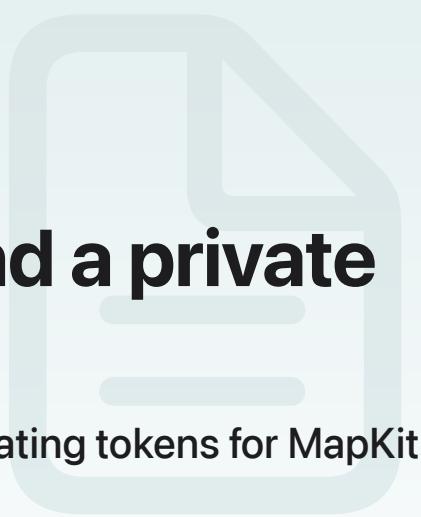


[Apple Maps Server API](#) / Creating a Maps identifier and a private key

Article

Creating a Maps identifier and a private key

Create a Maps identifier and a private key before generating tokens for MapKit JS.



Overview

MapKit JS uses a token to authenticate map initializations and other API requests. Before you can create a token, you need a Maps identifier (Maps ID) and private key that's associated with a Maps ID.

You create a Maps ID and private keys through your [Apple Developer Account](#). After you complete the steps to obtain these items you can construct a token and sign it with your private key, as [Creating and using tokens with Maps Server API](#) describes.

Create a Maps ID

A Maps ID is a string that you provide to identify a domain or environment that calls the MapKit JS API. Maps IDs use reverse-domain-style notation with three or four fields separated by a dot. The first field must be maps; the remainder of this string can be a name that's meaningful to you. For example, your string might resemble maps.com.mywebsite or maps.com.mycompany.mywebsite.

To create a Maps ID, follow these steps:

1. Go to [developer.apple.com/account](#) and log in with your Apple Developer credentials.
2. Under [Certificates, Identifiers & Profiles](#), click Identifiers in the sidebar.
3. At the top of the identifiers list, click the Add Identifiers button (+).
4. On the following page, select the Maps IDs checkbox, and then click the Continue button at the top of the page.

5. Enter a string for the description. This can be your app name, team name, project name, or anything that conveys context and is meaningful to you.
6. Enter a reverse-domain-style string for the identifier (for example, maps.com.mycompany.mywebsite), then click Continue.
7. Review the information, then click Register.

If you have multiple environments, such as a development environment and a production environment, it's good practice to create separate Maps IDs and keys for each environment. That way, if a key you use in production becomes compromised, revoking it doesn't affect your UAT or test environments.

Obtain a MapKit JS private key

After you create a Maps ID, the next steps are to create a private key, add the MapKit JS service to this key, and associate it with a Maps ID:

1. In Certificates, Identifiers & Profiles, click Keys in the sidebar, then click the Keys + button at the top of the keys list.
2. Under Key Name, enter a unique name for the key.
3. Below that, select the checkbox next to MapKit JS. Note that the MapKit JS checkbox isn't in an enabled state until you create a Maps ID.
4. Near the top right of the page, click Configure. On the next page, choose a Maps ID to associate to this key from the pop-up menu, then click Save.
5. Click Continue, review the key configuration, then click Register.
6. Optionally, click Download to download the key. The private key is available to download a single time. If the Download button isn't in an enabled state, you previously downloaded the key associated with this identifier.
7. Click Done.

After executing these steps, you have a private key that you can use to sign tokens, and you can begin Creating and using tokens with Maps Server API.

Important

The private key is a shared secret between you and Apple. To keep your private key secure, don't put your private key in public facing client-side code, and don't share the key.

The private key doesn't expire, but you can revoke it. Revoking a key makes it invalid and affects calls to the MapKit JS API. If you lose a key or if someone else starts using your key, revoke it. See [Revoke, edit, and download keys](#) for more information.

See Also

Essentials

-  Creating and using tokens with Maps Server API

Sign JSON Web Tokens to use Maps Server API and debug common signing errors.

Generate a Maps token

Returns a JWT maps access token that you use to call the service API.

-  Debugging an Invalid token

Inspect the JavaScript console logs, the token, and events to determine why a token is invalid.

-  Common objects

Understand the common JSON objects that API responses contain.

-  Integrating the Apple Maps Server API into Java server applications

Streamline your app's API by moving georelated searches from inside your app to your server.