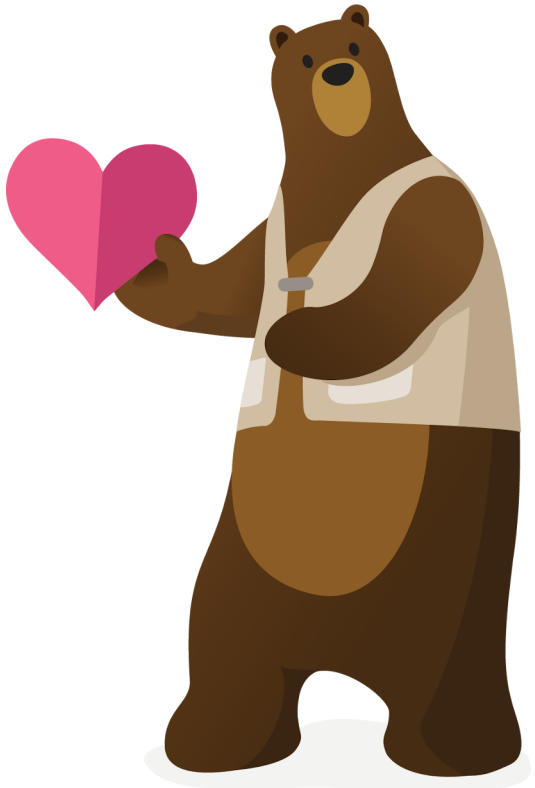
Design Without Limits

Salesforce, Spring '21



'21

@salesforcedocs

Last updated: December 1, 2020

CONTENTS

DESIGN FOR ENTERPRISE SCALE

As architects and developers, you know that enterprise app usage can rapidly accelerate, placing demands on an org. But with good planning and efficient coding techniques, your app can stay responsive.

Have No Fear of Limits

Limits are a major concern for architects and developers of large-scale deployments. Of course, limits are out there: design-time limits, like excessive custom objects, and run-time limits, such as too many synchronous callouts. But with the following solutions, you can design your app so that users never ever hit them.

Limits protect your org. Because Salesforce orgs live in a multi-tenant environment, we enforce limits to ensure that applications or processes run by other customers don't impact the performance of your org.

We do have a few critical limits. Learn what they are before you start designing your application to avoid hitting them. Some limits vary based on your edition. Know the threshold for your org, and design for it.

Design Your Development Life Cycle and Code

Important parts of the design process include writing, testing, and deploying your app. However, it's easy to overlook or postpone this part of the process. This guide offers pointers to resources to help you include those best practices early in the life cycle of your app.

Use These Best Practices

Before you start to build your application, think about the demand that your app's usage will have on your org. Concerned that your application will encounter common limit thresholds? This guide gives architects and lead developers solutions for the most common app usage patterns. Pair these techniques with good development strategies to build apps that never encounter limits.

MANY SIMULTANEOUS USERS WITH CALLOUTS TO EXTERNAL SYSTEMS

For example, in a call center application, many users are generating synchronous transactions in a Visualforce page. If the application integrates with other systems by using callouts, you have lots of callouts running simultaneously. When the response from the callout takes longer than 5 seconds, you reach the concurrent long-running transaction limit.

Other Examples

- Ordering System
- Large Customer Community
- Human Resources Hub

Considerations

- Governor Limits
- Long-running Callouts
- Many Custom objects

Bulkify

The number of API calls per 24-hour period is limited, so many Salesforce APIs support the idea of sending bulk data to a single call.

Data Manipulation Language (DML) operations in Apex and APIs support sending an array of DML changes, rather than having to make a separate DML call for each record change.

See the following for more ideas and best practices to bulkify your code:

- The Bulk API Introduction
- Loading Large Data Sets with the Bulk API
- Maximizing Parallelism and Throughput Performance
- Understanding the Bulk of the Salesforce Mobile Platform
- Apex Developer Resources (especially Apex Code Best Practices, Apex Design Patterns, and Trailhead links)
- Efficient Use of APIs from Heroku

Single or Chain Continuation

Delayed responses from a service can pile up and soon you've hit your concurrent request limit.

Implement a system that anticipates this behavior and avoids the concurrent request limit.

See the following for more ideas and best practices for implementing continuation:

- Avoiding Apex Speeding Tickets (Concurrent Request Limits via Synchronous Callouts)
- Chaining Asynchronous Callouts

For example, a Sales Cloud application connects to a legacy on-premise data store. A middleware-style integration connects the data together to replicate changes from one system in the other. This middleware makes many requests for information, especially when the transaction volume is large, and can encounter the daily API call threshold for your org.

Other Examples

- Integrations that continuously ping Salesforce
- Apps with long-running or large queries, especially with multiple users

Considerations

- Bulk API limits
- Concurrent API Requests
- API Requests per 24-hour period
- Triggers and Workflows when migrating data

Streaming API

For existing apps that poll the SOAP or REST API to detect changes, consider refactoring the app to use the Streaming API for better performance and lower resource consumption than frequent polling.

See the following for more ideas and best practices for implementing the Streaming API:

- Streaming API Developer Guide
- Streaming API Trailhead Module

Platform Cache

The Platform Cache layer provides faster performance and better reliability when caching Salesforce session and org data.

Specify what to cache and for how long without using custom objects and settings or overloading a Visualforce view state. Platform Cache improves performance by distributing cache space so that some applications or operations don't steal capacity from others.

See the following for more information about Platform Cache:

- Platform Cache Overview
- Platform Cache Basics Trailhead Module

Idempotence

Design your operations to produce the same outcome, whether they are executed once or multiple times.

See the following for Idempotent operation examples and guidance:

- Implementing Idempotent Operations with Salesforce
- Integration Patterns and Practices

Apex Web Services

Lets you consolidate a bunch of API calls into one, and expose the logic as a web service. With this unified interface, you'll reduce the volume and frequency of API calls within a single org while still providing remote access to data.

See the following for more information on Apex Web Services:

- Apex Web Services and Callouts Summary Article
- Web Service Methods
- Apex Web Services Trailhead Module

For example, an Salesforce Platform application sends three orders to three different fulfillment systems for every transaction entered.

This spawns lots of concurrent requests. The design might include many custom objects, and require tight coordination between systems and subscriber orgs.

Other Examples

- Remote Order System
- Billing Data Importing
- On-premise Data Warehouse

Considerations

- Concurrent API Requests
- Governor Limits
- Many External Objects

Integration Best Practices

Use Visualforce or Workflow-driven outbound messaging, compare SOAP and REST API benefits, employ best practices, and learn which scenarios to avoid.

See the following for a variety of system integration best practices:

- Integration Patterns and Practices

Single or Chain Continuation

Delayed responses from a service can pile up and soon you've hit your concurrent request limit.

Implement a system that anticipates this behavior and avoids the concurrent request limit.

See the following for more ideas and best practices for implementing continuation:

- Avoiding Apex Speeding Tickets (Concurrent Request Limits via Synchronous Callouts)
- Chaining Asynchronous Callouts

Streaming API

For existing apps that poll the SOAP or REST API to detect changes, consider refactoring the app to use the Streaming API for better performance and lower resource consumption than frequent polling.

See the following for more ideas and best practices for implementing the Streaming API:

- Streaming API Developer Guide
- Streaming API Trailhead Module

Platform Events

Platform events are part of Salesforce's event-driven enterprise messaging platform for large distributed systems.

Leverage a publish-subscribe model to communicate changes in one system to another system. Create new events similarly to how you create new objects, and notify subscribers when a change occurs in near-real time. This approach reduces the number of point-to-point integrations.

See the following to get started using platform events:

- Platform Events Developer Guide
- Platform Events Trailhead Module

Salesforce Connect

The right solution might be our own Salesforce Connect product for integration of data stored outside your Salesforce org.

Salesforce Connect maps external objects to data tables in external systems, for on-demand access. Mapping reduces the need for copying or importing the data.

See the following to learn more about Salesforce Connect:

- Salesforce Connect Overview
- Salesforce Connect Trailhead Module
- General Limits for Salesforce Connect

For example, you've acquired a company that has its own Salesforce org. To ensure seamless integration and continuous productivity, you want to move all the related Sales and Service data from the acquired org into your org.

This could include loading very large files or many records.

Other Examples

- Inventory app
- Document management
- Large email volume with attachments
- Initial data load

Considerations

- Storage limits
- Large data requests

Big Objects

Use Big Objects, Salesforce's own solution for managing large amounts of data on the Salesforce platform. Use it for long-term tracking, extended data models, or archiving of immutable data. Process Big Objects with Async SOQL, SOQL, Bulk, REST and SOAP APIs.

- Big Objects Implementation Guide

Best Practices for Deployments with Large Data Volumes

Use suggested techniques and learn from case studies to store and manage data efficiently, including Skinny Tables.

- Best Practices for Deployments with Large Data Volumes
- Fifteen Things to Consider Before Your Next Data Migration

Best Practices for Deployments with Large Data Volumes

When you're dealing with hundreds of millions of records, PK chunking leverages the Bulk API to efficiently chunk records with sequential IDs. Then submit separate queries to extract the data in each chunk and combine the results.

- Use PK Chunking to Extract Large Data Sets from Salesforce

Data Archiving

Any implementation using a large scale data set definitely needs a backup, archiving, and restoration strategy.

Whether you build one yourself or leverage a third-party solution, data archiving frees up necessary storage and gives you expanded control over data versioning and recovery.

Find the right archiving solution with the following information:

- Backup and Restore Essentials
- Heroku-based Repository
- Backup Solutions on AppExchange
- How to Monitor Storage

Custom Metadata Types

When you're merging orgs together, reduce your reliance on custom objects by defining custom metadata types.

Custom metadata types can define the custom fields to associate with records. They're reusable and package deployable for smaller deployments.

- Custom Metadata Types Help
- Custom Metadata Types Basics Trailhead Module

For example, you share a weekly sales report with a lot of filters to narrow the scope of the report. Over time, the number of objects, users and filters has increased exponentially.

The report now takes many times longer to render results than it initially did. You need to fine tune it so you spend more time looking at the results than running the report.

Other Examples

- Frequently refreshed dashboards
- Many dashboards on commonly used pages
- Reports on large objects

Considerations

- Slow reports
- Slow queries

Build Efficient Reports and Tune Existing Ones

You can create more efficient reports and list views in several ways, such as avoid negative filters because it takes longer to analyze what doesn't match a query compared to what does match. For existing reports and dashboards, use Iora Lite, our free diagnostic tool.

See the following for tips to improve report and dashboard performance:

- Build Reports That Fly
- Salesforce Field Indexes
- Use Iora Lite to Diagnose and Tune Slow-Running Dashboards and Reports
- Improve Report Performance

Query Optimization

Write efficient and selective queries. You can create rich reports with many data joins or expansive parameters, but this approach becomes inefficient as you accumulate more users and data grows.

Salesforce has to perform more analysis to match records. A few optimization techniques can save a lot of processing time.

See the following to improve the efficiency of your queries:

- Query & Search Optimization Cheatsheet
- Developing Selective Queries
- Query Optimizer Webinar Slide Deck
- Improving Query Performance for Salesforce Knowledge Articles
- Query Plan Tool

For example, you have Apex triggers that send emails to inform your support team every time someone updates a customer case.

Other Examples

- Frequent email marketing campaigns
- Notifications to a mass audience

Considerations

- Mass email limit
- Push notifications limit
- Daily email limit

Marketing Cloud

If you're planning a large email marketing campaign, use Salesforce Marketing Cloud. It has several editions, including one focused on email and mobile messaging.

Learn about the Salesforce Marketing Cloud:

- Salesforce Marketing Cloud Overview

Chatter

Chatter has its own API, is designed for frequent communication, and centralizes information. Use it to notify your users instead of email or push notifications. Leverage it in your org, or build Chatter features into your app.

Get started using and leveraging Chatter:

- Learn to develop with Chatter
- Chatter Basics Trailhead Module

Temporary Email Limit Increase

Yes, you can request a temporary increase to your mass email limit under some circumstances. You cannot get an increase for large marketing campaigns (use Marketing Cloud instead).

Here's how you can increase your email limit:

- Increase Your Daily Mass Email Limit

HIGH VOLUME OF ACTIVITIES (TASKS AND EVENTS)

For example, your regional sales teams share contacts, leads, and opportunities. They track meetings with customers and related files. You're building your own contact management app. As your customer base and sales teams grow, you need your app to stay responsive.

Other Examples

- Calendar apps
- Contact archiving

Considerations

- Ownership data skew
- Slow updates
- Calendar sync load
- Slow archiving

Manage Record Ownership

When you have concentrated ownership of data so that a single user or all members of a single role or public group owns most or all the records for a particular object (like hundreds of thousands of contacts for one account), devise a strategy for distributing the ownership of records across a greater number of users.

See the following to learn how to better manage record access:

- Designing Record Access for Enterprise Scale
- Avoid Account Data Skew for Peak Performance
- Managing Lookup Skew in Salesforce to Avoid Record Lock Exceptions
- Record-Level Access: Under the Hood

Sharing Rule Best Practices

Record sharing can have a big Considerations on org performance. A single owner with many records (10,000+) can prompt long recalculations when records or sharing changes. Although no single sharing model fits all needs, you can implement the right sharing model for your org.

Focus on sharing architecture with the following resources:

- A Guide to Sharing Architecture
- Behind the Scenes of Record Ownership in Salesforce
- Designing Record Access for Enterprise Scale

Data Archiving

Any implementation using a large scale data set definitely needs a backup, archiving, and restoration strategy.

Whether you build one yourself or leverage a third-party solution, data archiving frees up necessary storage and gives you expanded control over data versioning and recovery.

Find the right archiving solution with the following information:

- Backup and Restore Essentials
- Heroku-based Repository
- Backup Solutions on AppExchange
- How to Monitor Storage

For example, you quickly grow your development team to 25 developers, who now support your existing Sales Cloud application and your new, very customized Service Cloud application.

Other Examples

- Team development
- New feature development
- Integration with external apps and systems
- Large deployments

Considerations

- Governor limits
- Concurrent requests
- Long-running transactions
- Multiple teams, lots of triggers, complex testing
- Apex features that duplicate declarative features

Efficient Coding and Process

Working on large projects, sometimes with multiple teams, has its own complexity.

Make sure that your design process takes into account:

- Release management
- Sandbox strategy
- Source code control
- Integration strategy

Most importantly, avoid redundancy. Sometimes developers use Apex for functionality, like field-validation, when Salesforce provides a declarative solution, such as validation rules. Or multiple teams develop a project. These individual teams might build triggers without knowing and testing each other's code. It's important that these teams coordinate to avoid redundant or inefficient code. Redundant code counts toward Apex governor limits.

Use the following resources to guide you through your Apex development:

- Apex Developer Resources
  - Apex Code Best Practices
  - Apex Design Patterns
- Trigger Frameworks and Apex Trigger Best Practices
- Apex Execution Governors and Limits
- Testing Apex
- Salesforce DX

Idempotence

Design your operations to produce the same outcome, whether they are executed once or multiple times.

See the following for Idempotent operation examples and guidance:

- Implementing Idempotent Operations with Salesforce
- Integration Patterns and Practices

Batch Apex

Divide large processing tasks into manageable pieces with Batch Apex. Salesforce provides this programmatic interface to create a queue of jobs or schedule jobs to run at a specific time when demand isn't so high.

See the following guide to learn all about Batch Apex:

- Batch Apex

CRITICAL LIMITS

Our documentation has a lot of information about limits, and some features have their own limits. You won't encounter nearly all of them. We'll cut to the chase here. These are the critical limits for an enterprise-scale app.

- API Request Limits
- Apex Execution and Governor Limits
- Data and File Storage Allocations

	Personal Edition	Contact Manager	Group Edition	Professional Edition	Enterprise Edition	Unlimited and Performance Editions	Developer Edition
Custom Objects	N/A	5	50	50	200	2,000	400
Custom Fields per Object	100	25	100	100	500	800	500
Territories: maximum account assignment rules each	N/A	N/A	N/A	N/A	15	15	15
Permission sets: maximum (created)	N/A	1	1	10	1,000	1,000	1,000
Permission sets: maximum (created and added as part of an installed managed AppExchange package)	N/A	N/A	1,500	1,500	1,500	1,500	1,500
Divisions	N/A	N/A	N/A	100	100	100	N/A
External Objects	N/A	N/A	N/A	N/A	100	100	100
Certificates	N/A	N/A	N/A	N/A	50	50	50
Formulas: VLOOKUP functions per object	10	10	10	10	10	10	10

Sharing Rules

You can create up to 300 sharing rules per object, including up to 50 criteria-based rules (not in personal, contact, or group).

Joins per query across external objects (and other types of objects)

4

Workflow rules

500 per object. 2,000 per org.

This limit applies to any combination of workflow, assignment, auto-response, and escalation rules, both active and inactive.