Using the EDGE API

The Estimating Edge provides an API so you can access The EDGE™ database from your applications. The EDGE API is a REST API, which communicates securely over HTTPS using a JSON/xml format.

Like most web-based APIs, The EDGE API uses the standard HTTP verbs:

- **get**: Requests data from The EDGE database. This is the most common type of request.
- **post**: Adds new data to The EDGE database. Using this type of request, you can, for example, add a new contact to CRM.
- **put**: Changes existing information. For example, using this type of request, it would be possible to change the color or value of an existing product.
- **delete**: Deletes existing information.

Authentication

Before you can use The EDGE API, you must request access from Estimating Edge. When your access is approved, you will get:

- The URL to access the API.
- An API Key. This uniquely identifies your database from that of other users.
- Client Key(s). You will get one Client Key for each client application you use. For example, if you are using a CRM application and an accounting application, each one will get its own unique Client Key.

To prevent unauthorized use of The EDGE API, you must authenticate with the API. The EDGE API uses the OAuth2 protocol. Your client application sends the API Key and the Client Key and The EDGE API returns an access token and a refresh token. Your client application then uses the access token in its API calls to prove it is authorized. The default life of an access token is 8 hours, but it can be extended using the refresh token.

The following illustration shows the Authentication process:
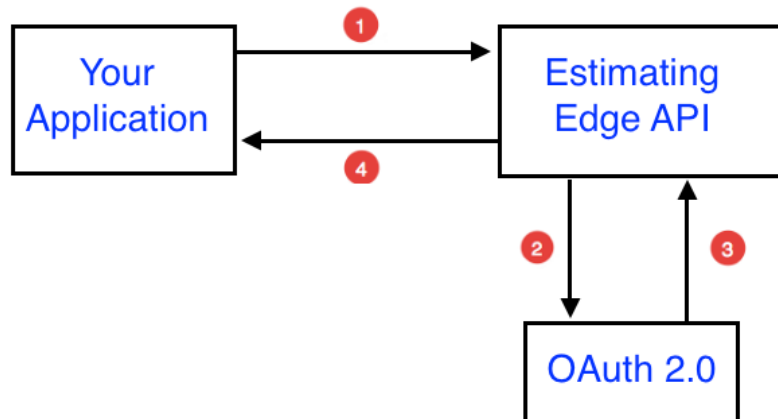
1. Your application sends your API Key and Client Key to the URL provided by Estimating Edge.
2. The EDGE API forwards the keys to the OAuth authenticator.
3. The OAuth authenticator generates an access token and a refresh token and sends them to The EDGE API.
4. The EDGE API sends the tokens to your application.

Making the Authentication Call

To make the authentication call to The EDGE API URL, your application must first send a post request that includes the API Key and Client Key. It looks like this:

```
{
  "post": "/token",
  "apikey": "bearer <API Key as supplied by Estimating Edge>",
  "grant_type": "password",
  "api_client": "<Client Key as supplied by Estimating Edge>"
}
```

The response from The EDGE API server is

```
{
  "access_token": "<256 byte hexadecimal token>",
  "token_type": "bearer",
  "expires_in": 28799,
  "refresh_token": "<256 byte hexadecimal token>",
```
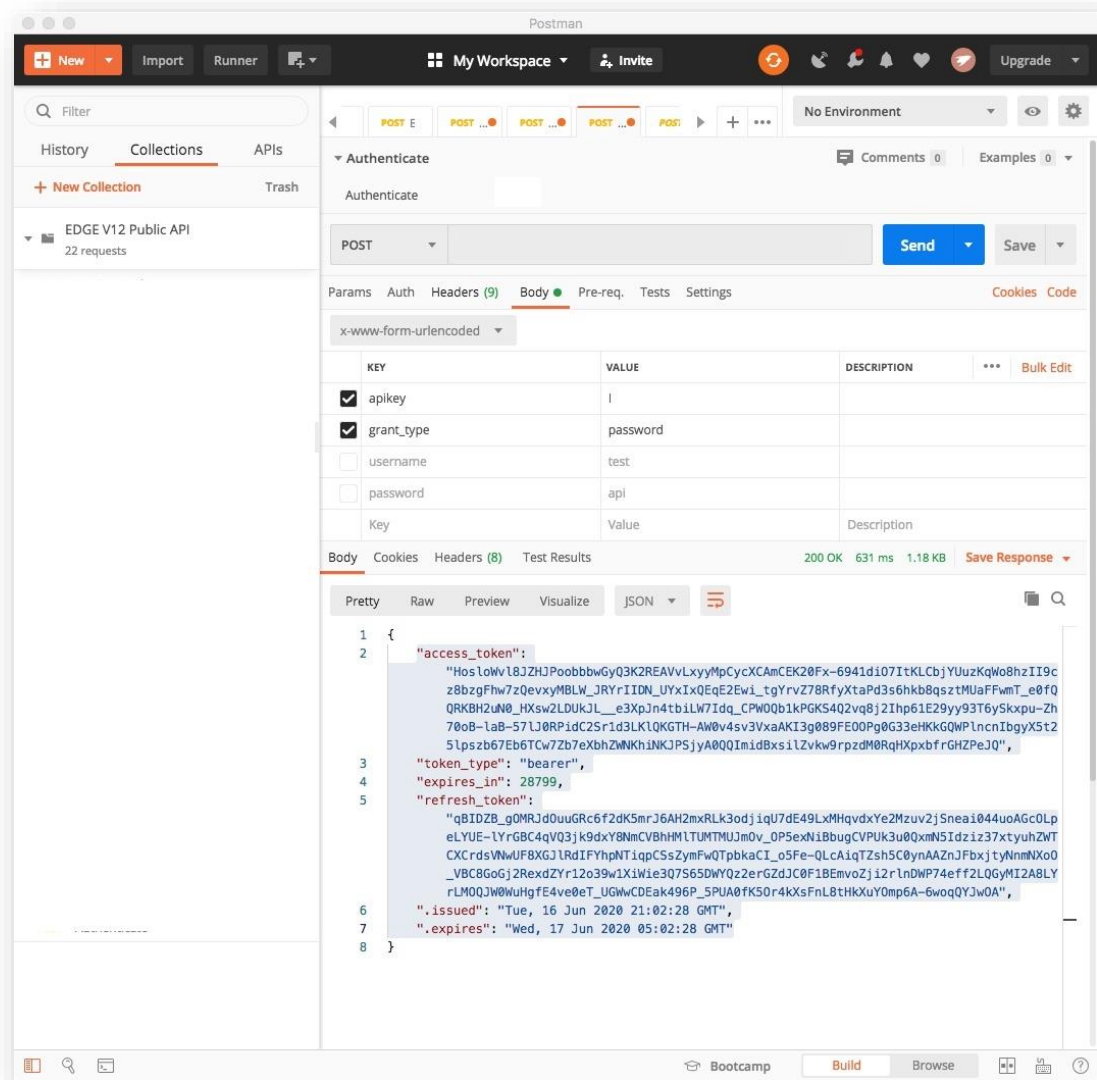
```
       ".issued": "Thu, 22 May 2020 15:16:21 GMT",
       ".expires": "Thu, 22 May 2020 23:16:21 GMT"
   }
```

Your application should capture the access token, the refresh token, and the expiration timestamp. Your application will use the access token to authorize all future API calls.

The following illustration shows the use of the popular Postman® application to place the authentication call:

You can see the access_token and refresh_token returned by the call.

Refreshing the Access Token

If your application is still using The EDGE API close to the expiration of the access token, it can refresh its authorization by sending the refresh token. That will grant a new access token. The call is:

```
{
  "post": "/token",
  "grant_type": "refresh_token",
  "refresh_token": "<Refresh Token>"
}
```

Each access token has a life of 8 hours.

Making Your First Request

The following web page has the complete list of available endpoints and their call syntax and response syntax:

https://api.edgeestimator.com/12.8/swagger/ui/index

The syntaxes can change over time, so make sure to check the contents of this web page to ensure you are always using the most updated information.

Here is an example of a call that you can use to model your calls. This is using the /account/info endpoint, which returns account information.

Request:
```
{
  "get", "/account/info",
  "Authorization": "bearer <256-byte hexadecimal access token"
}
```

The following illustration shows the use of the popular Postman application to place the account/info call:
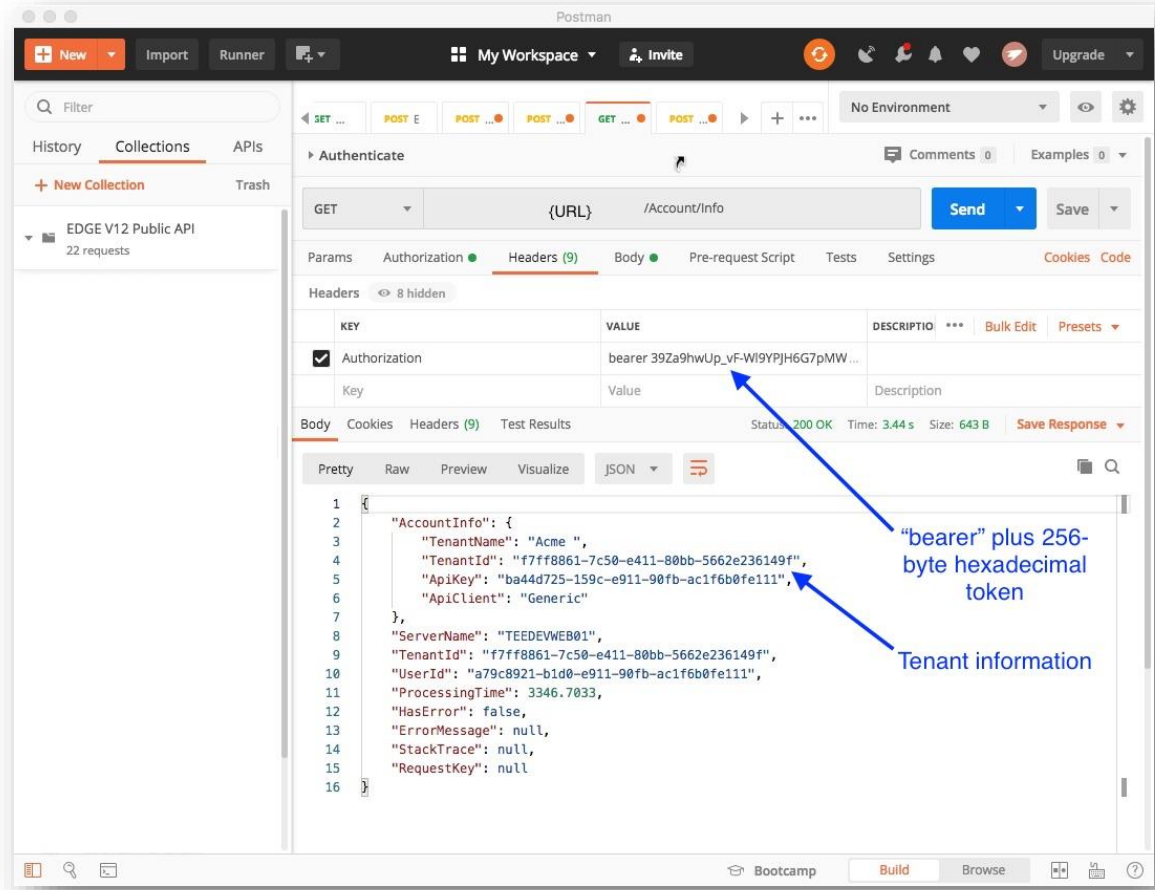
When you place the account/info call, the API returns the tenant information.