    "OwnerId" : { 
      "displayValue" : null,
      "value" : "005RM000001cNuJYAU"
    },
    "Owner" : { 
      "displayValue" : null,
      "value" : { 
        "apiName" : "User",
        "childRelationships" : { },
        "eTag" : "671bc4877d13cd4f43d28a671636f873",
        "fields" : { 
          "Alias" : { 
            "displayValue" : null,
            "value" : "AUser"
          },
          "Id" : { 
            "displayValue" : null,
            "value" : "005RM000001cNuJYAU"
          }
        },
        "id" : "005RM000001cNuJYAU",
        "recordTypeInfo" : null
      }
    },
    "Phone" : { 
      "displayValue" : null,
      "value" : "(312) 596-1000"
    }
  }
},
"OwnerId" : { 
  "displayValue" : null,
  "value" : "005RM000001cNuJYAU"
},
"Phone" : { 
  "displayValue" : null,
  "value" : "(312) 596-1000"
User Interface API Resources

Get Most Recently Used List View Records and Metadata

```
{,
  "Site" : {
    "displayValue" : null,
    "value" : null
  },
  "SystemModstamp" : {
    "displayValue" : null,
    "value" : "2018-04-11T00:37:56.000Z"
  },
  "id" : "001RM000003UNu0YAG",
  "recordTypeInfo" : null
},
...

Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>fields</td>
<td>String[]</td>
<td>Additional fields queried for the records returned. These fields don’t create visible columns. If the field is not available to the user, an error occurs.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>optionalFields</td>
<td>String[]</td>
<td>Additional fields queried for the records returned. These fields don’t create visible columns. If the field is not available to the user, no error occurs and the field isn’t included in the records.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>pageSize</td>
<td>String</td>
<td>The number of list records viewed at one time. The default value is 50. Value can be 1–2000.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>pageToken</td>
<td>String</td>
<td>A token that represents the page offset. To indicate where the page starts, use this value with the pageSize parameter. The maximum offset is 2000 and the default is 0.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>sortBy</td>
<td>String[]</td>
<td>The API name of the field the list view is sorted by. If the name is preceded with “-”, the sort order is descending. For example, &quot;Name&quot; sorts by name in ascending order. &quot;-CreatedDate&quot; sorts by created date in descending order.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
</tbody>
</table>
Response body for GET
Most Recently Used List UI

Apps

Use these resources to get data and metadata about apps displayed in the Salesforce UI.

Get Apps
Get metadata for all the apps a user has access to. Metadata for the selected app includes tabs on the app’s navigation bar. Metadata for other apps doesn’t include tabs on the navigation bar.

Get an App
Get metadata about an app.

Update Last Selected App
Returns metadata for an app, and saves an app as the last selected for a user.

Get Last Selected App
Retrieves the app the current user last selected or the app the user sees by default.

Get Personalized Navigation Items
Get a user’s personalized navigation items (tabs).

Get All Navigation Items
Gets all navigation items (tabs) that the user has access to.

Update Personalized Navigation Items
Updates the order of personalized navigation items (tabs) and adds a navigation item to the list in the order specified.

Get Apps
Get metadata for all the apps a user has access to. Metadata for the selected app includes tabs on the app’s navigation bar. Metadata for other apps doesn’t include tabs on the navigation bar.

Resource
/ui-api/apps

Available Version
43.0

HTTP Method
GET

Example
As an example, let’s say that the user has access to three apps: Service, Marketing, and Bolt Solutions. To get metadata for all the apps, make this request:

GET /services/data/v47.0/ui-api/apps?formFactor=Large

The request returns metadata for the apps and their associated navigation tabs.

```json
{
    "apps" : [ {
```
"appId" : "06mRM000000AFIrYAO",
"description" : "Manage customer service with accounts, contacts, cases, and more",

"developerName" : "Service",
"formFactors" : [ "Large" ],
"headerColor" : "#0070D2",
"iconUrl" :
"https://yourInstance.salesforce.com/logos/Salesforce/ServiceCloud/icon.png",
"isNavAutoTempTabsDisabled" : false,
"isNavPersonalizationDisabled" : false,
"label" : "Service",
"logoUrl" :
"https://yourInstance.salesforce.com/logos/Salesforce/ServiceCloud/logo.png",
"mobileStartUrl" : null,
"navItems" : [ {
  "availableInClassic" : true,
  "availableInLightning" : true,
  "color" : "EF7EAD",
  "content" : "home__desktopDefault",
  "custom" : false,
  "developerName" : "standard-home",
  "iconUrl" :
  "https://yourInstance.salesforce.com/img/icon/t4v35/standard/home_120.png",
  "id" : null,
  "itemType" : "Standard",
  "label" : "Home",
  "objectApiName" : "Home",
  "pageReference" : {
    "attributes" : {
      "pageTitle" : "home",
    },
    "state" : {},
    "type" : "standard__namedPage"
  },
  "standardType" : "Home"
}, {
  "availableInClassic" : false,
  "availableInLightning" : true,
  "color" : "489DD0",
  "content" : null,
  "custom" : false,
  "developerName" : "standard-Feed",
  "iconUrl" :
  "https://yourInstance.salesforce.com/img/icon/t4v35/standard/feed_120.png",
  "id" : null,
  "itemType" : "Standard",
  "label" : "Chatter",
  "objectApiName" : "Feed",
  "pageReference" : {
    "attributes" : {
      "pageTitle" : "chatter",
    },
    "state" : {},
    "type" : "standard__namedPage"
  }
} ]
Get Apps

User Interface API Resources

... Additional actions removed for brevity ...


}
Request Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The form factor for each app that the user has access to.</td>
<td>Required</td>
<td>43.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Large—App supports the desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Medium—App supports the tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Small—App supports the phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>userCustomizations</td>
<td>Boolean</td>
<td>If true, gets custom and standard navigation tabs. If false, gets only standard navigation tabs.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
</tbody>
</table>

Response body for GET

Apps

SEE ALSO:

App
Apps
Manage Apps

Get an App

Get metadata about an app.

Resource

/ui-api/apps/${appId}

Available version

43.0

HTTP methods

GET

Example

As an example, let’s say that the user has an app with an ID of 06mRM000000AFIrYAO. To get metadata for this app, make this request:

GET /services/data/v43.0/ui-api/apps/06mRM000000AFIrYAO?formFactor=Large

The request returns metadata for the app and its associated navigation tabs.

```json
{
  "appId": "06mRM000000AFIrYAO",
```
"description": "Manage customer service with accounts, contacts, cases, and more",
"developerName": "Service",
"formFactors": ["Large"],
"headerColor": "#0070D2",
"iconUrl": "https://yourInstance.salesforce.com/logos/Salesforce/ServiceCloud/icon.png",
"isNavAutoTempTabsDisabled": false,
"isNavPersonalizationDisabled": false,
"label": "Service",
"logoUrl": "https://yourInstance.salesforce.com/logos/Salesforce/ServiceCloud/logo.png",
"mobileStartUrl": null,
"navItems": [{
  "availableInClassic": true,
  "availableInLightning": true,
  "color": "EF7EAD",
  "content": "home__desktopDefault",
  "custom": false,
  "developerName": "standard-home",
  "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/standard/home_120.png",
  "id": null,
  "itemType": "Standard",
  "label": "Home",
  "objectApiName": "Home",
  "pageReference": {
    "attributes": {
      "pageTitle": "home"
    },
    "state": {},
    "type": "standard__namedPage"
  },
  "standardType": "Home"
},
{ "availableInClassic": false,
  "availableInLightning": true,
  "color": "489DD0",
  "content": null,
  "custom": false,
  "developerName": "standard-Feed",
  "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/standard/feed_120.png",
  "id": null,
  "itemType": "Standard",
  "label": "Chatter",
  "objectApiName": "Feed",
  "pageReference": {
    "attributes": {
      "pageTitle": "chatter"
    },
    "state": {},
    "type": "standard__namedPage"
  },
  "standardType": "Feed"}
Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The form factor for each app that the user has access to.</td>
<td>Required</td>
<td>43.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—App supports the desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—App supports the tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—App supports the phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>userCustomizations</td>
<td>Boolean</td>
<td>If true, gets custom and standard navigation tabs. If false, gets only standard navigation tabs.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
</tbody>
</table>

Response body for GET

**App**

Update Last Selected App

Returns metadata for an app, and saves an app as the last selected for a user.

**Resource**

/ui-api/apps/${appId}

**Available version**

43.0

**HTTP methods**

PATCH

**Example**

As an example, let’s say that the user has an app with an ID of 06mRM000000AFIrYAO. To get metadata for this app, make this request:

```
PATCH /services/data/v43.0/ui-api/apps/06mRM000000AFIrYAO?formFactor=Large
```
The PATCH request doesn’t require a request body. The request returns metadata for the app and its associated navigation tabs. It also updates the last selected app to be this app.

```json
{
    "appId": "06mRM000000AFIrYAO",
    "description": "Manage customer service with accounts, contacts, cases, and more",
    "developerName": "Service",
    "formFactors": ["Large"],
    "headerColor": "#0070D2",
    "iconUrl": "https://yourInstance.salesforce.com/logos/Salesforce/ServiceCloud/icon.png",
    "isNavAutoTempTabsDisabled": false,
    "isNavPersonalizationDisabled": false,
    "label": "Service",
    "logoUrl": "https://yourInstance.salesforce.com/logos/Salesforce/ServiceCloud/logo.png",
    "mobileStartUrl": null,
    "navItems": [{
        "availableInClassic": true,
        "availableInLightning": true,
        "color": "EF7EAD",
        "content": "home__desktopDefault",
        "custom": false,
        "developerName": "standard-home",
        "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/standard/home_120.png",
        "id": null,
        "itemType": "Standard",
        "label": "Home",
        "objectApiName": "Home",
        "pageReference": {
            "attributes": {
                "pageName": "home"
            },
            "state": {},
            "type": "standard__namedPage"
        },
        "standardType": "Home"
    }, {
        "availableInClassic": false,
        "availableInLightning": true,
        "color": "489DD0",
        "content": null,
        "custom": false,
        "developerName": "standard-Feed",
        "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/standard/feed_120.png",
        "id": null,
        "itemType": "Standard",
        "label": "Chatter",
        "objectApiName": "Feed",
        "pageReference": {
            "attributes": {
                "pageName": "chatter"
            }
        }
    }]
}
```
Request parameters for PATCH

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The form factor for each app that the user has access to.</td>
<td>Required</td>
<td>43.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—App supports the desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—App supports the tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—App supports the phone display size.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Response body for PATCH

App

Get Last Selected App

Retrieves the app the current user last selected or the app the user sees by default.

Resource

 `/ui-api/apps/selected`

Available version

43.0

HTTP methods

GET

Example

To get metadata for the last selected app, make this request:

GET /services/data/v43.0/ui-api/apps/selected?formFactor=Large

The request returns metadata for the app and its associated navigation items.

```json
{
    "appId": "06mRM000000AFIrYAO",
    "description": "Manage customer service with accounts, contacts, cases, and more",
    "formFactor": "Large",
    "name": "cases",
    "navigationItems": [
        {
            "id": "06mRM000000AFIrYAO",
            "name": "Cases",
            "type": "standard__namedPage"
        }
    ]
}
```
"developerName": "Service",
"formFactors": ["Large"],
"headerColor": "#0070D2",
"iconUrl": "https://yourInstance.salesforce.com/logos/Salesforce/ServiceCloud/icon.png",

"isNavAutoTempTabsDisabled": false,
"isNavPersonalizationDisabled": false,
"label": "Service",
"logoUrl": "https://yourInstance.salesforce.com/logos/Salesforce/ServiceCloud/logo.png",

"mobileStartUrl": null,
"navItems": [{
    "availableInClassic": true,
    "availableInLightning": true,
    "color": "EF7EAD",
    "content": "home__desktopDefault",
    "custom": false,
    "developerName": "standard-home",
    "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/standard/home_120.png",
    "id": null,
    "itemType": "Standard",
    "label": "Home",
    "objectApiName": "Home",
    "pageReference": {
        "attributes": {
            "pageName": "home"
        },
        "state": {},
        "type": "standard__namedPage"
    },
    "standardType": "Home"
},
{"availableInClassic": false,
 "availableInLightning": true,
 "color": "489DD0",
 "content": null,
 "custom": false,
 "developerName": "standard-Feed",
 "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/standard/feed_120.png",
 "id": null,
 "itemType": "Standard",
 "label": "Chatter",
 "objectApiName": "Feed",
 "pageReference": {
        "attributes": {
            "pageName": "chatter"
        },
        "state": {},
        "type": "standard__namedPage"
    },
    "standardType": "Feed"}]}
Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The form factor for each app that the user has access to.</td>
<td>Required</td>
<td>43.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—App supports the desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—App supports the tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—App supports the phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>userCustomizations</td>
<td>Boolean</td>
<td>If true, gets custom and standard navigation tabs. If false, gets only standard navigation tabs.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
</tbody>
</table>

Response body for GET

Get Personalized Navigation Items

Get a user’s personalized navigation items (tabs).

Resource

`/ui-api/apps/${appId}/user-nav-items`

Available version

43.0

HTTP methods

GET

Example

To get a user’s personalized navigation tabs for an app, make this request using the app ID:

GET /services/data/v43.0/ui-api/apps/06mRM000000AFIrYAO/user-nav-items

The request returns metadata for the personalized navigation tabs associated with this app.

```json
{
    "navItems" : [ 
        {
            "availableInClassic" : true,
            "availableInLightning" : true,
        }
    ]
}
```
Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The form factor for each app that the user has access to.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—App supports the desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—App supports the tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—App supports the phone display size.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Response body for GET

Get All Navigation Items

Gets all navigation items (tabs) that the user has access to.

Resource

```
/ui-api/nav-items
```

Available version

45.0

HTTP methods

GET

Example

To get all navigation tabs for an app, make this request:

```
GET /services/data/v47.0/ui-api/nav-items?&page=3&pageSize=2
```
The request returns metadata for all navigation tabs associated with this app.

```
{
  "currentPageUrl": "services/data/v47.0/ui-api/nav-items?page=3&pageSize=2"
  "navItems": [{
    "availableInClassic": true,
    "availableInLightning": true,
    "color": "EF7EAD",
    "content": "home__desktopDefault",
    "custom": false,
    "developerName": "standard-home",
    "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/standard/home_120.png",
    "id": null,
    "itemType": "Standard",
    "label": "Home",
    "objectApiName": "Home",
    "pageReference": {
      "attributes": {
        "pageName": "home"
      },
      "state": {},
      "type": "standard__namedPage"
    },
    "standardType": "Home"
  }, {
    "availableInClassic": false,
    "availableInLightning": true,
    "color": "489DD0",
    "content": null,
    "custom": false,
    "developerName": "standard-Feed",
    "iconUrl": "https://yourInstance.salesforce.com/img/icon/t4v35/standard/feed_120.png",
    "id": null,
    "itemType": "Standard",
    "label": "Chatter",
    "objectApiName": "Feed",
    "pageReference": {
      "attributes": {
        "pageName": "chatter"
      },
      "state": {},
      "type": "standard__namedPage"
    },
    "standardType": "Feed"
  }],
  "nextPageUrl": "services/data/v47.0/ui-api/nav-items?page=4&pageSize=2"
}
```
### Request parameters for GET

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>formFactor</td>
<td>String</td>
<td>The form factor for each navigation tab that the user has access to.</td>
<td>Optional</td>
<td>45.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Large—Navigation tab supports the desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Medium—Navigation tab supports the tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Small—Navigation tab supports the phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pageSize</td>
<td>Int</td>
<td>The maximum number of child records to return on a page. The default value is 25.</td>
<td>Optional</td>
<td>45.0</td>
</tr>
<tr>
<td>page</td>
<td>Int</td>
<td>The page offset from which to begin returning records. The default value is 0, which returns records from the first page. For example, for page=2 and pageSize=10, the first record returned is the 21st record in the list.</td>
<td>Optional</td>
<td>45.0</td>
</tr>
</tbody>
</table>

### Response body for GET

**Navigation Items**

### Update Personalized Navigation Items

Updates the order of personalized navigation items (tabs) and adds a navigation item to the list in the order specified.

**Resource**

```
/ui-api/apps/{appId}/user-nav-items
```

**Available Version**

47.0

**HTTP Method**

PUT

**Example**

To update the order that navigation items display in an app and add an item to the list, make this request using the app’s ID.

```
PUT /services/data/v47.0/ui-api/apps/06mRM000000U6pOYAS/user-nav-items
```

The request body contains the list of navigation items in the order they should appear. To add a navigation item, include a `pageReference` describing the item.

```
{
    "navItems": [{
        "id": "0QkRM000000581NOAQ"
    }]
}
The request returns metadata for the navigation items associated with this app.

```json
{
  "currentPageUrl":
  "https://yourinstance.salesforce.com/services/data/v47.0/ui-api/apps/06mRM000000U6pOYAS/user-nav-items?formFactor=large",
  "navItems": [{
    "availableInClassic": true,
    "availableInLightning": true,
    "color": "2ECBBE",
    "content": "https://yourinstance.salesforce.com/00O/o",
    "custom": false,
    "developerName": "standard-report",
    "iconUrl":
    "https://yourinstance.salesforce.com/img/icon/t4v35/standard/report_120.png",
    "id": "0QkRM00000058lN0AQ",
    "itemType": "Entity",
    "label": "Cars",
    "objectApiName": "Car__c",
    "objectLabel": "Car",
    "objectLabelPlural": "Cars",
    "pageReference": {
      "attributes": {
        "objectApiName": "Car__c",
        "actionName": "home"
      },
      "state": {},
      "type": "standard__objectPage"
    },
    "standardType": null
  },
  {
    "availableInClassic": false,
    "availableInLightning": true,
    "color": "7F8DE1",
    "content": "001RM000004QlpqYAC",
    "custom": true,
    "developerName": null,
    "iconUrl":
    "https://yourinstance.salesforce.com/img/icon/t4v35/standard/account_120.png",
    "id": "0QkRM00000058lR0AQ",
    "itemType": "Record"
  }]
}
```
**Update Personalized Navigation Items**

```json
"label": "My Top Account",
"objectApiName": "Account",
"objectLabel": "Account",
"objectLabelPlural": "Accounts",
"pageReference": {
  "attributes": {
    "recordId": "001RM000004QlpqYAC",
    "objectApiName": "Account",
    "actionName": "view"
  },
  "state": {},
  "type": "standard__recordPage"
},
"standardType": null,

  "availableInClassic": true,
  "availableInLightning": true,
  "color": "EF6E64",
  "content": "https://yourinstance.salesforce.com/01Z/o",
  "custom": false,
  "developerName": "standard-Dashboard",
  "iconUrl": "https://yourinstance.salesforce.com/img/icon/t4v35/standard/dashboard_120.png",
  "id": "0QkRM00000058ll0AA",
  "itemType": "Standard",
  "label": "Dashboards",
  "objectApiName": "Dashboard",
  "objectLabel": "Dashboard",
  "objectLabelPlural": "Dashboards",
  "pageReference": {
    "attributes": {
      "objectApiName": "Dashboard",
      "actionName": "home"
    },
    "state": {},
    "type": "standard__objectPage"
  },
  "standardType": "Dashboards"
},
"nextPageUrl": null
```

**Request Parameters**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>navItems</td>
<td>[] on page 143</td>
<td>A list of navigation items (tabs) in the updated order they should appear. The updated order persists across desktop and mobile environments.</td>
<td>Required</td>
<td>47.0</td>
</tr>
</tbody>
</table>
Get Active Theme

Get a Salesforce org's active theme. A theme uses colors, images, and banners to change the overall appearance of Salesforce. Administrators can define themes and switch themes to provide a different look. The User Interface API response matches the Admin's selection.

Resources

`/ui-api/themes/active`

Available Version

42.0

HTTP Methods

GET

Response Body

Theme
CHAPTER 4 User Interface API Request Bodies

The successful execution of a request to a resource can return a response body in JSON format.
A request to a resource always returns an HTTP response code, whether the request was successful or not.

Favorite Batch Collection Input
A collection of all the favorites in a batch. Pass this request body to update a batch of favorites.

Favorite Batch Input
Represents a single favorite when updating favorites in batch.

Favorite Input
A favorite to create or update.

Navigation Item Input
Represents the list of navigation items in the order that you want them to appear in the app.

Page Reference Input
To add a navigation item, include a pageReference. A pageReference is a JSON object that describes the navigation item’s page type, its attributes, and the state of the page.

Record Input
A description of a record to use in a request to create or update a record.

Favorite Batch Collection Input
A collection of all the favorites in a batch. Pass this request body to update a batch of favorites.

**JSON example**

```json
{
  favorites: [
    {
      id: "0MVR00000004DhnOAE",
      name: "Q4 Perf"
    },
    {
      id: "0MVR00000004DhsOAE"
    },
    {
      id: "0MVR00000004DiGOAU"
    },
    {
      id: "0MVR000000001e2OAA",
      name: "Office Group"
    },
    {
      id: "0MVR00000004GGlOAM"
    }
  ]
}
```
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>favorites</td>
<td>Favorite Batch Input[]</td>
<td>The list of favorites to keep and update.</td>
<td>Required</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- [Update a Batch of Favorites](#)

---

### Favorite Batch Input

Represents a single favorite when updating favorites in batch.

**JSON example**

```json
{
  favorites: [
    {
      id: "0MVR00000004DhnOAE",
      name: "Q4 Perf"
    },
    {
      id: "0MVR00000004DhsOAE"
    },
    {
      id: "0MVR00000004DiGOAU"
    },
    {
      id: "0MVR00000004GGlOAM",
      name: "Office Group"
    },
    {
      id: "0MVR00000004GGlOAM"
    }
  ]
}
```

**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>The ID of the favorite.</td>
<td>Required</td>
<td>41.0</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the favorite.</td>
<td>Required</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- [Update a Batch of Favorites](#)

---

### Favorite Input

A favorite to create or update.
### JSON example

```json
{
    name: "Most Important Accounts",
    sortOrder: 1,
    target: "00BR0000000tTWMA4",
    targetType: "ListView"
}
```

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the favorite.</td>
<td>You must specify either name or sortOrder. You may specify both.</td>
<td>41.0</td>
</tr>
<tr>
<td>sortOrder</td>
<td>Integer</td>
<td>The sort order of the favorite, from 1 to N.</td>
<td>You must specify either name or sortOrder. You may specify both.</td>
<td>41.0</td>
</tr>
<tr>
<td>target</td>
<td>String</td>
<td>The record, API name, or content being favorited.</td>
<td>Required in a POST request. Do not specify in a PATCH request.</td>
<td>41.0</td>
</tr>
<tr>
<td>targetType</td>
<td>String</td>
<td>The type of favorite. One of these values:</td>
<td>Required in a POST request. Do not specify in a PATCH request.</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ListView—A favorited list view.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ObjectHome—A favorited object home.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Record—A favorited record.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tab—A favorited tab.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SEE ALSO:
- Create a Favorite
- Get Favorites
- Update a Favorite
- Update Usage of a Favorite

### Navigation Item Input

Represents the list of navigation items in the order that you want them to appear in the app.
Example

PUT /services/data/v47.0/ui-api/apps/06mRM000000U6pOYAS/user-nav-items

```json
{
  "navItems": [
    {
      "id": "0QkRM0000000581N0AQ"
    },
    {
      "id": "0QkRM0000000581R0AQ",
      "label": "My Top Account"
    },
    {
      "pageReference": {
        "type": "standard__objectPage",
        "attributes": {
          "objectApiName": "Dashboard",
          "actionName": "home"
        },
        "state": {}
      }
    }
  ]
}
```

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>The ID of the navigation item to include.</td>
<td>Required</td>
<td>47.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The updated label for the navigation item in list views and records only.</td>
<td>Optional</td>
<td>47.0</td>
</tr>
<tr>
<td>pageReference</td>
<td>Page Reference Input</td>
<td>The page reference of the navigation item to add to the list. You can add one page reference per request.</td>
<td>Optional</td>
<td>47.0</td>
</tr>
</tbody>
</table>

Page Reference Input

To add a navigation item, include a pageReference. A pageReference is a JSON object that describes the navigation item’s page type, its attributes, and the state of the page.

Example

PUT /services/data/v47.0/ui-api/apps/06mRM000000U6pOYAS/user-nav-items

The request body contains the list of navigation items in the order they should appear. To add a navigation item, include a pageReference describing the item.

```json
{
  "navItems": [
    {
      "id": "0QkRM0000000581N0AQ"
    },
    {
      "id": "0QkRM0000000581R0AQ",
      "label": "My Top Account"
    }
  ]
}
```
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>String</td>
<td>The page reference type generates a unique URL format and defines attributes that apply to all pages of that type. See PageReference Types.</td>
<td>Required</td>
<td>47.0</td>
</tr>
<tr>
<td>attributes</td>
<td>Map&lt;String, Object&gt;</td>
<td>Map of values for each attribute specified by the page definition, for example, objectAPIName or actionName.</td>
<td>Required</td>
<td>47.0</td>
</tr>
<tr>
<td>state</td>
<td>Map&lt;String, Object&gt;</td>
<td>Map of conditional values that customize content on the page, such as filterName.</td>
<td>Optional</td>
<td>47.0</td>
</tr>
</tbody>
</table>

**Record Input**

A description of a record to use in a request to create or update a record.

**Example**

```json
POST /ui-api/records

{
  "apiName": "Account",
  "fields": {
    "Name": "Universal Containers"
  }
}
```

To provide an address, provide the individual fields as separate inputs. The compound field is only available in the response.

```json
{
  "apiName": "Account",
  "fields": {
    "Name": "Local Boxes",
    "BillingState": "WA",
    "BillingStreet": "123 Main Street",
    "BillingCity": "Seattle",
    "BillingPostalCode": "98101"
  }
}
```
```json
{
  "BillingCountry": "USA"
}
```

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required or Optional</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>allowSaveOnDuplicate</td>
<td>Boolean</td>
<td>Specifies whether to save a duplicate record (true), or not (false). The default value is false.</td>
<td>Optional</td>
<td>43.0</td>
</tr>
<tr>
<td>apiName</td>
<td>String</td>
<td>To create a record, specify the API name of an Object from which to create the record. To update a record, use <code>null</code> or don’t pass this property.</td>
<td>Required</td>
<td>41.0</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required or Optional</td>
<td>Available Version</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>fields</td>
<td>Map&lt;String, Object&gt;</td>
<td>Map of field names to field values. Format data types according to these rules:</td>
<td>Required</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Address—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Base64—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Boolean—JSON Boolean, true or false</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Currency—JSON Float</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Date—JSON string in correct format</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DateTime—JSON string in ISO 8601 format</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Double—JSON Float</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Email—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- EncryptedString—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Int—JSON Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Location—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- MultiPicklist—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percent—JSON Float</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Phone—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Picklist—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Reference—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- String—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TextArea—JSON String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Time—JSON String in format HH:MM:SS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Url—JSON String</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To specify the master record type, either don’t specify the recordTypeId field, or set it to null.

SEE ALSO:
- Create a Record
- Update a Record
A request to a resource returns a response body. Some response bodies also contain nested response bodies. A request to a resource always returns an HTTP response code, whether the request was successful or not.

User Interface API Top-Level Response Bodies
These response bodies are returned from requests to User Interface API resources. They may contain nested response bodies.

User Interface API Nested Response Bodies
These response bodies are nested in top-level response bodies.

User Interface API Top-Level Response Bodies
These response bodies are returned from requests to User Interface API resources. They may contain nested response bodies. A request to a resource always returns an HTTP response code, whether the request was successful or not.

Action
The actions for a single record or a collection of records.

App
The metadata for a single app.

Apps
A list of apps that the current user has access to.

Batch Results
The results from a request to a batch resource. Results include information for each subrequest in the batch.

Favorite
A single favorite.

Favorite Collection
A list of favorites.

List Metadata
Metadata that describes a list view.

List Record Collection
A collection of list view records.

List UI
The record data and metadata for a list view.

List View Summary Collection
A collection of list views for a given object.

Lookup Values
Records in a lookup relationship, organized by object type.
Most Recently Used List Metadata
Metadata that describes a most recently used (MRU) list view.

Most Recently Used List Record Collection
A collection of an object’s most recently used list view records.

Most Recently Used List UI
The record data and metadata for an object’s most recently used (MRU) list view.

Navigation Items
An ordered list of navigation items (tabs) for the current user.

Object Info
The metadata for an object.

Object Info Directory
A directory of objects that are supported by User Interface API and available to the context user.

Picklist Values
The picklist values for a field, scoped to a record type. If a picklist is dependent, this response includes the values of its immediate controlling field and how they map to the picklist.

Picklist Values Collection
A collection of picklist values for all the picklists of a specified record type.

Record
The field data, API name, child relationship data, and record type information for a record.

Record Collection
A paginated collection of Record response bodies.

Record Defaults
The default information and data needed to create or clone a record. Use these values in a request to POST /ui-api/records.

Record Layout
The layout information for a record.

Record UI
The layout information, field information, and data for a record.

Theme
A set of images and banners that make up a theme. Banners display across the top of a page and have a higher aspect ratio than images.

SEE ALSO:
User Interface API Nested Response Bodies

---

**Action**

The actions for a single record or a collection of records.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actions</td>
<td>Map&lt;String, Object Action&gt;</td>
<td>A map of record IDs to the actions for each record.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
## App

The metadata for a single app.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>String</td>
<td>The URL of the current request.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

### App

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>appId</td>
<td>String</td>
<td>The unique ID of the app.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>appType</td>
<td>String</td>
<td>The type of the app. One of these values:</td>
<td>Small, 47.0</td>
<td>47.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Classic</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Community</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Connected</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Lightning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A description of the app.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>developerName</td>
<td>String</td>
<td>The API name of the app.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>formFactors</td>
<td>String[]</td>
<td>The form factor of the app. One or more of these values:</td>
<td>Medium, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Large—App supports the desktop display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medium—App supports the tablet display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Small—App supports the phone display size.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>headerColor</td>
<td>String</td>
<td>The primary color for the app as selected by an admin. For example, #0070D2.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>iconUrl</td>
<td>String</td>
<td>The URL for the app's icon.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>isNavAutoTempTabsDisabled</td>
<td>Boolean</td>
<td>If true, the navigation automatically creates temporary tabs settings.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>isNavPersonalizationDisabled</td>
<td>Boolean</td>
<td>If true, navigation personalization is disabled.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The label of the app.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>Property Name</td>
<td>Type</td>
<td>Description</td>
<td>Filter Group and Version</td>
<td>Available Version</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>logoUrl</td>
<td>String</td>
<td>The logo URL of the app as selected by the admin.</td>
<td>Medium, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>mobileStartUrl</td>
<td>String</td>
<td>The mobile launch URL for the app. This field is used with connected apps and Lightning communities. For communities only, this location is a fully qualified domain name. For other apps, it's a relative URL.</td>
<td>Medium, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>navItems</td>
<td>Navigation Item[]</td>
<td>The metadata for the navigation tabs of the app.</td>
<td>Big, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>selected</td>
<td>Boolean</td>
<td>If true, this app is the default app for the user.</td>
<td>Big, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>startUrl</td>
<td>String</td>
<td>The launch URL of the app.</td>
<td>Medium, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>uiType</td>
<td>String</td>
<td>The UI type for the app. One of these values:</td>
<td>Small, 43.0 - 46.0</td>
<td>43.0 - 46.0</td>
</tr>
<tr>
<td>userNavItems</td>
<td>Navigation Item[]</td>
<td>The user’s navigation tabs for the app.</td>
<td>Big, 43.0</td>
<td>43.0</td>
</tr>
</tbody>
</table>

**Note:** Only navItems with supported formFactors values are returned.

SEE ALSO:
- Apps
- Get an App
- Get Last Selected App
- Manage Apps

**Apps**

A list of apps that the current user has access to.
### Batch Results

The results from a request to a batch resource. Results include information for each subrequest in the batch.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasErrors</td>
<td>Boolean</td>
<td><code>true</code> if at least one of the results in the result set is an HTTP status code in the 400 or 500 range; <code>false</code> otherwise</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>results</td>
<td>Batch Result Item[]</td>
<td>Collection of batch result items</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

### Favorite

A single favorite.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessCount</td>
<td>Integer</td>
<td>The number of times this favorite has been used.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>iconColor</td>
<td>String</td>
<td>The color of the icon for this favorite, usually the related object’s color.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>iconUrl</td>
<td>String</td>
<td>The URL of the icon for the favorite, usually the related object’s icon.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>The ID of the favorite.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>lastAccessDate</td>
<td>String</td>
<td>The last time this favorite was used. The value is in ISO 8061 date and time format.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the favorite.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
### Favorite Collection

A list of favorites.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>favorites</td>
<td>Favorite[]</td>
<td>A list of favorites.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- Create a Favorite
- Get Favorites
- Update a Favorite
- Update Usage of a Favorite

### List Metadata

Metadata that describes a list view.
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloneable</td>
<td>Boolean</td>
<td>Indicates whether the list view can be cloned.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>createable</td>
<td>Boolean</td>
<td>Indicates whether a new list view for this list view's object can be created.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>deletable</td>
<td>Boolean</td>
<td>Indicates whether the list view can be deleted.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>listViewApiName</td>
<td>String</td>
<td>The list view's API name. For example, &quot;AllAccounts&quot;.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>displayColumns</td>
<td>List Column[]</td>
<td>All display columns for this list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>filterLogicString</td>
<td>String</td>
<td>The filter logic string, such as &quot;(1 OR 2) and 3&quot;. Indexes start with 1.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>filteredByInfo</td>
<td>List Filter by Info[]</td>
<td>Filtering information for the list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>id</td>
<td>List View ID</td>
<td>Identity information for the list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The list view's display label. For example, &quot;All Accounts&quot;.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>orderedByInfo</td>
<td>List Order by Info[]</td>
<td>Ordering information for the list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>updateable</td>
<td>Boolean</td>
<td>Indicates whether the list view can be updated.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>userPreferences</td>
<td>List User Preference</td>
<td>User preferences for this list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>visibility</td>
<td>String</td>
<td>The list view's visibility. One of these values:</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Private</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Public</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Shared</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visibilityEditable</td>
<td>Boolean</td>
<td>Indicates whether the visibility of the list view can be edited.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

**List Record Collection**

A collection of list view records.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>listInfoETag</td>
<td>String</td>
<td>An ETag that indicates whether the metadata has changed.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>records</td>
<td>Record Collection</td>
<td>Collection of records for the list.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>
### List UI

The record data and metadata for a list view.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>metadata</td>
<td>List Metadata</td>
<td>The metadata for the list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>records</td>
<td>List Record Collection</td>
<td>The record data for the list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

### List View Summary Collection

A collection of list views for a given object.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>Integer</td>
<td>The total number of list views returned.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>currentPageToken</td>
<td>String</td>
<td>The pageToken of the current page of list views.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>currentPageUrl</td>
<td>String</td>
<td>The URL of the current page of list views.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>lists</td>
<td>List View Summary</td>
<td>Collection of list views for the given object</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>nextPageToken</td>
<td>String</td>
<td>The pageToken of the next page of list views.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>nextPageUrl</td>
<td>String</td>
<td>The URL of the next page of list views.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>previousPageToken</td>
<td>String</td>
<td>The pageToken of the previous page of list views.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>previousPageUrl</td>
<td>String</td>
<td>The URL of the previous page of list views.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- [Get List Views for an Object](#)

### Lookup Values

Records in a lookup relationship, organized by object type.
### User Interface API Response Bodies

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>lookupResults</td>
<td>Map &lt;String, Record Collection&gt;</td>
<td>A map of object API names to Record Collection response bodies containing the lookup results for that object type.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

**SEE ALSO:**
Get Lookup Field Suggestions

### Most Recently Used List Metadata

Metadata that describes a most recently used (MRU) list view.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloneable</td>
<td>Boolean</td>
<td>Indicates whether the list view can be cloned.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>createable</td>
<td>Boolean</td>
<td>Indicates whether a new list view for this list view's object can be created.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>deletable</td>
<td>Boolean</td>
<td>Indicates whether the list view can be deleted.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>displayColumns</td>
<td>List Column[]</td>
<td>All display columns for this list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>filterLogicString</td>
<td>String</td>
<td>The filter logic string, such as &quot;(1 OR 2) and 3&quot;. Indexes start with 1.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>filteredByInfo</td>
<td>List Filter by Info[]</td>
<td>Filtering information for the list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>id</td>
<td>List View ID</td>
<td>Identity information for the list view.</td>
<td>Small, 42.0—42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The list view's display label. E.g. &quot;All Accounts&quot;.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>listReference</td>
<td>List Reference</td>
<td>An object representing the identity of the list.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>listViewApiName</td>
<td>String</td>
<td>The list view's api name. E.g. &quot;AllAccounts&quot;.</td>
<td>Small, 42.0—42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>orderedByInfo</td>
<td>List Order by Info[]</td>
<td>Ordering information for the list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>updateable</td>
<td>Boolean</td>
<td>Indicates whether the list view can be updated.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>userPreferences</td>
<td>List User Preference</td>
<td>User preferences for this list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>visibility</td>
<td>String</td>
<td>The list view's visibility. One of these values:</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Private</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Public</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Available Version

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>visibilityEditable</td>
<td>Boolean</td>
<td>Indicates whether the visibility of the list view can be edited.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

SEE ALSO:  
[Get Most Recently Used List View Metadata](#)

### Most Recently Used List Record Collection

A collection of an object’s most recently used list view records.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>Integer</td>
<td>The total count of records returned.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>currentPageToken</td>
<td>String</td>
<td>The token for the current page of records.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>currentPageUrl</td>
<td>String</td>
<td>The URL of the current page of records.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>listInfoETag</td>
<td>String</td>
<td>An ETag that indicates whether the metadata has changed.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>nextPageToken</td>
<td>String</td>
<td>The token for the next page of records.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>nextPageUrl</td>
<td>String</td>
<td>The URL of the next page of records.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>previousPageToken</td>
<td>String</td>
<td>The token for the previous page of records.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>previousPageUrl</td>
<td>String</td>
<td>The URL of the previous page of records.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>records</td>
<td>Record Collection</td>
<td>Collection of records for the MRU list view.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
</tbody>
</table>

SEE ALSO:  
[Get Most Recently Used List View Records](#)

### Most Recently Used List UI

The record data and metadata for an object’s most recently used (MRU) list view.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>info</td>
<td>List Metadata</td>
<td>The metadata for the list view.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
</tbody>
</table>
### Available Version

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>records</td>
<td>List Record Collection</td>
<td>The record data for the list view.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
</tbody>
</table>

SEE ALSO:

- Get Most Recently Used List View Records and Metadata

### Navigation Items

An ordered list of navigation items (tabs) for the current user.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>navItems</td>
<td>Navigation Item[]</td>
<td>Metadata for navigation tabs.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>currentPageUrl</td>
<td>String</td>
<td>URL to the current page of navigation tabs.</td>
<td>Medium, 45.0</td>
<td>45.0</td>
</tr>
<tr>
<td>nextPageUrl</td>
<td>String</td>
<td>URL to the next page of navigation tabs.</td>
<td>Medium, 45.0</td>
<td>45.0</td>
</tr>
</tbody>
</table>

SEE ALSO:

- Get Personalized Navigation Items
- Update Personalized Navigation Items

### Object Info

The metadata for an object.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>The object’s API name.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>childRelationships</td>
<td>Child Relationship[]</td>
<td>The object’s child relationships.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>createable</td>
<td>Boolean</td>
<td>Indicates whether the object can be created.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>custom</td>
<td>Boolean</td>
<td>Indicates whether the object is custom.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>defaultRecordTypeId</td>
<td>String</td>
<td>The ID for the default record type for this object, if any. If there isn’t a default record type, this value is the master record type, which is 012000000000000AAA. If the record is a nested record (for example, in a request for an Account, the User</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>Property Name</td>
<td>Type</td>
<td>Description</td>
<td>Filter Group and Available Version</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>object is a nested record), the value is null.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>deleteable</td>
<td>Boolean</td>
<td>Indicates whether the object can be deleted. Only available in API version 41.0. In API version 42.0, we spelled it correctly.</td>
<td>Small, 41.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>41.0-41.0</td>
<td></td>
</tr>
<tr>
<td>deletable</td>
<td>Boolean</td>
<td>Indicates whether the object can be deleted.</td>
<td>Small, 42.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>42.0</td>
<td></td>
</tr>
<tr>
<td>dependentFields</td>
<td>Map&lt;String, Object&gt;</td>
<td>A map of the dependent fields tree structure. Each nested object is another Map&lt;String, Object&gt;. When the object is empty, it indicates a leaf of the tree, which is a field that doesn’t control other fields. An object can have multiple independent trees, which means this property can have multiple root objects. Imagine an object with the picklists Continents__c, Countries__c, and Cities__c. Continents__c is the root. Cities__c is a leaf. Continents__c and Countries__c are controlling fields. Countries__c and Cities__c are dependent picklists.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Big, 42.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A field dependency causes the values in a picklist to be dynamically filtered based on the value selected by the user in another field. The field that drives filtering is called the &quot;controlling field.&quot; Standard and custom checkboxes and picklists with at least one and less than 300 values can be controlling fields. The field that has its values filtered is called the &quot;dependent field.&quot; Custom picklists and multi-select picklists can be dependent fields.</td>
<td>42.0</td>
<td></td>
</tr>
<tr>
<td>Property Name</td>
<td>Type</td>
<td>Description</td>
<td>Filter Group and Version</td>
<td>Available Version</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>feedEnabled</td>
<td>Boolean</td>
<td>Indicates whether the object can have feeds.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>fields</td>
<td>Map&lt;String, Field&gt;</td>
<td>A map of field API name to information about the field. This map only contains fields relevant to the requested layout and mode.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>keyPrefix</td>
<td>String</td>
<td>The key prefix for IDs of this object.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The object's display label.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>labelPlural</td>
<td>String</td>
<td>The plural form of the object's display label.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>layoutable</td>
<td>Boolean</td>
<td>Indicates whether the object can have a layout.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>mruEnabled</td>
<td>Boolean</td>
<td>Indicates whether the object can appear in Most Recently Used lists.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>nameFields</td>
<td>String[]</td>
<td>A collection of the API names of the fields used to identify the name field for standard and custom objects. Typically there is one name field per object, except where FirstName and LastName fields are used.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>queryable</td>
<td>Boolean</td>
<td>Indicates whether the context user can query the object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>recordTypeInfos</td>
<td>Map&lt;String, Record Type Info&gt;</td>
<td>A map of record type IDs to information about the record type. All record types are visible whether the user has access to them.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>searchable</td>
<td>Boolean</td>
<td>Indicates whether the object can be searched.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>themeInfo</td>
<td>Theme Info</td>
<td>Theme information for the object.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
Available Version
Filter Group and Version
Available Version

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>updateable</td>
<td>Boolean</td>
<td>Indicates whether the object can be updated.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- Get Record Data and Object Metadata
- Get Object Metadata
- Get Default Values to Clone a Record
- Get Default Values to Create a Record
- Record UI
- Record Defaults

Object Info Directory

A directory of objects that are supported by User Interface API and available to the context user.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>objects</td>
<td>Map&lt;String, Object Info Directory Entry&gt;</td>
<td>A map of objects supported by User Interface API to their object info directory entries.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- Get a Directory of Supported Objects

Picklist Values

The picklist values for a field, scoped to a record type. If a picklist is dependent, this response includes the values of its immediate controlling field and how they map to the picklist.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
</table>
| controllerValues | Map<String, Integer> | If the picklist is dependent, this property is a map of its immediate controlling field’s picklist values to their indexes. 
- If the controlling field is a picklist, the String is the picklist value and the integer is the value’s index. 
- If the controlling field is a checkbox, the values in the map are "false": 0 and "true": 1. | Small, 41.0 | 41.0 |
### Picklist Values Collection

A collection of picklist values for all the picklists of a specified record type.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>picklistFieldValues</td>
<td>Map&lt;String, Picklist Values&gt;</td>
<td>A map of field names to Picklist Values response bodies. This map contains all the picklist values for all the picklists of a specified record type, including dependent picklists. If a field isn’t a picklist, it isn’t represented in the map.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- Get Values for All Picklist Fields of a Record Type
- Build UI for Picklists

### Record

The field data, API name, child relationship data, and record type information for a record.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>The record’s API name.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
## User Interface API Response Bodies

### Record Collection

A paginated collection of Record response bodies.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>Integer</td>
<td>The page size.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>currentPageToken</td>
<td>String</td>
<td>Token identifying the current page.</td>
<td>Small, 44.0</td>
<td>44.0</td>
</tr>
<tr>
<td>currentPageUrl</td>
<td>String</td>
<td>User Interface API URL identifying the current page.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
### Record Defaults

The default information and data needed to create or clone a record. Use these values in a request to `POST /ui-api/records`.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>layout</td>
<td>Record Layout</td>
<td>Record layout information.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>objectInfo</td>
<td>Object Info</td>
<td>Object metadata.</td>
<td>Big, 41.0</td>
<td>41.0-41.0</td>
</tr>
<tr>
<td>objectInfos</td>
<td>Object Info[]</td>
<td>A collection of object metadata.</td>
<td>Big, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>record</td>
<td>Record[]</td>
<td>Pre-populated record data.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- Get Default Values to Clone a Record
- Get Default Values to Create a Record
- Create a Record

### Record Layout

The layout information for a record.
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>The layout ID.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>layoutType</td>
<td>String</td>
<td>The layout type for the record. One of these values:</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Compact—Use this value to get a layout that contains a record’s key fields.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Full—(Default) Use this value to get a full layout.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mode</td>
<td>String</td>
<td>The access mode for the record. This value determines which fields to get from a layout.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Layouts have different fields for create, edit, and view modes. For example, formula fields are</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>rendered in view mode, but not in create mode because they’re calculated at run time, like</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>formulas in a spreadsheet. One of these values:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Create—Use this mode if you intend to build UI that lets a user create a record. This mode is</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>used by the /ui-api/record-defaults/create/{apiName} resource.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Edit—Use this mode if you intend to build UI that lets a user edit a record. This mode is used by</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>the /ui-api/record-defaults/clone/{recordId} resource.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• View—(Default) Use this mode if you intend to build UI that displays a record.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sections</td>
<td>Record Layout Section[]</td>
<td>A collection of layout sections.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- Record UI
- Record Defaults
- Get Record Layout Metadata
- Get Default Values to Clone a Record
- Get Default Values to Create a Record

**Record UI**

The layout information, field information, and data for a record.
### User Interface API Response Bodies

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>layoutUserStates</td>
<td>Map&lt;String, Record&lt;Layout Section User State&gt;&gt;</td>
<td>A map of layout IDs to user state information.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>layouts</td>
<td>Map&lt;String, Map&lt;String, Map&lt;String, Record&lt;Layout&gt;&gt;&gt;&gt;</td>
<td>A map of object API names to layout information for each object.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>objectInfos</td>
<td>Map&lt;String, ObjectInfo&gt;</td>
<td>A map of object API names to each object’s metadata.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>records</td>
<td>Map&lt;String, Record&gt;</td>
<td>A map of record IDs to each record’s data.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

**SEE ALSO:**
- Get Record Data and Object Metadata

### Theme

A set of images and banners that make up a theme. Banners display across the top of a page and have a higher aspect ratio than images.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>brandColor</td>
<td>String</td>
<td>The brand color of the theme.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>brandImage</td>
<td>Theme Image</td>
<td>The brand image of the theme.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>defaultGroupBanner</td>
<td>Theme Banner</td>
<td>The default banner for groups. The group banner displays on a group’s home page.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>defaultGroupImage</td>
<td>Theme Image</td>
<td>The default image for groups. The group image displays beside a group name.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>defaultPageBanner</td>
<td>Theme Banner</td>
<td>The default banner for pages other than group and user pages.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>defaultUserBanner</td>
<td>Theme Banner</td>
<td>The default user banner of the theme. The user banner displays on a user’s home page.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>defaultUserImage</td>
<td>Theme Image</td>
<td>The default user image of the theme. The user image displays beside a username.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>density</td>
<td>String</td>
<td>The display density in the user interface. One of these values:</td>
<td>Small, 44.0</td>
<td>44.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ViewOne—User interface elements have a lower density on the page, which means they have more</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
white space between them. In Lightning Experience, this setting is called **Comfy**.

- **ViewTwo**—User interface elements have a higher density on the page, which means they have the least amount of white space between them. In Lightning Experience, this setting is called **Compact**.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>headerColor</td>
<td>String</td>
<td>The header color of the theme.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>linkColor</td>
<td>String</td>
<td>The link color of the theme.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>pageColor</td>
<td>String</td>
<td>The page color of the theme.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

**SEE ALSO:**

Get Active Theme

### User Interface API Nested Response Bodies

These response bodies are nested in top-level response bodies. A request to a resource always returns an **HTTP status code**, whether the request was successful or not.

- **Batch Result Item**
  The results of one subrequest in a batch request.

- **Canvas Layout Component**
  A canvas component on a record page layout.

- **Child Relationship**
  The child relationship on a parent object. Relationships are connections between records. Use relationships to display data about related object records on a record’s detail page.

- **Custom Link Layout Component**
  A custom link component on a record page layout.

- **Duplicate Record Error**
  Information about duplicate record errors. Duplicate rules are used to control whether and when you can save duplicate records within Salesforce. Duplicate rules tell Salesforce what action to take when you attempt to create a duplicate record.

- **Error Message**
  An error code and error message.

- **Error with Output**
  Contains extra information about errors. In rare cases, an error message isn’t enough to describe the reason for a failure. For example, when a conflicting precondition exists, the error result can include the information about the cause of the conflict.

- **Field**
  A field’s metadata.
**Field Layout Component**
A field in a record layout.

**Field Value**
The raw and displayable field values for a field in a record.

**Filtered Lookup Info**
Metadata for a lookup filter.

**Lead Status Picklist Value Attributes**
Additional picklist value attributes for lead statuses.

**List Column**
A column in a list.

**List Filter by Info**
Information used to filter a list.

**List Order by Info**
Ordering information for a list.

**List Reference**
Information about the list view referenced in the metadata.

**List User Preference**
User preferences for the list view.

**List View ID**
Information that identifies a list view.

**List View Summary**
A summary of list views for an object.

**Location Field**
A complex location value for a record field.

**Match Result**
A duplicate rule match. Duplicate rules are used to control whether and when you can save duplicate records within Salesforce. Duplicate rules tell Salesforce what action to take when you attempt to create a duplicate record. Each duplicate rule requires at least one matching rule to identify which existing records are possible duplicates.

**Navigation Item**
The metadata for a single navigation item (tab).

**Object Action**
The actions for an object.

**Object Info Directory Entry**
An entry in the Object Info Directory, which is a directory of objects that User Interface API supports and that the context user can access.

**Opportunity Stage Picklist Value Attributes**
Additional picklist value attributes for opportunity stages.

**Page Reference**
A page reference.

**Picklist Value**
A single picklist value.
Platform Action
The metadata, layout information, and data for a platform action.

Record Exceptions
A collection of record exception errors.

Record Exception Error
Information about a record exception error.

Record Layout Component
A concrete record layout component.

Record Layout Item
An item in a record layout.

Record Layout Row
A row in a record layout.

Record Layout Section
A section in a record layout.

Record Layout Section User State
The user state information for a record layout section.

Record Layout User State
The user state information for a record layout.

Record Type Info
Information about a record type.

Reference To Info
Information about a reference field's referenced types and the name field names of those types.

Report Layout Component
A report chart component on a record layout page.

Theme Banner
A theme banner image. Theme banners have a higher aspect ratio than theme images. The banner size may change slightly, so code accordingly.

Theme Image
A theme image at three sizes: small, medium, and large. Specific sizes may change slightly, so code accordingly.

Theme Info
Color and icon information for a theme.

Visualforce Layout Component
A Visualforce component on a record layout page.

SEE ALSO:
User Interface API Top-Level Response Bodies

Batch Result Item
The results of one subrequest in a batch request.
**User Interface API Response Bodies**

**Canvas Layout Component**

A canvas component on a record page layout.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>The API name of the canvas app.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>componentType</td>
<td>String</td>
<td>The value is Canvas.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>displayLocation</td>
<td>String</td>
<td>The location in the application where the canvas app is being called from.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

**result**

One of these types:
- **Error Message**

*Important:*
If the result is an error, the type is a collection of error messages.

- **Record**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>statusCode</td>
<td>Integer</td>
<td>An HTTP status code indicating the status of this individual request in the batch.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

**SEE ALSO:**

- [Batch Results](#)

---

**Canvas Layout Component**

A canvas component on a record page layout.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>The API name of the canvas app.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>componentType</td>
<td>String</td>
<td>The value is Canvas.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>displayLocation</td>
<td>String</td>
<td>The location in the application where the canvas app is being called from.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

- **Chatter**—The canvas app was called from the Chatter tab.
- **ChatterFeed**—The canvas app was called from a Chatter canvas feed item.
- **MobileNav**—The canvas app was called from the navigation menu in the Salesforce app.
- **OpenCTI**—The canvas app was called from an Open CTI component.
Available Version

Filter Group and Version

Available Version

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>height</td>
<td>String</td>
<td>The height of the component.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>referenceId</td>
<td>String</td>
<td>The unique ID of the canvas app definition.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>showScroll</td>
<td>Boolean</td>
<td>Indicates whether to show the scroll bar on the canvas component (true) or not (false).</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
Child Relationship

The child relationship on a parent object. Relationships are connections between records. Use relationships to display data about related object records on a record’s detail page.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>fieldName</td>
<td>String</td>
<td>The field on the child object that contains the reference to the parent object. Instead of defining relationships through primary keys and foreign keys, the Salesforce database uses reference fields. A reference field stores the ID of the related parent record.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>junctionIdListNames</td>
<td>String[]</td>
<td>The names of the JunctionIdList fields associated with an object. Each ID is a polymorphic key, which is an ID that can refer to more than one type of object.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>junctionReferenceTo</td>
<td>String[]</td>
<td>A collection of object names that the polymorphic keys in the junctionIdListNames property can reference. You can query these object names.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>childObjectApiName</td>
<td>String</td>
<td>The API name of the child object.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
### Relationship Name

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>relationshipName</td>
<td>String</td>
<td>The name of the relationship.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

A name for the child relationship that is unique to the parent. The name is the plural form of the child object name. For example, Account has child relationships to Assets, Cases, and Contacts, among other objects. Account has a `relationshipName` for each object: Assets, Cases, and Contacts.

### Custom Link Layout Component

A custom link component on a record page layout.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>If a field powers this component, this property contains the API name of the field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>behavior</td>
<td>String</td>
<td>Indicates how a link behaves on open. One of these values:</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NewWindow—Open a new window.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NoSidebar—No sidebar</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• OnClickJavaScript—When the link is clicked, run JavaScript.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Replace—Replace the current page.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Sidebar—Sidebar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>componentType</td>
<td>String</td>
<td>The component type. The value is CustomLink.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>customLinkUrl</td>
<td>String</td>
<td>The custom link URL.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The custom link label.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

### See Also:
- [Object Info](#)
- [Record Layout Item](#)
**Duplicate Record Error**

Information about duplicate record errors. Duplicate rules are used to control whether and when you can save duplicate records within Salesforce. Duplicate rules tell Salesforce what action to take when you attempt to create a duplicate record.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>matchResults</td>
<td>Match Result []</td>
<td>A collection of duplicate rule match results. Each duplicate rule requires at least one matching rule to identify which existing records are possible duplicates.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- [Record Exception Error](#)

**Error Message**

An error code and error message.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorCode</td>
<td>String</td>
<td>An error code with information about the error, for example, INSUFFICIENT_PRIVILEGES</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>message</td>
<td>String</td>
<td>Description of error</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- [Batch Result Item](#)

**Error with Output**

Contains extra information about errors. In rare cases, an error message isn't enough to describe the reason for a failure. For example, when a conflicting precondition exists, the error result can include the information about the cause of the conflict.

User Interface API enforces Salesforce validation rules. If a validation rule fails, the error is reported here.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>enhancedErrorType</td>
<td>String</td>
<td>Indicates the type of the output property. The value is either null or RecordError.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>message</td>
<td>String</td>
<td>Description of the error.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>output</td>
<td>Any response body</td>
<td>The response body returned by the requested resource. For example, if a successful request returns a Object Info response body but an error</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Description</td>
<td>Filter Group and Version</td>
<td>Available Version</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>triggers the Error with Output response body, the value of the output property is an Object Info response body. When an error occurs related to record create or update, this value is Record Exceptions.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**

A field's metadata.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>The API name for this field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>calculated</td>
<td>Boolean</td>
<td>Indicates whether the field is a custom formula field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>compound</td>
<td>Boolean</td>
<td>Indicates whether the field is a top-level compound field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>compound Component Name</td>
<td>String</td>
<td>If this field is a component field of a compound field, the value is the normalized component name, otherwise the value is null. For example, if the field represents &quot;BillingStreet&quot; of an Address compound field, this property contains &quot;Street&quot;.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>compound Field Name</td>
<td>String</td>
<td>If this field is a component field of a compound field, this property contains the top-level compound field. Otherwise, the value of this property is null.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>controller Name</td>
<td>String</td>
<td>If this field is a dependent picklist, this property is the name of the field that controls the values of the picklist.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>controlling Fields</td>
<td>String[]</td>
<td>If this field is a dependent picklist, this property is a collection of fields that control the values in the picklist. When there's a hierarchy of controlling fields, the collection starts with the immediate parent and moves up the tree.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>Property Name</td>
<td>Type</td>
<td>Description</td>
<td>Filter Group and Version</td>
<td>Available Version</td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>-------------</td>
<td>--------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>createable</td>
<td>Boolean</td>
<td>Indicates whether the field can be created.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>custom</td>
<td>Boolean</td>
<td>Indicates whether the field is custom.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>dataType</td>
<td>String</td>
<td>Field type. One of these values:</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Base64</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Boolean</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ComboBox</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ComplexValue—Complex Value Type (CVT).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Currency</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DateTime</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Double</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- EncryptedString</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Int</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Location</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- MultiPicklist—To populate the multi-select picklist, Get Values for a Picklist Field or Get Values for All Picklist Fields of a Record Type.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percent</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Phone</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Picklist—To populate the picklist, Get Values for a Picklist Field or Get Values for All Picklist Fields of a Record Type.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Reference</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- String</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TextArea</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Url</td>
<td></td>
<td></td>
</tr>
<tr>
<td>extraTypeInfo</td>
<td>String</td>
<td>More data type information. One of these values:</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ExternalLookup—External lookup</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ImageUrl—Image url</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property Name</td>
<td>Type</td>
<td>Description</td>
<td>Filter Group and Version</td>
<td>Available Version</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>filterable</td>
<td>Boolean</td>
<td>Indicates whether the field is filterable. If true, then this field can be specified in the WHERE clause of a SOQL statement.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>filteredLookupInfo</td>
<td>Filtered Lookup Info</td>
<td>If the field is a reference field type with a lookup filter, this value contains the lookup information.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>highScaleNumber</td>
<td>Boolean</td>
<td>Indicates whether the field stores numbers to 8 decimal places regardless of what's specified in the field details.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>htmlFormatted</td>
<td>Boolean</td>
<td>Indicates whether the field has been formatted for HTML and should be encoded for display in HTML. Also indicates whether a field is a custom formula field that has an IMAGE text function.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>inlineHelpText</td>
<td>String</td>
<td>The text that displays in the field-level help hover text.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>Text label that is displayed next to the field in the Salesforce user interface. This label may be localized.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>length</td>
<td>Integer</td>
<td>For string fields, the maximum size of the field in number of Unicode characters (not bytes).</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>nameField</td>
<td>Boolean</td>
<td>Indicates whether the field is a name field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>polymorphicForeignKey</td>
<td>Boolean</td>
<td>Indicates whether the field is a foreign key over a domain of multiple objects.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>precision</td>
<td>Integer</td>
<td>For double fields, indicates the maximum number of digits that can be</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>Property Name</td>
<td>Type</td>
<td>Description</td>
<td>Filter Group and Version</td>
<td>Available Version</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>reference</td>
<td>Boolean</td>
<td>Indicates whether the field is a foreign key to another record.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A reference field contains an Id value that points to a unique record (usually the parent record) on another object.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>referenceTargetField</td>
<td>String</td>
<td>For indirect lookup relationships on external objects, this property is the target custom field of the referenced object.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>referenceToInfos</td>
<td>Reference To Info[]</td>
<td>For fields that refer to other objects, this list includes information about the object types and name fields of the referenced objects.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>relationshipName</td>
<td>String</td>
<td>The name of the relationship, if this relationship is a master-detail relationship field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>required</td>
<td>Boolean</td>
<td>Indicates whether the field is required when creating or editing a record.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>To determine which fields are required in a layout, use the RecordLayoutItem.required property.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>searchPrefilterable</td>
<td>Boolean</td>
<td>Indicates whether a foreign key (relationship field) can be included in a SOSL WHERE clause.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>scale</td>
<td>Integer</td>
<td>For double fields, indicates the number of digits to the right of the decimal point.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>sortable</td>
<td>Boolean</td>
<td>Indicates whether the field is sortable. If true, then this field can be specified in a SOQL ORDER BY clause.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>unique</td>
<td>Boolean</td>
<td>Indicates whether a field’s value must be unique.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
Field Layout Component

A field in a record layout.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>The API name of the field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>componentType</td>
<td>String</td>
<td>The value is Field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The label of the field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:

Record Layout Item

Field Value

The raw and displayable field values for a field in a record.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>displayValue</td>
<td>String</td>
<td>The displayable value for a field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

This field is non-null in these cases.

- The value property can be localized to the context user’s language.
- The value property is a date or currency that can be formatted for display.
- The value property is a related record.
### Value

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Object</td>
<td>The value of a field in its raw data form. If the field is a related record, the raw value contains a nested <code>Record</code> response body. If this value is <code>null</code>, the <code>displayValue</code> is also <code>null</code>.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:  
- [Record](#)

### Filtered Lookup Info

Metadata for a lookup filter.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>controllingFields</td>
<td>String[]</td>
<td>A collection of controlling fields when the lookup filter is dependent on the source object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>dependent</td>
<td>Boolean</td>
<td>Indicates whether the lookup filter is dependent on the source object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>optionalFilter</td>
<td>Boolean</td>
<td>Indicates whether the lookup filter is optional.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:  
- [Field](#)

### Lead Status Picklist Value Attributes

Additional picklist value attributes for lead statuses.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>converted</td>
<td>Boolean</td>
<td>If this status indicates the lead has been converted, the value is <code>true</code>, otherwise <code>false</code>. For more info, see the documentation for the LeadStatus standard object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
### List Column

A column in a list.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>picklistApiName</td>
<td>String</td>
<td>The value is LeadStatus. Indicates that these value attributes are associated with the status of a Lead.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

### List Filter by Info

Information used to filter a list.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>fieldApiName</td>
<td>String</td>
<td>The API name for the field.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The label of the field.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>sortable</td>
<td>Boolean</td>
<td>Indicates whether the list column is sortable.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>fieldApiName</td>
<td>String</td>
<td>The API name for the field used to filter the list.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The label for the field used to filter the list.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>operandLabels</td>
<td>String[]</td>
<td>The values (or label if one exists) used to filter the list.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>operator</td>
<td>String</td>
<td>The filter operator. One of these values:</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Contains</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Equals</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Excludes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GreaterOrEqual</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GreaterThan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Includes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Includes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• LessOrEqual</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• LessThan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NotContain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NotEqual</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• StartsWith</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
List Order by Info

Ordering information for a list.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>fieldApiName</td>
<td>String</td>
<td>The API name for the field.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>isAscending</td>
<td>Boolean</td>
<td>Indicates whether the list column is ascending or descending.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The localized label of the field.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

List Reference

Information about the list view referenced in the metadata.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>Identity information for the list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>listViewApiName</td>
<td>String</td>
<td>The list view's api name. E.g. &quot;AllAccounts&quot;.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>objectApiName</td>
<td>String</td>
<td>A supported object, such as Account.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>The list view type.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
Most Recently Used List Metadata

List User Preference

User preferences for the list view.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>columnWidths</td>
<td>Map&lt;String, Integer&gt;</td>
<td>Column width preferences for the list view. Maps a column name to a width.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>columnWrap</td>
<td>Map&lt;String, Boolean&gt;</td>
<td>Column wrapping preferences for the list view. Maps a column name to a boolean</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>
### List View ID

Information that identifies a list view.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>The list view ID.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>objectApiName</td>
<td>String</td>
<td>The object API name for the list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>The type of the list view.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

### List View Summary

A summary of list views for an object.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>The list view’s object API name.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>ID of the list view.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The list view’s label.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>listUiUrl</td>
<td>String</td>
<td>The list view’s UI URL.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
</tbody>
</table>

SEE ALSO:

[List View Summary Collection](#)

### Location Field

A complex location value for a record field.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>latitude</td>
<td>Double</td>
<td>Gets the latitude of the location. This value maybe null if both latitude and longitude are null.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
### Match Result

A duplicate rule match. Duplicate rules are used to control whether and when you can save duplicate records within Salesforce. Duplicate rules tell Salesforce what action to take when you attempt to create a duplicate record. Each duplicate rule requires at least one matching rule to identify which existing records are possible duplicates.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>The API name of the object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>isAllowSave</td>
<td>Boolean</td>
<td>Indicates whether the rule allows a duplicate to be saved.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>matchRecordIds</td>
<td>String[]</td>
<td>The IDs of the matching records.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>matchRule</td>
<td>String</td>
<td>The developer name of the matching duplicate rule.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>objectLabel</td>
<td>String</td>
<td>The object’s label.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>objectLabelPlural</td>
<td>String</td>
<td>The object’s label in plural form.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>themeInfo</td>
<td>Theme Info</td>
<td>Information about the object’s color and icon.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:

Duplicate Record Error

### Navigation Item

The metadata for a single navigation item (tab).

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>availableInClassic</td>
<td>Boolean</td>
<td>If true, the tab is available for Salesforce Classic.</td>
<td>Medium, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>availableInLightning</td>
<td>Boolean</td>
<td>If true, the tab is available for Lightning Experience.</td>
<td>Medium, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>color</td>
<td>String</td>
<td>Color of the navigation tab.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>content</td>
<td>String</td>
<td>Launch URL of the navigation tab.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>Property Name</td>
<td>Type</td>
<td>Description</td>
<td>Filter Group and Version</td>
<td>Available Version</td>
</tr>
<tr>
<td>---------------</td>
<td>------------</td>
<td>-------------------------------------------------------</td>
<td>--------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>custom</td>
<td>Boolean</td>
<td>If true, this navigation tab is a custom tab.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>developerName</td>
<td>String</td>
<td>API name of the navigation tab.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>iconUrl</td>
<td>String</td>
<td>URL of the icon of the navigation tab.</td>
<td>Medium, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>ID of the navigation tab.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>itemType</td>
<td>String</td>
<td>The navigation tab menu type. One of these values:</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CanvasConnectedApp: The Canvas Collected App menu type. This value matches Describe App Menu value CanvasConnectedApp.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Connected App: The Connected App menu type. This value matches Describe App Menu value ConnectedApp.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Entity: The Entity menu type as found from search-driven sections. This value is not sourced from the Describe App Menu.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Standard: The Standard special case menu type as found from search-driven sections. This value matches Describe App Menu values Standard.*. The rest of the type information is in standarddType.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TabApexPage: The Visualforce tab menu type. This value matches Describe App Menu value Tab.apexPage.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TabAura: The Aura tab menu type. This value matches Describe App Menu value Tab.flexiPage.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TabSObject: The SObject tab menu type. This value matches Describe App Menu value Tab.sObject.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TabWeb: The web tab menu type. This type is only available in Lightning Experience. This value is not sourced from the Describe App Menu.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- FullSite: The Full Site tab menu type. This value is not sourced from the Describe App Menu.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Property Name | Type | Description | Filter Group and Version | Available Version
--- | --- | --- | --- | ---
label | String | Localized label name of the navigation tab. | Small, 43.0 | 43.0
objectApiName | String | Corresponding object API name of the navigation tab. | Small, 43.0 | 43.0
objectLabel | String | The label of the corresponding object. | Small, 44.0 | 44.0
objectLabelPlural | String | The plural label of the corresponding object. | Small, 44.0 | 44.0
pageReference | Page Reference | Corresponding page reference for the navigation tab. | Small, 43.0 | 43.0

- **Help**: The Help menu type. This value is not sourced from the Describe App Menu.
- **Logout**: The Logout menu type. This value is not sourced from the Describe App Menu.
- **UserProfile**: The User Profile menu type. This value is not sourced from the Describe App Menu.
- **Notification Settings**: The Notification Settings menu type. This value is not sourced from the Describe App Menu.
- **Record**: The Record menu type. This value is not sourced from the Describe App Menu.
- **ListView**: The ListView menu type. This value is not sourced from the Describe App Menu.
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>standardType</td>
<td>String</td>
<td>The subtype of a Standard menu type for the app. One of these values:</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Dashboards</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Events</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Feeds</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Groups</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Home</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MyDay</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PendingInterviews</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• People</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ProcessInstanceWorkitem</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reports</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tasks</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Topics</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• News</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DistributedMarketing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Forecasting</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Forecasting3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ForecastingLightning</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Development: Lightning Development Experience</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• AppLauncher: Lightning Experience</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DataAssessmentLightning</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DiscoveryForAccounts</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• WaveHome</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• WaveHomeLightning</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• WaveHomeLightningEacFree</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• B2bHome</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• B2bPardotCampaigns</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• B2bEmail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• B2bMarketablePeople</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• B2bAutomation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• B2bSocialSearch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• B2bContent</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• B2bPardotSettings</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• OmniSupervisorLightning</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ReactNative</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

187
### Object Action

The actions for an object.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actions</td>
<td>PlatformActionRepresentation[]</td>
<td>A list of actions associated with the object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>links</td>
<td>String[]</td>
<td>The subcontext URLs of the current request</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>url</td>
<td>String</td>
<td>The URL of the current request.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

### Object Info Directory Entry

An entry in the Object Info Directory, which is a directory of objects that User Interface API supports and that the context user can access.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>The API name of the object.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>keyPrefix</td>
<td>String</td>
<td>The key prefix of the object.</td>
<td>Small, 47.0</td>
<td>47.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The label of the object.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>labelPlural</td>
<td>String</td>
<td>The plural label of the object.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>nameFields</td>
<td>String[]</td>
<td>A list of the API names of the name fields of the object. For example, the Order object has one name field, OrderNumber. However, objects that use a first and last name have multiple name fields. For example, the Contact object has these name fields: FirstName, LastName, and Name.</td>
<td>Small, 47.0</td>
<td>47.0</td>
</tr>
</tbody>
</table>
### Opportunity Stage Picklist Value Attributes

Additional picklist value attributes for opportunity stages.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectInfoUrl</td>
<td>String</td>
<td>The URL to get a full object info response for this object</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>/ui-api/object-info/{objectApiName}</code>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>Boolean</td>
<td>Indicates whether this opportunity stage value represents a closed opportunity. Multiple opportunity stage values can represent a closed opportunity. For more info, see the documentation for the OpportunityStage standard object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>defaultProbability</td>
<td>Double</td>
<td>The default percentage estimate of the confidence in closing a specific opportunity for this opportunity stage value. May be <code>null</code> if forecasting is not enabled for the org. For more info, see the documentation for the OpportunityStage standard object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>forecastCategoryName</td>
<td>String</td>
<td>The default forecast category value for this opportunity stage value. May be <code>null</code> if forecasting is not enabled for the org. For more info, see the documentation for the OpportunityStage standard object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>picklistAttributesValueType</td>
<td>String</td>
<td>The value is <code>OpportunityStage</code>. Indicates that these value attributes are associated with the status of a Opportunity.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>won</td>
<td>Boolean</td>
<td>Indicates whether this opportunity stage value represents a won opportunity. Multiple opportunity stage values can represent a won opportunity. For more info, see the documentation for the OpportunityStage standard object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:

[Object Info Directory](#)
Page Reference

A page reference.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>attributes</td>
<td>Map&lt;String, Object&gt;</td>
<td>Values for each attribute specified by the page definition.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>state</td>
<td>Map&lt;String, Object&gt;</td>
<td>Optional parameters that are not integral to the resolution of the reference.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>Name of the corresponding page definition.</td>
<td>Small, 43.0</td>
<td>43.0</td>
</tr>
</tbody>
</table>

Picklist Value

A single picklist value.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
</table>
| attributes    | Either null or one of these response bodies:  
  - Lead Status Picklist Value Attributes  
  - Opportunity Stage Picklist Value Attributes | This property might contain a nested response body to help consumers understand the meaning of the picklist value. | Medium, 41.0 | 41.0 |
| label         | String | The displayable value of the picklist to use in a UI | Small, 41.0 | 41.0 |
| validFor      | Integer[] | If the picklist is a dependent picklist, the property contains a list of the controlling value indexes for which this value is valid. If the picklist is an independent picklist, the list is empty. | Small, 41.0 | 41.0 |
| value         | String | The value of the picklist to use in the API. | Small, 41.0 | 41.0 |

SEE ALSO:
Picklist Values

Platform Action

The metadata, layout information, and data for a platform action.
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionListContext</td>
<td>String</td>
<td>The context of the action. One of these values:</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Chatter</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Dockable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• FlexiPage</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ListView</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ListViewRecord</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Lookup</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MruList</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ObjectHomeChart</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Photo</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Record</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RecordEdit</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RelatedList</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RelatedListRecord</td>
<td></td>
<td></td>
</tr>
<tr>
<td>actionTarget</td>
<td>String</td>
<td>The URL to invoke or describe the action when the action is invoked. Applies only to quick actions.</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>actionTargetType</td>
<td>String</td>
<td>The type of the target when this action is invoked. One of these values:</td>
<td>Big, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Describe—applies to actions with a user interface, such as quick actions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Invoke—applies to actions with no user interface, such as action links or invocable actions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Visualforce—applies to standard buttons overridden by a Visualforce page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apiName</td>
<td>String</td>
<td>The API name of the action.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In API version 46.0 and later, for global actions the prefix „Global“ is prepended to the API name.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>externalId</td>
<td>String</td>
<td>External information associated with the action.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>iconUrl</td>
<td>String</td>
<td>The URL of the action’s icon image.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>The ID of this platform action record, which is the 18-character PlatformAction SObject ID, with a prefix of „0JV“.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>isMassAction</td>
<td>String</td>
<td>Indicates whether the action can be performed on multiple records.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>Property Name</td>
<td>Type</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The label to display for this action.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>primaryColor</td>
<td>String</td>
<td>The primary color of the icon for this action, in Hex color code. For example, 7F8DE1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>relatedListRecordId</td>
<td>String</td>
<td>When the actionListContext property is RelatedListRecord, this field represents the ID of the record in an object’s related list.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>relatedSourceObject</td>
<td>String</td>
<td>When the actionListContext property is RelatedList or RelatedListRecord, this field represents the API name of the related list to which the action belongs.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>section</td>
<td>String</td>
<td>The section of the user interface the action resides in. One of these values:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ActivityComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CollaborateComposer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SingleActionLinks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sourceObject</td>
<td>String</td>
<td>The object that this action is associated with. Either an API name or record ID, depending on the resource.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>subtype</td>
<td>String</td>
<td>The subtype of the action. For quick actions, one of these values:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Canvas</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CaseComment</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ChangeDueDate</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ChangePriority</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ChangeStatus</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Create</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• LightningComponent</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• LogACall</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MobileCreateFull</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MobileSmartActions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Post</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SendEmail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SocialPost</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Update</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• VisualforcePage</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Platform Action User Interface API Response Bodies
### User Interface API Response Bodies

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>targetObject</td>
<td>String</td>
<td>The target object that is created when the action is invoked.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>targetUrl</td>
<td>String</td>
<td>The target URL for custom button actions.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>The type of the action. One of these values:</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

- **CustomButton**—A button that opens a URL or Visualforce page or executes JavaScript.
- **ProductivityAction**—A pre-defined Salesforce action, attached to a limited set of objects.
- **QuickAction**—A global or object-specific action.
- **StandardButton**—A pre-defined Salesforce button, such as **New**, **Edit**, and **Delete**.

For custom buttons, one of these values:
- flow
- javascript
- page
- sControl
- url

For action links, one of these values:
- Api
- ApiAsync
- Download
- Ui

Standard buttons and productivity actions have no subtype.

1. The format for a quick action is:

   \[
   \text{OrgId:SourceObject::Context:deviceFormat::QuickActionDefinitionId}
   \]

   For example:

   \[
   00Dxx0000001gGh:x01xx0000000007AAA::Record:Phone:09Dxx00000000B6
   \]

   The format for a standard action is:

   \[
   \text{OrgId:SourceObject::Context:deviceFormat::StandardButton:ApiName}
   \]

### Record Exceptions

A collection of record exception errors.
### Record Exception Error

Information about a record exception error.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>constituentField</td>
<td>String</td>
<td>If the field is a constituent of a compound field, this property contains the API name of the constituent field, and the <code>field</code> property contains the API name of the compound field. If the field is not a constituent of a compound field, or if no field applies to the error, the value is <code>null</code>.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>duplicateRecordError</td>
<td>Duplicate Record Error</td>
<td>Information about possible duplicate records. When the error code is <code>DUPLICATES_DETECTED</code>, this property might contain a value.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>errorCode</td>
<td>String</td>
<td>An error status code.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>field</td>
<td>String</td>
<td>A field API name. If no field applies to the error, the value is <code>null</code>.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>fieldLabel</td>
<td>String</td>
<td>A field label. If no field applies to the error, the value is <code>null</code>.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>message</td>
<td>String</td>
<td>An error message.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- **Record Exceptions**

---

### Record Layout Component

A concrete record layout component.
### Record Layout Item

An item in a record layout.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>If a field powers this component, this property contains the API name of the field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>componentType</td>
<td>String</td>
<td>One of these values:</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Canvas</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CustomLink</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• EmptySpace</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Field</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ReportChart</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• VisualforcePage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>editableForNew</td>
<td>Boolean</td>
<td>Indicates whether the item can be edited when creating a new record.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>editableForUpdate</td>
<td>Boolean</td>
<td>Indicates whether the item can be edited when updating a record.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>label</td>
<td>String</td>
<td>The text label for the item.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>layoutComponents</td>
<td>Collection of components</td>
<td>A collection of components that make up the item.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Canvas Layout Component</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Custom Link Layout Component</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Field Layout Component</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Report Layout Component</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Record Layout Component</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Visualforce Layout Component</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lookupIdApiName</td>
<td>String</td>
<td>The ID field name of a lookup field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
### Record Layout Row

A row in a record layout.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>layoutItems</td>
<td>Record Layout Item[]</td>
<td>A collection of items in the row, from left to right. Might not extend to the last column.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- [Record Layout Section](#)

### Record Layout Section

A section in a record layout.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>collapsible</td>
<td>Boolean</td>
<td>Indicates whether the section can be collapsed.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>columns</td>
<td>Integer</td>
<td>The number of columns in the section.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>heading</td>
<td>String</td>
<td>The heading text for the section.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>The ID of the section.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>layoutRows</td>
<td>Record Layout Row[]</td>
<td>A collection of the rows in the section.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>rows</td>
<td>Integer</td>
<td>The number of rows in the section.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

### Record Layout Item

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>required</td>
<td>Boolean</td>
<td>Indicates whether the field is required in a layout when creating or updating a record. This information is useful to know if you wanted to render required fields with a different treatment, such as a red outline.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>sortable</td>
<td>Boolean</td>
<td>Reserved for future use.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- [Record Layout Row](#)
**Record Layout Section User State**

The user state information for a record layout section.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>useHeading</td>
<td>Boolean</td>
<td>Indicates whether the heading text is expected to be displayed.</td>
<td>Medium, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>collapsed</td>
<td>Boolean</td>
<td>Indicates whether the section is initially displayed as collapsed (<code>true</code>) or expanded (<code>false</code>).</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>The ID of a layout section.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

**Record Layout User State**

The user state information for a record layout.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>The ID of a layout.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>sectionUserStates</td>
<td>Map&lt;String, Record Layout Section User State&gt;</td>
<td>A map of layout section user states, keyed to section IDs.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

**Record Type Info**

Information about a record type.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>available</td>
<td>Boolean</td>
<td>Indicates whether this record type is available to the context user when creating a record.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>
## Default Record Type Mapping

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>defaultRecordTypeMapping</td>
<td>Boolean</td>
<td>Indicates whether this record type mapping is the default for the associated object.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

## Master

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>master</td>
<td>Boolean</td>
<td>Indicates whether this record type is the master record type. The master record type is the default record type that's used when a record has no custom record type associated with it.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

## Name

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The record type's label name.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

## Record Type ID

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordTypeId</td>
<td>String</td>
<td>The ID of the record type.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

---

**SEE ALSO:**
- [Record](#)
- [Object Info](#)

## Reference To Info

Information about a reference field's referenced types and the name field names of those types.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>The object API name of a type being referenced in a relationship.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>nameFields</td>
<td>String[]</td>
<td>A collection of names of the name fields for this object type. Combined with the relationship name, you can determine how to query this type's name fields, for example, Parent.Name. Typically there is one name field per object, except where FirstName and LastName fields are used.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

---

**SEE ALSO:**
- [Field](#)

## Report Layout Component

A report chart component on a record layout page.
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>If a field powers this component, this property contains the API name of the field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>cacheData</td>
<td>Boolean</td>
<td>Indicates whether to cache data (true) or not (false).</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>componentType</td>
<td>String</td>
<td>The value is ReportChart.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>error</td>
<td>String</td>
<td>An error message.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>filter</td>
<td>String</td>
<td>A filter for the report.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>hideOnError</td>
<td>Boolean</td>
<td>Indicates whether to hide the component when there's an error (true) or not (false).</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>placeholder</td>
<td>String</td>
<td>A placeholder for the report.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>reportId</td>
<td>String</td>
<td>The component’s report ID.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>showTitle</td>
<td>Boolean</td>
<td>Indicates whether to show the title of the report (true) or not (false).</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>size</td>
<td>String</td>
<td>The size of the component.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:

Record Layout Item

Theme Banner

A theme banner image. Theme banners have a higher aspect ratio than theme images. The banner size may change slightly, so code accordingly.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>fullSizeUrl</td>
<td>String</td>
<td>The URL of the banner image.</td>
<td>Small, 42.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

SEE ALSO:

Theme

Theme Image

A theme image at three sizes: small, medium, and large. Specific sizes may change slightly, so code accordingly.
### Theme Info

Color and icon information for a theme.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>String</td>
<td>Color</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>iconUrl</td>
<td>String</td>
<td>Icon URL</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:  
  - [Theme](#)

### Visualforce Layout Component

A Visualforce component on a record layout page.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Filter Group and Version</th>
<th>Available Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiName</td>
<td>String</td>
<td>If a field powers this component, this property contains the API name of the field.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>componentType</td>
<td>String</td>
<td>The value is VisualforcePage.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>height</td>
<td>String</td>
<td>The height of the component.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>showLabel</td>
<td>Boolean</td>
<td>Indicates whether to show the section label with this Visualforce component (true) or not (false).</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>showScroll</td>
<td>Boolean</td>
<td>Indicates whether to show the scroll bar on the canvas component (true) or not (false).</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>visualforceUrl</td>
<td>String</td>
<td>The Visualforce URL.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>Property Name</td>
<td>Type</td>
<td>Description</td>
<td>Filter Group and Version</td>
<td>Available Version</td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>-------------------------------</td>
<td>--------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>width</td>
<td>String</td>
<td>The width of the component.</td>
<td>Small, 41.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

SEE ALSO:
- Record Layout Item
Index

List View Records and Metadata 112
List View Summary 183
List View Summary Collection 155
List Views 39
Location Field 183
Lookup Field Actions 92
Lookup Values 155

M
Match Result 184
Most Recently Used List Metadata 156
Most Recently Used List Record Collection 157
Most Recently Used List UI 157
Most Recently Used List View Metadata 119
MRU List View Actions 94

N
Navigation Items 158
NavItemRepresentation 184
Node.js 12

O
Object Action 188
Object Info 158
Object Info Directory 161
Object Info Directory Entry 188
Object Layout, Get. 48
Opportunity Stage Picklist Value Attributes 189

P
PageReferenceRepresentation 190
Picklist Value 190
Picklist Values 161
Picklist Values Collection 162
Platform Action 190

R
React 12
React Native 12
Record 162
Record Collection 163
Record Defaults 164
Record Detail Page Actions 74
Record Edit Page Actions 76
Record Exception 193
Record Exception Error 194
Record Input 145
Record Layout 164
Record Layout Component 194

Record Layout Item 195
Record Layout Row 196
Record Layout Section 196
Record Layout User State 197
Record Type Info 197
Record UI 165
Record UI Section User State 197
Record UI, Get 45
Records, Batch Get 56
Records, Get 44, 53, 66
Records, Get a Page 59
Records, Post 59
Records, Update 64
Records, working with 25
Reference To Info 198
Related List Actions 78
Related List Record Actions 80
Report Layout Component 198
Request bodies
  Favorite Batch Collection Input 141
  Favorite Batch Input 142
  FavoriteInputRepresentation 142
  Record Input 145

Resources
  child relationships, get 51
  create a favorite 99
  delete favorite 103
  enter lowercase resource name 116, 120, 127, 129, 131, 133
  error responses 23
  get active themes 140
  get default values to clone a record 61
  get default values to create a record 62
  get directory of supported objects 49
  get favorite 100
  get favorites 101
  get list views for an object 105
  get lookup search results 66
  get lookup search results for an object 69
  get object metadata 50
  get picklist values 51
  get values for a picklist field 50
  global actions 72
  group photo actions 96
  Lightning page actions 90
  list view chart actions 88
  list view header actions 82
  list view metadata 110
  list view record action 85
  list view records 106
Index

Resources (continued)
  list view records and metadata 112
  lookup field actions 92
  most recently used list view metadata 119
  MRU list view actions 94
  object layout, get 48
  record detail page actions 74
  record edit page actions 76
  record UI, get 45
  records, batch get 56
  records, get 44, 53, 66
  records, get a page 59
  records, post 59
  records, update 64
  related list actions 78
  related list record actions 80
  update a batch of favorites 103
  update a favorite 101
  update usage of a favorite 104
Response bodies
  Action 149
  Apps 151
  Batch Result Item 169
  Batch Results 152
  Canvas Layout Component 170
  Child Relationship 172
  Custom Link Layout Component 173
  Duplicate Record Error 174
  Error Message 174
  Error with Output 174
  Favorite 152
  favorite collection 153
  Field 175
  Field Layout Component 179
  Field Value 179
  Filtered Lookup Info 180
  Lead Status Picklist Value Attributes 180
  List Filter by Info 181
  List Column 181
  List Metadata 153
  List Order by Info 182
  List Record Collection 154
  List Reference 182
  List UI 155
  List User Preference 182
  List View ID 183
  List View Summary 183
  list view summary collection 155
  Location Field 183
Response bodies (continued)
  Lookup values 155
  Match Result 184
  Most Recently Used List Metadata 156
  Most Recently Used List Record Collection 157
  Most Recently Used List UI 157
  Navigation Items 158
  NavItemRepresentation 184
  Object Action 188
  Object Info 158
  Object Info Directory 161
  object info directory entry 188
  Opportunity Stage Picklist Value Attributes 189
  PageReferenceRepresentation 190
  Picklist Value 190
  Picklist Values 161
  Picklist Values Collection 162
  Platform Action 190
  Record 162
  Record Collection 163
  Record Defaults 164
  Record Exception 193
  Record Exception Error 194
  Record Layout 164
  Record Layout Component 194
  Record Layout Item 195
  Record Layout Row 196
  Record Layout Section 196
  Record Layout Section User State 197
  Record Layout User State 197
  Record Type Info 197
  Record UI 165
  Reference To Info 198
  Report Layout Component 198
  Theme 166
  Theme Banner 199
  Theme Image 199
  Theme Info 200
  Visualforce Layout Component 200

S
  safelist 12
  sample apps 12
  supported objects 12

T
  Theme 166
  Theme Banner 199
  Theme Image 199
Theme Info 200
Themes, Active 140
Time 20

U
Update a Batch of Favorites 103
Update a Favorite 101
Update Usage of a Favorite 104
Upload binary files 33

User Interface API quick start 2

V
Versioning 20
Visualforce Layout Component 200

W
whitelist 12