Identity and access management.

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

**Understanding Authentication**

Three protocols that Salesforce and other identity vendors follow to implement identity solutions.

* SAML
* OAuth 2.0
* OpenID Connect

**OAuth**

Salesforce uses the OAuth protocol to allow users of applications to securely access data without having to reveal username and password credentials.

Supported OAuth flows include:

* [Web server flow](https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/intro_understanding_web_server_oauth_flow.htm), where the server can securely protect the consumer secret.  
  (When an identity provider is social media e.g. Facebook, Google etc.)
* [User-agent flow](https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/intro_understanding_user_agent_oauth_flow.htm), used by applications that cannot securely store the consumer secret.  
  (Mainly apps that reside on phones or computers.)
* [Username-password flow](https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/intro_understanding_username_password_oauth_flow.htm), where the application has direct access to user credentials.

Choosing an OAuth 2.0 Authentication Flow for Salesforce

As a Salesforce developer, you can use OAuth 2.0 with several authentication flows. When selecting the correct flow for your app, consider the use cases in this table. For details about each flow and its parameters, see references for the flow.

|  |  |  |
| --- | --- | --- |
| **Authentication Flow** | **Description and Use Cases** | **References** |
| Web Server | Apps hosted on a secure server use the web server authentication flow. A critical aspect of the web server flow is that the server must be able to protect the client secret. This flow uses an OAuth 2.0 authorization code grant type. | [OAuth 2.0 Web Server Authentication Flow](https://help.salesforce.com/articleView?id=remoteaccess_oauth_web_server_flow.htm&type=5&language=en_US)  [OAuth 2.0 authorization code grant type](http://tools.ietf.org/html/rfc6749#section-4.1) |
| User-Agent | Users can authorize a desktop or mobile application to access data using an external or embedded browser (or user agent) for authentication. These apps often use a scripting language, such as JavaScript, running within the browser. This flow uses the OAuth 2.0 implicit grant type. | [OAuth 2.0 User-Agent Flow](https://help.salesforce.com/articleView?id=remoteaccess_oauth_user_agent_flow.htm&type=5&language=en_US)  [OAuth 2.0 implicit grant type](http://tools.ietf.org/html/rfc6749#section-4.2) |
| JWT Bearer Token Flow | The main use case of the JWT Bearer Token Flow is server-to-server API integration. This flow uses a certificate to sign the JWT request and doesn’t require explicit user interaction. | [OAuth 2.0 JWT Bearer Token Flow](https://help.salesforce.com/articleView?id=remoteaccess_oauth_jwt_flow.htm&type=5&language=en_US) |
| Device Authentication Flow | Command-line apps or applications that run on devices with limited input and display capabilities, such as TVs, appliances, and other IoT devices, can use this flow. Users can connect these applications to Salesforce by accessing a browser on a device with more advanced input capabilities, such as a desktop or a smartphone. | [OAuth 2.0 Device Authentication Flow](https://help.salesforce.com/articleView?id=remoteaccess_oauth_device_flow.htm&type=5&language=en_US) |
| Asset Token Flow | Client applications use this flow to request an asset token from Salesforce for connected devices. An OAuth access token and an actor token are exchanged for an asset token. This flow combines issuing and registering asset tokens for efficient token exchange and automatic linking of devices to service cloud asset data. | [OAuth 2.0 Asset Token Flow](https://help.salesforce.com/articleView?id=remoteaccess_oauth_asset_token_flow.htm&type=5&language=en_US) |
| SAML Bearer Assertion Flow | An app can reuse an **existing authorization by supplying a signed SAML 2.0 assertion**, as specified in the [SAML 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants](http://tools.ietf.org/html/draft-ietf-oauth-saml2-bearer). A digital signature applied to the SAML assertion authenticates the authorized app. | [OAuth 2.0 SAML Bearer Assertion Flow](https://help.salesforce.com/articleView?id=remoteaccess_oauth_SAML_bearer_flow.htm&type=5&language=en_US) |
| SAML Assertion Flow | This flow is an alternative for orgs that are using SAML to access Salesforce and want to access **the web services API** in the same way. | [SAML Assertion Flow](https://help.salesforce.com/articleView?id=remoteaccess_oauth_web_sso_flow.htm&type=5&language=en_US) |
| Username and Password | Because the username and password flow passes credentials back and forth, avoid using this flow. Use it only for testing, when a user is not present at app startup, or with highly privileged apps. In these cases, set user permissions to minimize access and protect stored credentials from unauthorized access. | [OAuth 2.0 Username-Password Flow](https://help.salesforce.com/articleView?id=remoteaccess_oauth_username_password_flow.htm&type=5&language=en_US) |

Some of the questions – on Sept 27th during Dreamforce 2018 week.

* Post ideas from internal portal to salesforce ideas via API. What settings are required?
* When users click on the links on internal site, How to authenticate users and redirect them to salesforce ideas site?
* Integration between salesforce and google apps.
* How to reflect user changes in salesforce to google apps.
* Rest API
* Salesforce as an identity provider.
* How to use central dB as identity provider for multiple orgs.