Authentication and Authorization

Authentication & Authorization with OAuth 2.0

All of Manheim's modern APIs support OAuth 2.0, an authorization framework for providing limited access to data on a web server. Accessing an API with OAuth 2.0 involves obtaining a temporary access token and using this token to request a limited set of resources.

This document describes how to obtain an OAuth 2.0 access token with each of the supported OAuth 2.0 grant types, how to use the token in an API request, and how to check the status of a token. Because applications must be approved before accessing Manheim APIs, a brief discussion of authentication is also provided.

Note: If your app is client-facing and has a user with login credentials, contact us at eventer-team@coxautoinc.com to discuss your usecase.

Authentication Credentials

Before a client application can obtain an OAuth2 access token:

1. Manheim must review and evaluate the organization and application.
2. If the review is successful, the client application must be registered in **Mashery** for Manheim API access.
3. **Mashery** will provide a `package key` and `secret` which can be used as client credentials for obtaining an OAuth2 access token.

See API Access and Environments to learn more about getting access to Manheim API environments.

Note: If you use a Company ID today to request tokens, you need to contact us at eventer-team@coxautoinc.com to gain access to Manheim APIs.

Quotas and Caching

Manheim configures quotas for each API client to constrain the maximum number of requests that can be made per second and per day. This protects the APIs from runaway clients. It is the client's responsibility to cache each OAuth2 access token and request a new token upon token expiration instead of using a timer. This is a critical aspect of a well-behaved API client.

OAuth Grant Types

Manheim APIs support the *Client Credentials* and the *Resource Owner Password Credentials* OAuth2 grant types. For more information, refer to RFC 6749 - The OAuth 2.0 Authorization Framework.

Note that Maheim API customers will use their Mashery `package key` for the OAuth 2.0 Client ID, and the `secret` associated with the package key as the OAuth 2.0 secret code.

Request an Access Token with the Client Credentials Grant Type

With the Client_Credentials grant type, an application will use a Mashery package key and secret in each HTTP POST request to obtain an access token. No end user is involved.

Endpoints

Production

```
POST https://api.manheim.com/oauth2/token.oauth2
```

Pre-Production

```
POST https://uat.api.manheim.com/oauth2/token.oauth2
```

Note: Any API clients using Company ID today MUST continue to use the existing token endpoint (https://api.manheim.com/oauth2/token) for all requests.

Request

The request must be an HTTPS POST to one of the Manheim OAuth2 endpoints, as shown in this example:

```
curl -X POST \
https://api.manheim.com/oauth2/token.oauth2 \
-H 'authorization: Basic aDNod2dtNDQ0NWtqenB0XXXXXXXXXXXXX=' \
-H 'content-type: application/x-www-form-urlencoded' \
-d 'grant_type=client_credentials&scope=APP1%3AABC%20APP@%3ACDE'
```

A valid request contains the following headers and parameters (in snake case):

- *authorization* header consisting of 'Basic' followed by a base64-encoded string containing the `Mashery package key` and `secret`. See the "Preparing the Authorization Header" section for more information.
- *content-type* header with the value 'application/x-www-form-urlencoded'
- *grant_type* parameter, with the value 'client_credentials'
- *scope* parameter, with values for the requested scopes, if the API requested has scopes defined

Preparing the Authorization Header

Each token request requires an *authorization* header containing the `Mashery package key` and `secret`, but those credentials must be encoded before they can be used. The *authorization* header consists of the word 'Basic' followed by a base64-encoded string containing the package key and secret:

For example, if your application's package key is 'zq4hmfg72z3zabc4wr72euyu', and the secret is 'A2Qxe4z83X',

1. Put the package key and secret in a string separated by a colon, for example 'zq4hmfg72z3zabc4wr72euyu:A2Qxe4z83X'
2. Use a command-line tool or your language of choice to base64-encode the string
3. Insert the result after the 'Basic' keyword

About Scopes

If an API defines scopes for accessing specific types of data, the client must include a "scope" parameter in the access token request. Manheim determines which scope(s) are appropriate for each API client.

- A client can receive an access token without scopes if the client does not pass a "scope" parameter in the token request. However, an error will occur if the client attempts to use this token to access an API endpoint that requires a specific scope.
- In accordance with OAuth 2.0, the authorization server does not attach scopes to the token unless the scopes are passed by the client as part of the access token request. Thus its imperative that client passes scopes if the requirement is to access an API endpoint that requires a scope.
- The Client should pass only those scopes to which the client is entitled. Requesting a token with an unapproved or incorrect scope will result in an error with no token generation.

Response

If the request is successful, an access token will be returned as shown in the following example:

```
HTTP/1.1 200 OK
  Content-Type: application/json;charset=UTF-8
  {
    "access_token":"sz2vxvunynsu6f499y2qrgst",
    "token_type":"bearer"
  }
```

Request an Access Token with the Resource Owner Password Credentials Grant Type

Use this grant type when resource owner authentication or resource owner data retrieval is required. The application provides its own login form. As with the Client Credentials grant type, the application will also need to provide a `Mashery package key` and `secret` to obtain an access token.

Endpoints

Production

```
POST https://api.manheim.com/oauth2/token.oauth2
```

Pre-Production

```
POST https://uat.api.manheim.com/oauth2/token.oauth2
```

Note: Any API clients using Company ID today MUST continue to use the existing token endpoint (https://api.manheim.com/oauth2/token) for all requests.

Request

The request must be an HTTP POST to one of the Manheim OAuth2 endpoints, as shown in this example:

```
curl -X POST \
https://api.manheim.com/oauth2/token.oauth2 \
-H 'authorization: Basic aDNod2dtNDQ0NWtqenB0XXXXXXXXXXXXX=' \
-H 'content-type: application/x-www-form-urlencoded' \
-d 'grant_type=password&username=johndoe&password=abcde&scope=APP1%3AABC%20APP@%3ACDE'
```

A valid request contains the following headers and parameters (in snake case):

- *authorization* header consisting of 'Basic' followed by a base64-encoded string containing the `Mashery package key` and `secret`. See the "Preparing the Authorization Header" section above for more information.
- *grant_type* parameter, with the value 'password'
- *scope* parameter, with values for the requested scopes, if the API requested has scopes defined
- *username* parameter, which is the resource owner's user name
- *password* parameter, which is the resource owner's password

About Scopes

If an API defines scopes for accessing specific types of data, the client must include a "scope" parameter in the access token request. Manheim determines which scope(s) are appropriate for each API client.

- A client can receive an access token without scopes if the client does not pass a "scope" parameter in the token request. However, an error will occur if the client attempts to use this token to access an API endpoint that requires a specific scope.
- In accordance with OAuth 2.0, the authorization server does not attach scopes to the token unless the scopes are passed by the client as part of the access token request. Thus its imperative that client passes scopes if the requirement is to access an API endpoint that requires a scope.
- The Client should pass only those scopes to which the client is entitled. Requesting a token with an unapproved or incorrect scope will result in an error with no token generation.

Response

If the request is successful, an access token will be returned as shown in the following example:

```
{
< HTTP/1.1 200 OK
< Content-Type: application/json;charset=UTF-8
< {
<   "access_token":"Yn2ToFZFEjr1zIsCcMWB54",
<   "token_type":"bearer"
}
```

Use an Access Token in an API Request

With the *access_token* value provided by the OAuth2 token endpoint, you can now make API calls against other Manheim endpoints. The token should be supplied as part of the Authorization header in each API request, prefixed by the token type of "**Bearer**."

For example, after receiving the JSON object above, the API consumer's call to an OAuth2 protected resource would include the token in the Authorization header as shown here:

```
curl -X GET \
  https://api.manheim.com/locations \
  -H 'Authorization: Bearer Yn2ToFZFEjr1zIsCcMWB54'
```

Note: In the Authorization header, the keyword "**Bearer**" is case-sensitive. As shown in the example above, the keyword's first letter **must** be capitalized and the remaining letters **must** be in lowercase.

Identify the Expiration of a Temporary Token

After a temporary OAuth2 token has expired, API calls will return with 401 UNAUTHORIZED errors. Each response will also include a “WWW-Authenticate” header indicating that the token has expired.

```
WWW-Authenticate: Bearer realm="api.manheim.com", error="invalid_token"
```

After encountering this error, you should obtain a new token and retry the failed API request.

Check the Status of an Access Token

You can proactively check an access token to determine if it is active or has expired, and to see the scope(s) assigned to the token. To do so, send the token to Manheim's token status endpoint.

Endpoints

Production

```
GET https://api.manheim.com/oauth2/token/status
```

Pre-Production

```
GET https://uat.api.manheim.com/oauth2/token/status
```

Status Request Examples

Production

```
curl -X GET \
  https://api.manheim.com/oauth2/token/status \
  -H 'Authorization: Bearer sz2vxnuvynsu6f854y2qrgst'
```

Pre-Production

```
curl -X GET \
  https://uat.api.manheim.com/oauth2/token/status \
  -H 'Authorization: Bearer K5d7eg6t4mfhya9wdcbpvsc44'
```

Response Examples

If the access token is active, the API returns information about the token, including any scope(s) included with token.

```
{
    "mashery_headers": {
        "HTTP_X_MASHERY_OAUTH_ACCESS_TOKEN": "K5d7eg6t4mfhya9wdcbpvsc44",
        "HTTP_X_MASHERY_OAUTH_CLIENT_ID": "d5mn4p2svheyxje2bf98b7bt",
        "HTTP_X_MASHERY_OAUTH_SCOPE": "inventory:customer",
        "HTTP_X_MASHERY_OAUTH_USER_CONTEXT": "{\"authId\":\"LXzcGFueUlkOiAY43b4MTUw\",\"compan
yId\":\"4009988\"}"
    }
}
```

If the token has expired, the API will return a *401 Unauthorized* status code and the message "Developer Inactive".

```
<h1>Developer Inactive</h1>
```