Salesforce

Declarative development – minimal code

Heroku – developers can build highly scalable Web apps and back-end services. Includes DB connectors

Built on AWS services

Heroku Connect connects Salesforce data to Heroku Postgres data

Salesforce APIs – integrate with Salesforce from external apps

| **API** | **What you can do with it** |
| --- | --- |
| SOAP API | Integrate your org’s data with other applications using standard SOAP protocols. |
| REST API | Access objects in your org using standard REST protocols. |
| Metadata API | Manage customizations in your org and build tools that manage your metadata model. |
| Tooling API | Build custom development tools for platform applications. |
| Marketing Cloud API | Expose Marketing Cloud capabilities with the REST API and get comprehensive access to most email functionality with the SOAP API. |
| Bulk API | Load, delete, and perform asynchronous queries on large data sets. |
| Streaming API | Send and receive notifications securely and efficiently. Notifications can reflect data changes in your org, or custom events. |
| Connect REST API | Build UI for Commerce, CMS-Managed Content, Experience Cloud Sites, Files, Notifications, Topics, and more. |
| Mobile SDK | While it’s technically a software development kit, it’s worth including here. Integrate Native or Hybrid mobile apps directly with Salesforce. |

Mobile SDK – build native, HTML5 and hybrid apps

Setup > Schema Builder – shows you an ERD model of your objects

Programming Technologies in Salesforce:

Lightning Component Framework – UI development framework (similar to Angular or React)

Each component consists of a bundle

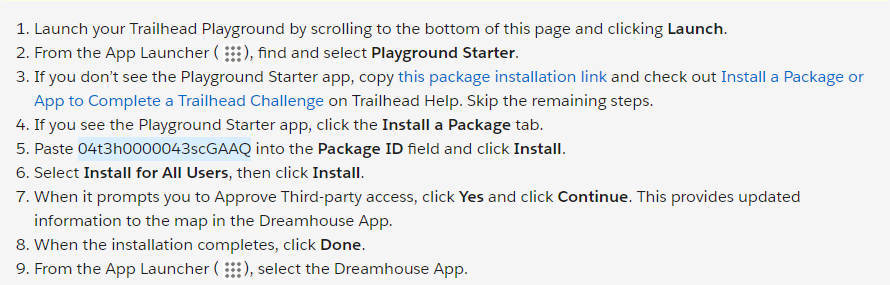
Mobile ready

Apex – Salesforce proprietary language

Visualforce – HTML-like markup language. You can include Apex and Javascript

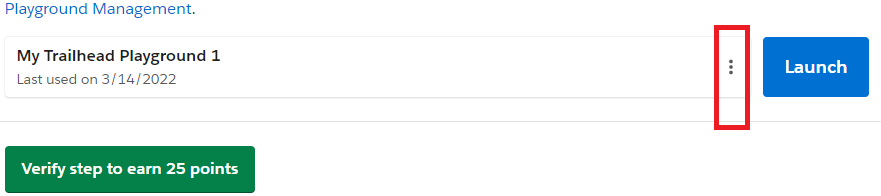
SOQL – Salesforce Query Language

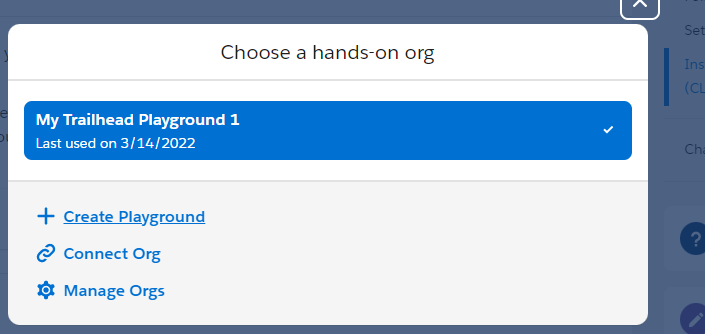
Org is a specific instance of Salesforce



To Create a new Playground or switch to another org:

Click the 3 dots:





To connect to the Playground from VS Code or CLI you will need to do a password reset so you can associate a password to the account. Then you connect to the playground using the password

1. Install the CLI

<https://developer.salesforce.com/docs/atlas.en-us.sfdx_setup.meta/sfdx_setup>

Test in CMD : sfdx or sfdx update

1. Open (or install) VS Code
2. Extensions: install Salesforce Extension Pack
3. Test in VS: ctrl + shift + P to open command palette or from Menu: View > command palette

Sfdx when you type this you should see a list of sfdx commands to choose from

1. <https://developer.salesforce.com/tools/vscode/en/getting-started/java-setup>

NOTE: Our environments have unique user ids with different extensions

Example:

URL: <https://ctdss--venus.my.salesforce.com>

Username: [heather.lynch@ct.gov](mailto:heather.lynch@ct.gov).venus

Optional:

Install Docker

<https://hub.docker.com/editions/community/docker-ce-desktop-windows>

Code Samples:

<https://developer.salesforce.com/code-samples-and-sdks>

<https://developer.salesforce.com/docs/component-library/overview/components>

1. From the gear icon ( The gear icon to open Setup.), click **Setup** to launch the setup page. We use Setup a lot, so remember this step!
2. Click the **Object Manager** tab.
3. Click **Contact**.
4. Under Fields & Relationships, click **New**.
5. A data type indicates what kind of information your field holds. For this field, pick Checkbox and click **Next**.
6. The Field Label is what you see on the Contact page. Enter Prequalified? and click **Next**.
7. Click **Next** again.
8. Check the checkbox to add the new field to all the Contact Page Layouts and then click **Save**.

You just customized your first object. Great job!

Let’s take a look at what we did. From the App Launcher ( App Launcher icon), find and select **Contacts**. Use the arrow The list view arrow icon. to view All Contacts and click a contact name. Under the Details tab, you can see your new field. Now it’s easier for Michelle and the other brokers to log and retrieve this important piece of client information.

Heroku platform – build highly scalable we apps and back end services using python…also has database tools

Lightning Component Framework – UI development (desktop and mobile) similar to Angular

Aura Components are the original? They can coexist with lightning components

Lightning components are mobile ready

They consist of client side js and server-side Apex controllers

Setup > search for Lightning, select lightning components

Displays a list (different for diff apps. Possibly need to install components)

Select the component

Developer console is the IDE you can use to develop, debug and test code

Note: YOU CAN ALWAYS ACCESS DEVELOPER CONSOLE BY CLICKING THE GEAR

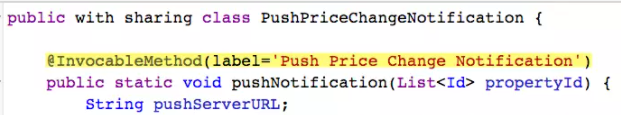
Apex – Salesforce’s programming language

Process Building – used to build flows

Can be extended with code:

Setup > search for Process Builder

NOTE: PROCESS BUILDER WILL BE DEPRECATED…EVERYTHING SHOULD BE BUILT IN FLOW BUILDER



@InvocableMethod label is the name other processes can use to execute this method)

SOQL (Salesforce Object Query Language)

https://developer.salesforce.com/docs/atlas.en-us.224.0.apexcode.meta/apexcode/langCon\_apex\_SOQL.htm

Account result = [SELECT ID,Name, (SELECT Id, Name FROM Contacts)

FROM Account

WHERE ID = :accountId];

**Visualforce**

Used to create pages whereas lightning components are things that go on pages

<https://trailhead.salesforce.com/en/modules/lex_dev_overview/units/lex_dev_overview_future>

<https://developer.salesforce.com/docs/atlas.en-us.224.0.lightning.meta/lightning/intro_framework.htm>

<https://developer.salesforce.com/docs/component-library/documentation/lwc>

<https://developer.salesforce.com/docs/atlas.en-us.224.0.apexcode.meta/apexcode/>

<https://developer.salesforce.com/docs/atlas.en-us.224.0.pages.meta/pages/pages_intro.htm>

APIs are automatically created for properties…including custom properties

<https://trailhead.salesforce.com/modules/api_basics>

| **API** | **What you can do with it** |
| --- | --- |
| SOAP API | Integrate your org’s data with other applications using standard SOAP protocols. |
| REST API | Access objects in your org using standard REST protocols. |
| Metadata API | Manage customizations in your org and build tools that manage your metadata model. |
| Tooling API | Build custom development tools for platform applications. |
| Marketing Cloud API | Expose Marketing Cloud capabilities with the REST API and get comprehensive access to most email functionality with the SOAP API. |
| Bulk API | Load, delete, and perform asynchronous queries on large data sets. |
| Streaming API | Send and receive notifications securely and efficiently. Notifications can reflect data changes in your org, or custom events. |
| Connect REST API | Build UI for Commerce, CMS-Managed Content, Experience Cloud Sites, Files, Notifications, Topics, and more. |
| Mobile SDK | While it’s technically a software development kit, it’s worth including here. Integrate Native or Hybrid mobile apps directly with Salesforce. |

APIs are used for the internal access

Heroku is used for web apps accessed by external customers

It is built on AWS

It is a web development platform

Can use Java, Python, PHP

Heroku has a Postgres database which you can connect to Salesforce data (no need to move it across platforms)

<https://trailhead.salesforce.com/modules/api_basics>

<https://developer.salesforce.com/>

Setup > Object Manager

Create objects

Select object > Fields and Relationships to add data fields to the objects

Schema Builder – shows relationships between objects. Can add properties here too

**Data Imports**

<https://help.salesforce.com/apex/HTViewHelpDoc?id=field_mapping_for_other_data_sources_and_organization_import.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=faq_import_general.htm&language=en_US>

<https://help.salesforce.com/apex/HTViewHelpDoc?id=import_prepare.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=data_loader.htm&language=en_US>

<http://pages.mail.salesforce.com/achievemore/managedata/?utm_source=trailhead&utm_medium=resources&utm_campaign=datamanagement&utm_content=importdata>

<http://salesforce.vidyard.com/watch/ARIjWm2qrDkJVJxEhReFug?_ga=2.144864564.1257339476.1646661535-1766590648.1646405561>

**Data Exports**

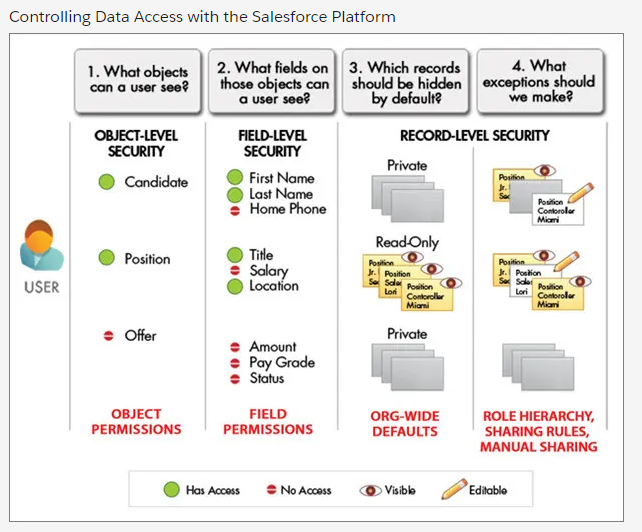
<https://help.salesforce.com/HTViewHelpDoc?id=data_loader.htm&language=en_US>

<https://help.salesforce.com/apex/HTViewHelpDoc?id=installing_the_data_loader.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=exporting_data.htm&language=en_US>

<http://pages.mail.salesforce.com/achievemore/managedata/?utm_source=trailhead&utm_medium=resources&utm_campaign=datamanagement&utm_content=exportdata>

<http://pages.mail.salesforce.com/page.aspx?QS=3935619f7de112ef3fdd44cfb3dcb15df8ef83257379ba31&utm_source=trailhead&utm_medium=resources&utm_campaign=072016>



<https://help.salesforce.com/HTViewHelpDoc?id=users_licenses_overview.htm&language=en_US>

<https://help.salesforce.com/apex/HTViewHelpDoc?id=controlling_login_access.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=admin_loginrestrict.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=admin_delegate.htm&language=en_US>

<https://developer.salesforce.com/docs/atlas.en-us.securityImplGuide.meta/securityImplGuide/login_ip_ranges.htm>

Profile is basically an AD group or LDAP role. Some OOB, some custom

Permission Set is a collection of access rules that extend the profile

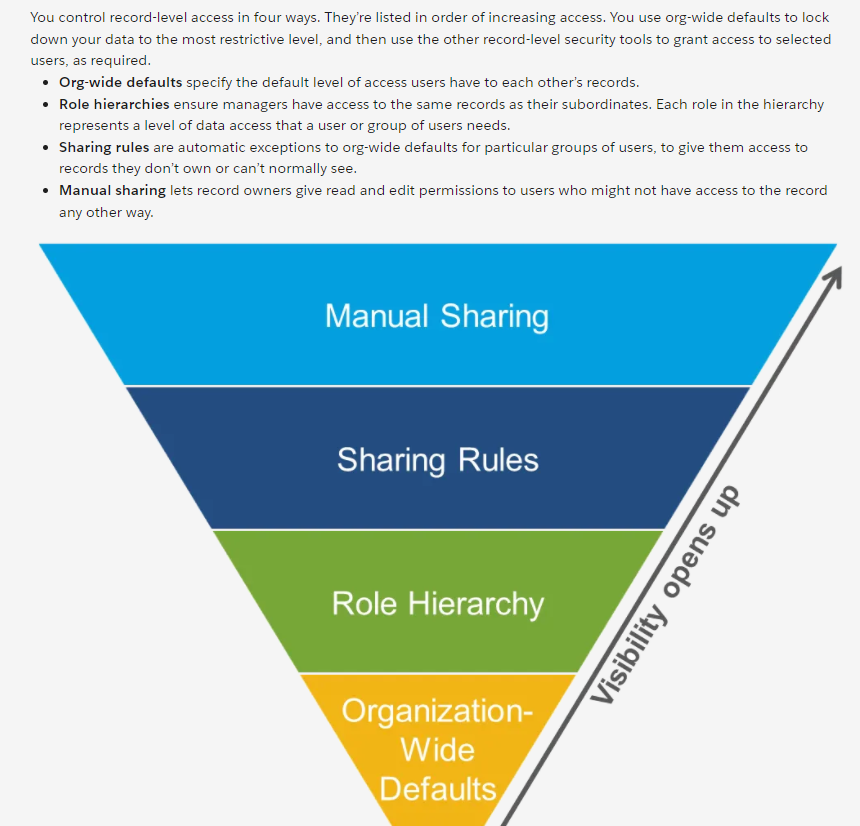
User – Profile

User can have multiple permission sets

Object access controlled through profiles or permissions sets

<https://help.salesforce.com/HTViewHelpDoc?id=users_profiles_about_enhanced_ui.htm&language=en_US>

<https://developer.salesforce.com/docs/atlas.en-us.224.0.securityImplGuide.meta/securityImplGuide/>



The visibility and access for any type of data is determined by the interaction of the above security controls, based on these key principles.

* A user’s baseline permissions on any object are determined by their profile.
* If the user has any permission sets assigned, these also set the baseline permissions in conjunction with the profile.
* Access to records a user does not own are set first by the org-wide defaults.
* If the org-wide defaults are anything less than **Public Read/Write**, you can open access back up for certain roles using the role hierarchy.
* You can use sharing rules to expand access to additional groups of users.
* Each record owner can manually share individual records with other users by using the Share button on the record.

You’ve already seen how to configure object-level and field-level access using profiles and permission sets. Now we’ll look at details of the various record-level security controls.

**Org-Wide Sharing**

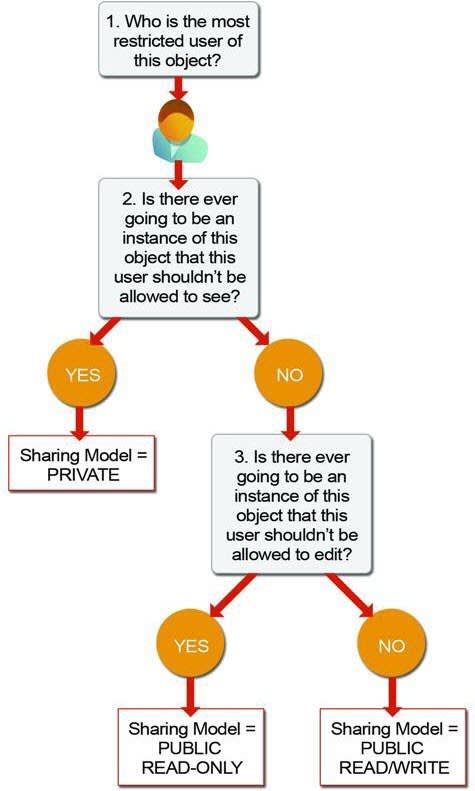
Org-wide defaults specify the baseline level of access that the most restricted user should have. Use org-wide defaults to lock down your data, and then use the other record-level security and sharing tools (role hierarchies, sharing rules, and manual sharing) to open up the data to users who need it.

Object permissions determine the baseline level of access for all the records in an object. Org-wide defaults modify those permissions for records a user doesn't own. Org-wide sharing settings can be set separately for each type of object.

Org-wide defaults can never grant users more access than they have through their object permission.

To determine the org-wide defaults you need for your app, ask yourself these questions about each object:

1. Who is the most restricted user of this object?
2. Is there ever going to be an instance of this object that this user shouldn't be allowed to see?
3. Is there ever going to be an instance of this object that this user shouldn't be allowed to edit?



Based on your answers, you can set the sharing model for that object to one of these settings.

**Private**  
Only the record owner, and users above that role in the hierarchy, can view, edit, and report on those records.  
  
**Public Read Only**

All users can view and report on records, but only the owner, and users above that role in the hierarchy, can edit them.

**Public Read/Write**  
All users can view, edit, and report on all records.

**Controlled by Parent**  
A user can view, edit, or delete a record if she can perform that same action on the record it belongs to.

When the org-wide sharing setting for an object is **Private**or **Public Read Only**, an admin can grant users additional access to records by setting up a role hierarchy or defining sharing rules. Sharing rules can only be used to grant additional access. They cannot be used to restrict access to records beyond what was originally specified with the org-wide sharing defaults.

As an example, let’s go through and answer the above list of questions for the Position object in the Recruiting app.

**Who is the most restricted user of this object?**  
A member of the Standard Employee profile. All that they're allowed to do is view a position.  
  
**Is there ever going to be an instance of this object that this user shouldn't be allowed to see?**  
No. Although the values for the minimum and maximum pay fields are hidden from standard employees, they're still allowed to view all position records.  
  
**Is there ever going to be an instance of this object that this user shouldn't be allowed to edit?**  
Yes. Standard employees aren’t allowed to edit any position record.  
  
Since we answered “Yes” to the third question, the sharing model for the Position object should be set to Public Read Only. By repeating the same exercise with the other recruiting objects, you can easily figure out the appropriate org-wide default settings for them. The Standard Employee profile is the most restricted user for each object, and there are going to be candidate, job application, and review records that particular employees won't be able to view. Consequently, the sharing model for the Candidate, Job Application, and Review objects should all be set to **Private**.

<https://help.salesforce.com/apex/HTViewHelpDoc?id=security_sharing_considerations.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=security_sharing_owd_default_settings.htm&language=en_US>

<https://developer.salesforce.com/docs/atlas.en-us.224.0.securityImplGuide.meta/securityImplGuide/>

<https://help.salesforce.com/HTViewHelpDoc?id=security_apex_sharing_reasons.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=security_controlling_access_using_hierarchies.htm&language=en_US>

sharing rules:

create a public group

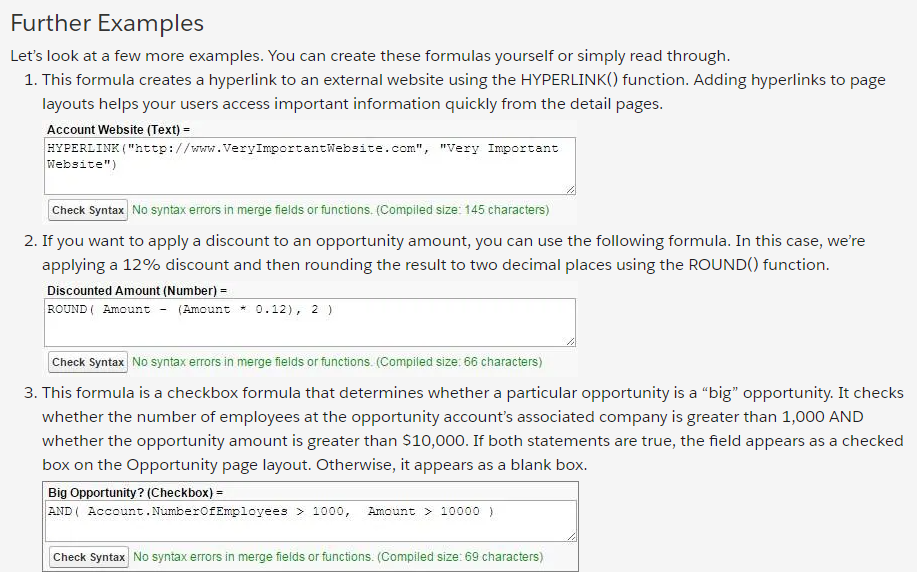
<https://help.salesforce.com/apex/HTViewHelpDoc?id=security_about_sharing_rules.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=security_sharing_data_set_categories.htm&language=en_US>

<https://help.salesforce.com/apex/HTViewHelpDoc?id=security_sharing_considerations.htm&language=en_US>

**Formula Fields**

When adding a field to an object, select Formula as the data type



<https://help.salesforce.com/articleView?id=customize_formuladef.htm&type=5>

<https://resources.docs.salesforce.com/224/latest/en-us/sfdc/pdf/salesforce_formulas_cheatsheet.pdf>

<https://developer.salesforce.com/forums/#!/feedtype=RECENT%26dc=Formulas_Validation_Rules_Discussion%26criteria=ALLQUESTIONS>

Roll up Summary Fields can show aggregate values for child objects

<https://help.salesforce.com/articleView?id=fields_about_roll_up_summary_fields.htm&type=5>

<https://help.salesforce.com/articleView?id=relationships_considerations.htm&type=5>

<https://help.salesforce.com/articleView?id=filter_operators.htm&type=5>

<https://help.salesforce.com/articleView?id=overview_of_custom_object_relationships.htm&type=5>

Validation Rules

<https://help.salesforce.com/articleView?id=fields_about_field_validation.htm&type=5>

<https://help.salesforce.com/articleView?id=fields_managing_field_validation.htm&type=5>

<https://help.salesforce.com/articleView?id=fields_validation_rules_tips.htm&type=5>

**Salesforce Flow**

Flow Builder – make flows

Process Builder – refine existing processes

Apex – build invokable methods

https://help.salesforce.com/HTViewHelpDoc?id=process\_which\_tool.htm&language=en\_US

Process consists of

Trigger

At least 1 criteria node

At least 1 action

Immediate or scheduled

Types:

Record change

Invokable (by another process)

Platform Event

<https://help.salesforce.com/HTViewHelpDoc?id=process_overview.htm&language=en_US>

**Flow Builder**

<https://trailhead.salesforce.com/trails/automate_business_processes/modules/screen_flow_distribution>

<https://help.salesforce.com/articleView?id=flow.htm&language=en_US>

<https://success.salesforce.com/_ui/core/chatter/groups/GroupProfilePage?g=0F9300000001rzc>

<https://trailhead.salesforce.com/modules/screen_flow_distribution>

Create a screen flow > Flows

Create a page > Lightning App Builder

Email template:

Embed properties in body

{!User.Manager}, The {!Opportunity.Name} has been discounted. Please approve this discount. Thank you.

Approval process

<https://help.salesforce.com/HTViewHelpDoc?id=approvals_creating_approval_processes.htm&language=en_US>

<https://help.salesforce.com/apex/HTViewHelpDoc?id=approvals_considerations.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=approvals_checklist.htm&language=en_US>

<https://help.salesforce.com/s/articleView?id=sf.flow_ref_elements_actions_approval.htm&type=5>

<https://trailhead.salesforce.com/projects/build-a-discount-approval-process>

**Mobile**

<https://developer.salesforce.com/signup>

<https://help.salesforce.com/HTViewHelpDoc?id=limits_mobile_sf1_parent.htm&language=en_US>

<https://trailhead.salesforce.com/modules/lex_salesforce1_basics>

<https://trailhead.salesforce.com/trails/salesforce1_mgmt/modules/salesforce1_rollout>

<https://resources.docs.salesforce.com/sfdc/pdf/salesforce1_mobile_security.pdf>

<https://help.salesforce.com/articleView?id=salesforce_app.htm&language=en_US>

<https://trailhead.salesforce.com/trails/salesforce1_mgmt/modules/mobile_strategy>

<https://help.salesforce.com/s/articleView?id=sf.salesforce_app_customize_lex_app.htm&type=5>

<https://help.salesforce.com/articleView?id=user_userdisplay_tabs_lex.htm&language=en_US>

To customize mobile menu:

Search for App Manager

Select the app

Edit

App Options

Verify Desktop and phone are both checked

Navigation Items

Make changes

Open Salesforce on the mobile app to test

Can’t set different menu configurations for different users

Pages must be enabled for mobile

Visualforce pages don’t always work as expected so test carefully

Quick Actions = shortcuts

Global (global actions can’t update any records)

Object-specific

<https://login.salesforce.com/services/walkthrough?path=/p/setup/link/ActionButtonLinkList%3FpageName=Global&type=Global&setupid=GlobalActionLinks&tour=create-global-publisher-action>

Setup > Actions

Quick Actions can have their own layout (simplified or with pre-defined values)

Must be added to global publisher layout (Setup > Publisher)

<https://help.salesforce.com/HTViewHelpDoc?id=actions_overview_global.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=predefined_field_values_notes.htm&language=en_US>

<https://help.salesforce.com/HTViewHelpDoc?id=assigning_global_publisher_layouts_to_profiles.htm&language=en_US>

Compact Layout – appears in the title

<https://help.salesforce.com/articleView?id=compact_layout_overview.htm&language=en_US>

**APEX**

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Salesforce platform.

Setup > Developer Console

An anonymous block is Apex code that does not get stored, but can be compiled and executed on demand right from the Developer Console. This is a great way to test your Apex Classes or run sample code.

**Debug | Open Execute Anonymous Window**

As a language, Apex is:

* Hosted—Apex is saved, compiled, and executed on the server—the Lightning Platform.
* Object oriented—Apex supports classes, interfaces, and inheritance.
* Strongly typed—Apex validates references to objects at compile time.
* Multitenant aware—Because Apex runs in a multitenant platform, it guards closely against runaway code by enforcing limits, which prevent code from monopolizing shared resources.
* Integrated with the database—It is straightforward to access and manipulate records. Apex provides direct access to records and their fields, and provides statements and query languages to manipulate those records.
* Data focused—Apex provides transactional access to the database, allowing you to roll back operations.
* Easy to use—Apex is based on familiar Java idioms.
* Easy to test—Apex provides built-in support for unit test creation, execution, and code coverage. Salesforce ensures that all custom Apex code works as expected by executing all unit tests prior to any platform upgrades.
* Versioned—Custom Apex code can be saved against different versions of the API.

Case Insensitive!

You can get the record ID for a database object from the URL

$A.get("e.force:navigateToURL").setParams(

{"url": "/apex/pageName"}).fire();

**Development Environment Setup**

1. Install Salesforce CLI

**Visual Studio code**

Training Module for VS Code:

<https://trailhead.salesforce.com/en/content/learn/projects/quickstart-vscode-salesforce/start-vscode>

<https://code.visualstudio.com/docs/introvideos/overview>

<https://trailhead.salesforce.com/live/videos/a2r3k000001vD4T/trail-together---quick-start-visual-studio-for-salesforce-development>

<https://developer.salesforce.com/docs/atlas.en-us.sfdx_setup.meta/sfdx_setup>

<https://developer.salesforce.com/tools/sfdxcli>

<https://developer.salesforce.com/tools/vscode/>

<https://developer.salesforce.com/docs/atlas.en-us.236.0.sfdx_dev.meta/sfdx_dev>

<https://developer.salesforce.com/docs/atlas.en-us.236.0.sfdx_cli_reference.meta/sfdx_cli_reference>

<https://oclif.io/>

**Create a Project**

1. Press **Command + Shift + P** on Mac or **Ctrl + Shift + P** on Windows to make the command palette appear.
2. Make sure the new prompt starts with >
3. Type **SFDX: Create Project** and press **Enter** to select the standard template.
4. Type the project name VSCodeQuickstart and press **Enter**.
5. Select your Desktop as the place to create the project in so it is easy to find later on.
6. Wait for the new Visual Studio Code window to open. You should see an indication that the extension is preparing your project before populating the file explorer.

There are a lot of VS commands here. I would need to follow a tutorial to use VS Code

This module does not tell you how to pull code from Salesforce into VS Code

**SOQL VS SOSL**

Use SOQL when you know in which objects or fields the data resides and you want to:

* Retrieve data from a single object or from multiple objects that are related to one another.
* Count the number of records that meet specified criteria.
* Sort results as part of the query.
* Retrieve data from number, date, or checkbox fields.

Use SOSL when you don’t know in which object or field the data resides and you want to:

* Retrieve data for a specific term that you know exists within a field. Because SOSL can tokenize multiple terms within a field and build a search index from this, SOSL searches are faster and can return more relevant results.
* Retrieve multiple objects and fields efficiently, and the objects might or might not be related to one another.
* Retrieve data for a particular division in an organization using the divisions feature, and you want to find it in the most efficient way possible.

Salesforce Federated Search

<https://developer.salesforce.com/docs/atlas.en-us.224.0.soql_sosl.meta/soql_sosl/sforce_api_calls_soql.htm>

<https://developer.salesforce.com/docs/atlas.en-us.224.0.soql_sosl.meta/soql_sosl/sforce_api_calls_sosl.htm>

## Resources

* SOQL and SOSL Reference
  + [Salesforce Object Query Language (SOQL)](https://developer.salesforce.com/docs/atlas.en-us.224.0.soql_sosl.meta/soql_sosl/sforce_api_calls_soql.htm)
  + [Salesforce Object Search Language (SOSL)](https://developer.salesforce.com/docs/atlas.en-us.224.0.soql_sosl.meta/soql_sosl/sforce_api_calls_sosl.htm)
* REST API Developer Guide
  + [Search](https://developer.salesforce.com/docs/atlas.en-us.224.0.api_rest.meta/api_rest/resources_search.htm)
  + [Search Suggested Records](https://developer.salesforce.com/docs/atlas.en-us.224.0.api_rest.meta/api_rest/resources_search_suggest_records.htm)
  + [Search Suggested Article Title Matches](https://developer.salesforce.com/docs/atlas.en-us.224.0.api_rest.meta/api_rest/resources_search_suggest_title_matches.htm)
  + [Search Suggested Queries](https://developer.salesforce.com/docs/atlas.en-us.224.0.api_rest.meta/api_rest/resources_search_suggest_queries.htm)
  + [Parameterized Search](https://developer.salesforce.com/docs/atlas.en-us.224.0.api_rest.meta/api_rest/resources_search_parameterized.htm)
  + [Query](https://developer.salesforce.com/docs/atlas.en-us.224.0.api_rest.meta/api_rest/resources_query.htm)
* SOAP API Developer Guide
  + [search()](https://developer.salesforce.com/docs/atlas.en-us.224.0.api.meta/api/sforce_api_calls_search.htm)
  + [query()](https://developer.salesforce.com/docs/atlas.en-us.224.0.api.meta/api/sforce_api_calls_query.htm)
* Apex Developer Guide
  + [SOQL and SOSL Queries](https://developer.salesforce.com/docs/atlas.en-us.224.0.apexcode.meta/apexcode/langCon_apex_SOQL.htm)
  + [Search Namespace](https://developer.salesforce.com/docs/atlas.en-us.224.0.apexcode.meta/apexcode/apex_namespace_Search.htm)
  + [Search Class](https://developer.salesforce.com/docs/atlas.en-us.224.0.apexcode.meta/apexcode/apex_methods_system_search.htm#apex_System_Search_find)
* Help Documentation
  + [How Does Search Break Up Information?](https://help.salesforce.com/apex/HTViewHelpDoc?id=search_terms.htm&language=en_US)
* Federated Search Developer Guide
  + [Introduction](https://developer.salesforce.com/docs/atlas.en-us.federated_search.meta/federated_search/federated_search_intro.htm)

Search within a single object

FIND {march 2016 email} RETURNING Campaign

Multiple objects

FIND {recycled materials} RETURNING Product2, ContentVersion, FeedItem

Custom object

FIND {pink hi\-top} RETURNING Merchandise\_\_c

<https://developer.salesforce.com/docs/atlas.en-us.224.0.soql_sosl.meta/soql_sosl/sforce_api_calls_soql.htm>

<https://developer.salesforce.com/docs/atlas.en-us.224.0.soql_sosl.meta/soql_sosl/sforce_api_calls_sosl.htm>

only search in specific fields

FIND {jsmith@cloudkicks.com} IN EMAIL FIELDS RETURNING Contact

FIND {Cloud Kicks} RETURNING Account (Name, Industry ORDER BY Name LIMIT 10 OFFSET 25)

FIND {race} RETURNING KnowledgeArticleVersion

(Id, Title WHERE PublishStatus='online' and language='en\_US')

WITH DATA CATEGORY Location\_\_c AT America\_\_c

| **I want to...** | **SOSL** | **SOQL** |
| --- | --- | --- |
| Limit the data that is searched | IN SearchGroup | WHERE |
| Specify the data to be returned in the response | Returning FieldSpec | SELECT |
| Sort results | ORDER BYLIMITOFFSET | ORDER BYLIMITOFFSET |
| Filter by data category | WITH DATA CATEGORY | WITH DATA CATEGORY |

/vXX.X/search/suggestTitleMatches?q=search string&language=article language&publishStatus=article publication status

<https://developer.salesforce.com/docs/atlas.en-us.224.0.api_rest.meta/api_rest/resources_search_suggest_records.htm>

<https://developer.salesforce.com/docs/atlas.en-us.224.0.api_rest.meta/api_rest/resources_search_suggest_title_matches.htm>

<https://developer.salesforce.com/docs/atlas.en-us.224.0.api_rest.meta/api_rest/resources_sobject_suggest_articles.htm>

<https://developer.salesforce.com/docs/atlas.en-us.224.0.api_meta.meta/api_meta/meta_synonymdictionary.htm>

<https://help.salesforce.com/apex/HTViewHelpDoc?id=promoted_search_terms.htm&language=en_US>