Article

Debugging an Invalid token

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

Overview

MapKit JS requires a valid token to initialize. If the token you provide isn't valid, MapKit JS won't initialize and doesn't provide a valid map to the user. When MapKit JS fails to initialize, errors are available in the JavaScript console.

Identify errors in the JavaScript console

To diagnose this situation, first check your browser's JavaScript console for messages that describe why your token may be invalid. Reasons why a token might be invalid include:

- The token is missing required fields.
- The token has all the required fields, but some values are incorrect. Verify that the `kid` claim matches the key identifier used to sign the token, and that the `iss` claim matches the 10-character Team ID for your Apple Developer Account.
- The string provided within the `origin` claim in your token doesn't exactly match the `Origin` header passed by the browser. This string must match exactly, include the protocol (`http`, `https`), trailing slashes, and so on.

For more information about the structure of MapKit-specific token, see Creating and using tokens with Maps Server API.

Examine the token

MapKit JS considers a token with duplicate headers to be invalid. You can view headers of the token and check for duplicates by decoding the token at the command line. For example, the following command decodes the headers of a token:

```
% echo "your-token-here" | tr "." "\n" | head -2 | base64 -D

// Prints
// {"typ":"JWT","alg":"ES256","kid":"KEY_ID","kid":"KEY_ID"}
```

In the above output, you can see that the token has two `kid` headers.

If you find duplicate headers in your token, review the software that you used to create the key to remove any duplicate headers.

> Note
>
> A token contains additional encrypted fields that aren’t decoded and made visible with the command above.

For more information on how to create a new token, see Creating and using tokens with Maps Server API.

Inspect the error event on MapKit JS

The MapKit JS API rejects an otherwise valid token when your Apple Developer team exceeds the number of requests allowed. You can detect when your team exceeds the number of allowed requests by listening to the `error` event on the `mapkit` object.

The following example adds an event listener to the `mapkit` object and inspects the `status` string to see if the emitted error lists too many requests.

```
mapkit.init({ authorizationCallback: function(done) { done("your-token"); }, ... });
mapkit.addEventListener("error", function(event) {
    switch (event.status) {
    case "Too Many Requests":
        // Exceeded number of requests allowed.
        break;
    }
});
```

For more information about MapKit JS events, see Handling initialization events.

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.

Creating a Maps identifier and a private key

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

`Generate a Maps token`

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

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.