Payroll

Pricebook

SalesTech

Scheduling Pro

Service Agreements

Settings

Task Management

Telecom

OAuth 2.0

OAuth 2.0 is an authentication framework as defined by the RFC-6749 standard. OAuth 2.0 focuses on client developer simplicity while providing specific authentication flows for web and desktop applications. Generally, OAuth provides clients with secure delegated access to server resources on behalf of a resource owner. It allows resource owners to authenticate and authorize third-party access to their server resources without the need to share personal credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth uses short-lived access tokens issued to a client by an authentication server with the approval of the resource owner. The client then uses the access token to access the protected resources hosted by the resource server.

Because the ServiceTitan API supports the OAuth 2.0 protocol, your application does not need to store or transmit user account names or passwords but instead relies on application credentials in the form of a Client ID and Secret key unique to your application.

Note: Currently ServiceTitan supports only Client Credentials Grant Type and its related use cases only.

OAuth 2.0 Roles

OAuth 2.0 flow has the following roles:

Role	**Description**
Resource Owner (ST Customer)	Entity that can grant access to a protected resource. Typically, this is the admin managing the ServiceTitan app within the customer's organization.
Resource Server (ServiceTitan API)	Server hosting the protected resources. This is the API you want to access.

Client (your application)	Application requesting access to a protected resource on behalf of the Resource Owner. This is your application.
Authorization server (ServiceTitan)	Server that authenticates the Resource Owner and issues access tokens after getting proper authorization. In this case, the ServiceTitan authorization server.

ServiceTitan OAuth 2.0 access tokens

Applications use access tokens to make API requests according to the OAuth 2.0 specification.

In the context of the ServiceTitan API, an access token represents the authorization for a specific application to access a user's data in ServiceTitan. Access tokens must remain confidential in both transit and storage. As a result, access tokens are only used over HTTPS (SSL) connections, since passing them over unencrypted channels renders them susceptible to interception and compromised security.

Currently, the ServiceTitan API employs only access tokens and does not issue refresh tokens as we are supporting only machine-to-machine calls under OAuth 2.0 client credentials grant type. By design, the ServiceTitan authorization server issues access tokens which expire in 900 seconds. Once the initial access token expires, your machine will need to call the ServiceTitan endpoint again using the ClientID and Client Secret to obtain a new access token.

Tip: We recommend you cache the access token instead of requesting a new token for every transaction, since ServiceTitan employs throttling of API requests to new access tokens.

Note: You should use an environment (integration versus production)-specific ClientID and Secret key when calling the endpoints to request the access token.

Token Type	**Expiration**	**Token request needs**	**Token endpoint**

Access token	900 seconds	Client ID & Secret	Integration environment: *https://auth-integration.servicetitan.io/connect/token* Production environment: *https://auth.servicetitan.io/connect/token*

OAuth 2.0 grant types

While we are working quickly to support all OAuth 2.0 grant types, we are currently only supporting Client Credentials Grant (machine-to-machine, “user less access”).

Using the OAuth 2.0 Client Credentials grant type

Currently, ServiceTitan API supports only the client credentials OAuth 2.0 flow. The client credentials flow is not associated with a specific ServiceTitan user (resource owner). In addition, it’s not necessary to first obtain an authorization code before retrieving an access token when using this client credentials grant type.

The client credentials grant type provides a specific grant flow in which the resource owner (the user) is not involved. When using this grant with ServiceTitan, the client application requests an access token using only its own credentials and uses the access token on behalf of the client application itself. This grant flow is best suited for API methods that are used by the client application in general, instead of methods that apply to a certain resource owner. For example, API methods for reporting, analytics, administrative tasks, and system maintenance. This method for using an API is also referred to as user less access.”

Common use cases for this grant type

There are a number of scenarios in which using the client credentials grant flow in the ServiceTitan API is the preferred approach.

- Server-to-server integrations where a specific user’s permission to access their data is not required (for example, non-delegated authorization)
- Report generators, data mining, or other integrations that access company-wide data
- Backend scripts, system maintenance and administration utilities