

# JWT API Authentication

## Getting Started

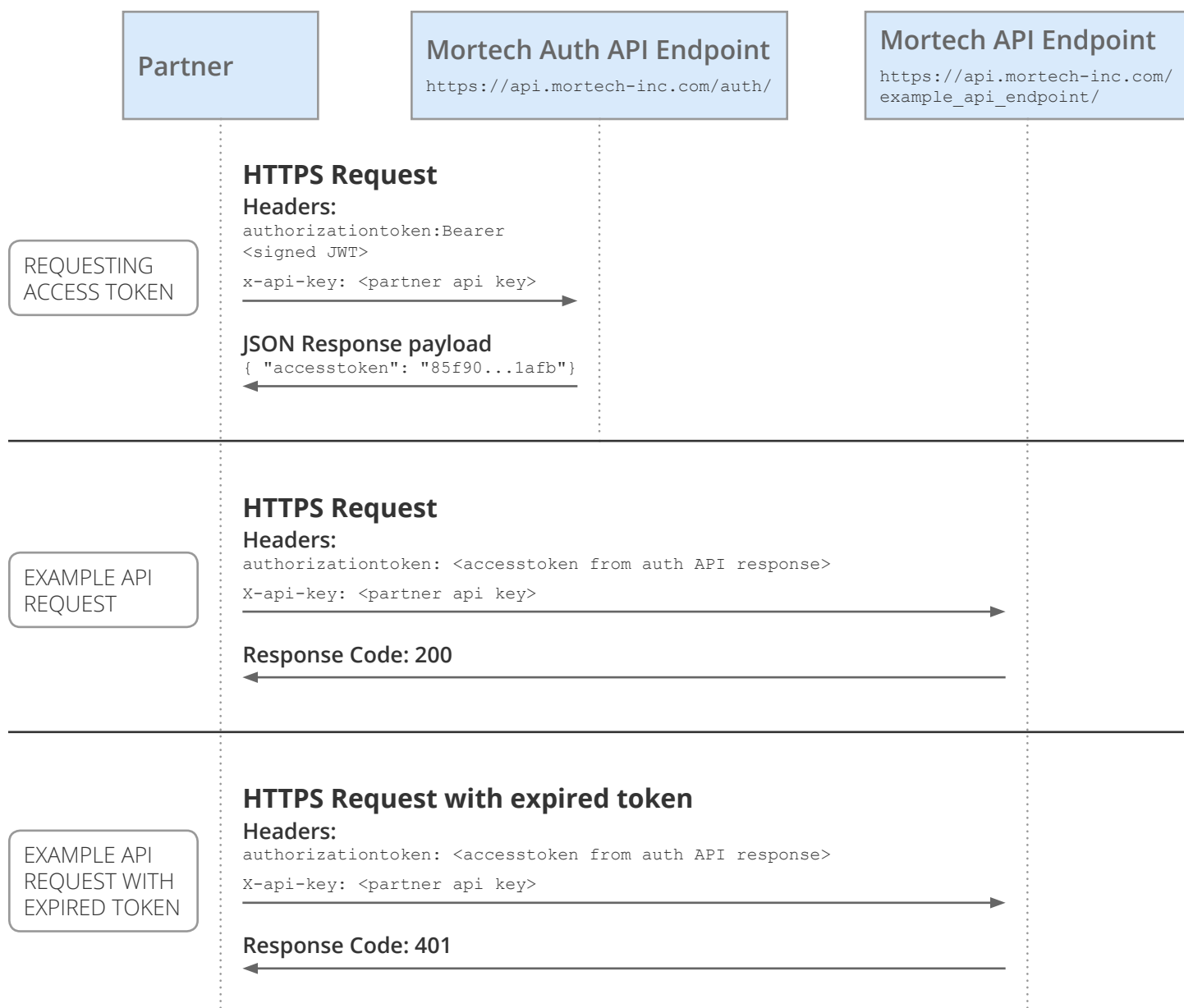
In order to get access to Mortech APIs you will need to get a Partner ID, API key, and a private key. Contact your Customer Success Manager to get setup started.

1. Mortech will generate a Partner account within our system and needs the email address of whomever will be receiving the Partner ID, API key, and private key
2. The Partner will receive an email with a link where they can receive their Partner ID, API key, and private key .pem file.
3. Follow the link provided in the email to download the private key file. This is needed to sign the JWT authorization token.
  - a. Mortech provides a Private Key during the provisioning process. Mortech keeps the Public Key equivalent for validating the signatures. The Private Key is generated from the Client's browser and is never transmitted on the internet. Therefore Mortech never has a copy of the Private Key. It is the responsibility of the Partner to keep the Private Key secure.

## JWT Authentication Process

Mortech uses JSON Web Token (JWT) authentication for their API services. For more information about JWT, please visit <https://jwt.io/introduction/>. The authentication API generates a temporary token which can be used to access Mortech's API services. The token will expire after a certain period of time. After that time, a new token will need to be generated to continue to access Mortech's API services.

The best way to know if you need a new token is to make an API request and check the response code. 401 means the `authorizationtoken` supplied is either invalid or expired. In this case the client should get a new token from the authenticate API and retry their request.



1. The Partner generates a signed JWT.
2. The Partner calls Mortech's Auth API with the JWT as a Bearer token of the `authorizationtoken` Header.
3. The Auth API responds with an `accesstoken`.
4. The Partner can then make multiple requests to various Mortech API endpoints using the token for a period of time until the token expires.
5. When the `accesstoken` expires the Partner will get a 401 response code with an "Unauthorized" message at which time they need to create a new JWT and request a new access token.

## Generating a Signed JWT

Please visit <https://jwt.io> which provides additional information and links to several open source libraries for using JWTs.

### Header of the JWT

The header consists of two parts:

- ``alg`` which is the hashing algorithm being used
- ``typ`` which is JWT

Mortech requires these two items to be in the header. They are not optional and they are fixed values. Do not use an ``alg`` or ``typ`` different from what is below.

#### Header JSON

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

### Payload of the JWT

The payload consists of three parts:

- ``partnerId`` which is issued during the provisioning process. (See Getting Started)
- ``customerId`` which is the ID of the customer the Partner is making requests on behalf of.
- ``iat`` which is the 'Issued-At Time' timestamp at which the JWT is created. This value must be within 5 minutes of the system time of Mortech's APIs to be considered valid.

#### Payload JSON

```
{
  "partnerId": "350",
  "customerId": "30bank01",
  "iat": 1495634289
}
```

## Signature of the JWT

The signature portion of the JWT is the signed combination of the encoded Header and Payload sections. To sign the JWT, Mortech provides a Private Key during the provisioning process (See Getting Started). Please visit <https://jwt.io/introduction/> for more info on signing your JWT

## Resulting JWT

The **header** and **payload** portions of the JWT are base64 encoded and the **signature** is encrypted.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXJ0bmVySWQiOiI0MjIiLCJjdX0b21lcklkIjoimzBqdXN0aW4wMSIsIm1hdCI6MTQ5NTYzNDI4OX0.MTvlYoxMha8EmJHSHCIagga6lFAnUdXj6qZR-qZqwaQ
```

## Example Auth Request

### API Key

All requests to Mortech API's require an API Key which is provided during the provisioning process (See Getting Started). It should be passed in the header as `x-api-key` (See the example below).

### Authorization Token

All requests to Mortech APIs require an `authorizationtoken` Header. The `authorizationtoken` header for the auth API will be a JWT generated and signed using the partners private key. The value for the `authorizationtoken` header will be a bearer token with a space between the word `Bearer` and the signed JWT.

### Authorization URL

```
https://api.mortech-inc.com/auth
```

## Request Method: GET

### Headers

```
authorizationtoken: "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXJ0bmVySWQiOiI0MjIiLCJjdXN0b21lcmlkIjoimZBqdXN0aW4wMSIsIm1hdCI6MTQ5NTYzNDI4OX0.MTvlYoxMha8EmJHSHCIaagga6lFAnUdXj6qZR-qZqwaQ"
x-api-key: "OPLcE7uALa7...6eXaKS6ZhK4"
```

### Result (200)

```
{
  "accesstoken": "85f900a7dce3...af9f59a8c1afb"
}
```

## Auth API Response Codes

Successful response codes	Meaning
200	The request was successful and the JSON response body will contain an `accesstoken`.
Unsuccessful response codes	Meaning
400	There was an issue with the request. Usually a bad input parameter.
401	The supplied `authorizationtoken` is invalid.
403	The supplied `authorizationtoken` is expired. The Partner should sign a new JWT and request a new token from the /auth API.
500	The service is experiencing problems. This error should be reported to Mortech.