codecademy

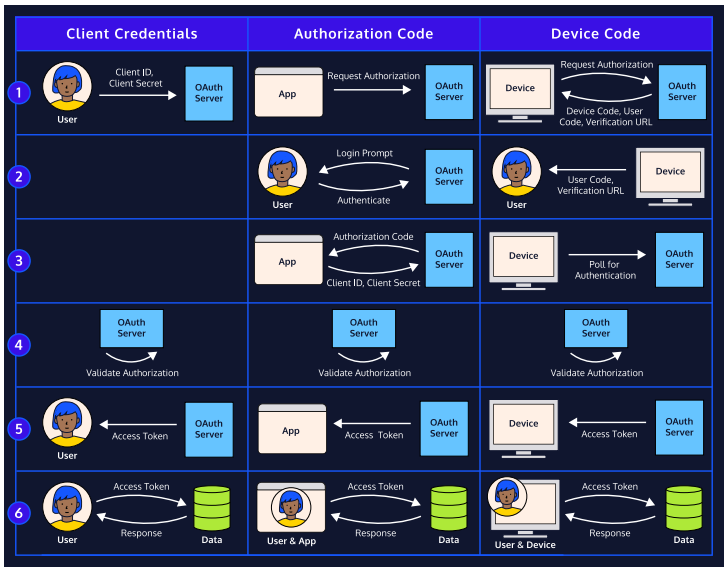
OAuth 2.0

OAuth 2.0

OAuth 2.0 is the current industry standard for authorization. It allows third-parties to access information across websites without needing the credentials for each website.

Grant Types

OAuth 2.0 has different grant types which affect the flow for obtaining an access token. Each grant type is optimized for a specific type of application based on complexity and severity. The grant type chosen will depend on whether the code can securely store a secret key and the trust level between a user and a client.



Access Tokens

Access tokens describe the authorization of an application to access certain aspects or operations of data. They are used as a part of an API request.

Access Token Lifetime

Access tokens have a certain amount of time in which they can be used called an *Access Token Lifetime*. This time is meant to be kept short.

codecademy

The `oauth-server` Module

The `oauth2-server` module is one of many modules that provide OAuth 2.0 authorization for Node.js applications.

The `oauth2-server` Model Object

An `oauth2-server` instance needs a model object which contains functions to retrieve, store, and validate our access tokens.

OAuth Roles

OAuth 2.0 defines four roles:

- Client
- Authorization server
- Resource server
- Resource owner.

Public Client

Public clients cannot securely store credentials. They can only use grant types that don't require the Client Secret.

Confidential Clients

Confidential clients are applications that can be secured without being exposed to a third-party application/server. It can be registered to an authorization server using a Client ID and a Client Secret as credentials.

The `getClient()` Function

Authorization flows require using the `getClient()` function to retrieve a client by the client's ID and/or Secret.

codecademy

The `saveToken()` Function

Authorization flows require using the `saveToken()` function to store the access token as a database object.

The `getUserFromClient()` Function

The `getUserFromClient()` function retrieves the user associated with the specified client. This function must be implemented to use the Client Credentials grant type.

The `getAccessToken()` Function

The `getAccessToken()` function retrieves tokens that were saved by the `saveToken()` function.

The `authenticate()` Method

The `authenticate()` method returns a `Promise` that resolves to an access token object. The token is retrieved via the `getAccessToken()` method of the provided model.

Client ID

A *Client ID* is a public identifier for apps that is unique across all clients and the authorization server.

Client Secret

A *Client Secret* is a secret key known only to the application and an authorization server.

Save Print Share ▾