Oracle® Cloud

CLI Reference for Oracle Mobile Hub

Release 20.2.3
E99925-05
August 2020

Oracle Cloud CLI Reference for Oracle Mobile Hub, Release 20.2.3

E99925-05

Primary Author: Patrick Keegan

Contents

Preface

Welcome to Oracle Mobile Hub.

Audience

This guide is intended for developers who use the command line reference (CLI) for Oracle Mobile Hub (Mobile Hub) to create and deploy artifacts for mobile applications and intelligent chatbots.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1 Overview

The Oracle Mobile Hub (Mobile Hub) command-line interface (CLI) enables you to create and manage artifacts without interacting with the Mobile Hub UI.

Here are some of the things you can do:

- Generate new artifacts, such as backends, APIs, and bots.
- Associate APIs, bots, and other artifacts with backends.
- Manage policies for a service instance.

Syntax

```
omce [options] <command>:<subcommand> [parameters]
```

Global Options

The following options are available for each command:

`-h` `--help`	Help for the given command
`-v` `--verbose`	Enables verbose output for debugging purposes

2
Setup

Here's what you need to do to set up the CLI on your system:

1. Install Node.js on your system.
2. Download the CLI from the Oracle Technology Network site.
3. Install the CLI:

   ```
   npm install -g <PATH_TO_CLI>oracle-omce-cli-1.0.0.tgz
   ```

4. Get the instance's details and credentials from the service's UI by clicking  to open the side menu, selecting **Development**, and then clicking the **Instance Details** link.

Logging In

First you declare the instance, where `<environment_alias>` is a name of your choosing to refer to the instance in subsequent commands:

```
omce env:add   <environment_alias> \
 --base-url "<BASE_URL>" \
  --token-endpoint "<TOKEN_ENDPOINT>" \
  --client <CLIENT_ID> \
  --secret <CLIENT_SECRET>
```

Then you log in:

```
omce env:login
```

3 Commands

api

Use this set of commands to create, delete, and manage custom APIs and their implementation code.

api:create

`omce api:create <api-name>/<api-version> --backend <backend-name>/<backend-version>`

Options

`-b <backend_name>/<backend-version>` `--backend <backend_name>/<backend-version>`	(Required) Associate the API with the named backend. If such a backend doesn't exist, create it.
`-e <environment_alias>` `--env <environment_alias>`	Associate the API with the given environment. (You must be logged in to that environment.)
`-r <raml_file>` `--raml <raml_file>`	Path to the RAML file on which to base the API.
`-t <template_name>` `--template <template_name>`	Generate the API's scaffolding based on the named template. Right now, the following template is available: • `bot`

Examples

Create an API called `FixItFastCustomer` (version 1.0).	`omce api:create FixItFastCustomer/1.0`

Create an API called `FixItFastCustomer` (version 1.0) and associate it with the backend `FixItFast`.	`omce api:create FixItFastCustomer/1.0 --backend FixItFast/1.0`
Create an API called `FixItFastCustomer` (version 1.0), associate it with the backend `FixItFast`, and generate scaffolding based on the `bot` template.	`omce api:create FixItFastCustomer/1.0 --template bot --backend FixItFast/1.0`

api:curl

Print a cURL command scaffold for calling an API. The scaffold is generated with an authorization token for the backend you are working with.

`omce api:curl <api-name>/<api-version> --backend <backend-name>/<backend-version>`

Options

`-b <backend_name>/<backend-version>` `--backend <backend_name>/<backend-version>`	(Required) Target backend.
`-e <environment_alias>` `--env <environment_alias>`	Target environment. (You must be logged in to that environment.)
`–u <Auth_method>` `--auth <Auth_method>`	Authorization type for the API. Valid values: • `anonymous` • `basic` • `oauth` Defaults to `anonymous` if not otherwise specified.

api:delete

Delete an API.

`omce api:delete <api-name>/<api-version>`

Options

`-e <environment_alias>` `--env <environment_alias>`	Specify the environment where the API is hosted. (You must be logged in to that environment.)

api:deploy

Create a zip file of a custom code implementation and upload it to an environment. Unless you specify the `--dir` option, you call this command from the source directory of the implementation.

```
omce api:deploy <api-name>/<api-version>
```

Options

`-d <dir_name>` `--dir <dir_name>`	Source directory of the implementation. If none specified, the current directory is used.
`-e <environment_alias>` `--env <environment_alias>`	Target environment for the API. (You must be logged in to that environment.)

api:fetch

Download the default implementation for the API.

```
omce api:fetch <api-name>/<api-version>
```

Options

`-d <dir_name>` `--dir <dir_name>`	Target directory for unpacking the implementation. By default, it's the current directory.
`-e <environment_alias>` `--env <environment_alias>`	Associate the API with the given environment. (You must be logged in to that environment.)

api:list

```
omce api:list
```

Options

`-e <environment_alias>` `--env <environment_alias>`	The environment containing the API you want shown. (You must be logged in to that environment.)

api:scaffold

Create a scaffold for the API's implementation. The scaffold takes the form of a Node.js module.

```
omce api:scaffold <api-name>/<api-version>
```

Options

`-d <dir_name>` `--dir <dir_name>`	Source directory of the API. If none specified, the current directory is used.
`-e <environment_alias>` `--env <environment_alias>`	Target environment for the API. (You must be logged in to that environment.)
`-t <template_name>` `--template <template_name>`	Generate the API's scaffolding based on the named template. Right now, the following template is available: • `bot`

api:security

Name the roles that can access the API (or enable anonymous user access).

```
omce api:security <api-name>/<api-version>
```

Options

`-a` `--anonymous`	Enable anonymous user access.
`-e <environment_alias>` `--env <environment_alias>`	Target environment for the API. (You must be logged in to that environment.)
`-o <roles>` `--roles <roles>`	List of roles, separated by commas.

api:serve

Run the API's custom code in a local container.

```
omce api:serve <api-name>/<api-version> --backend <backend-name>/
<backend-version>
```

Options

–a –auto	Automatically create debugger gateway API.
–b <backend_name>/<backend_version> --backend <backend_name>/<backend_version>	(Required) Target backend.
-c <container_name> --container <container_name>	Source path for the custom code container
-d --debug	Enable node debugger
-e <environment_alias> --env <environment_alias>	Associate the API with the given environment. (You must be logged in to that environment.)
–m <Node_version> --compat <Node_version>	Node version compatibility: • For customers up to Oracle Mobile Hub 19.4.3, the default is 8.9. • For customers from Oracle Mobile Hub 20.1.3 onwards, the default is 12.16. The default value is stored in CCC_DefaultNodeConfigurationValue. Valid values: • 6.10 • 8.9 • 12.16
–n <Node_installation_home> --node <Node_installation_home>	Location of node installation on your system.

–p <*port_number*> --port <*port_number*>	Local port to listen on. The default is 4000
–s <*src_path*> --src <*src_path*>	Source code path.

api:update

```
omce api:update <api-name>/<api-version>
```

-e <*environment_alias*> --env <*environment_alias*>	Associate the API with the given environment. (You must be logged in to that environment.)
-r <*raml_file*> --raml <*raml_file*>	Path to the RAML file on which to base the API.

backend

Use this set of commands to create, delete, and manage backends.

backend:add

Associate an artifact with a backend.

```
omce backend:add <backend_name>/<backend_version>
```

Options

-a <*api_name*>/<*api_version*> --api <*api_name*>/<*api_version*>	API to add to the backend.
-e <*environment_alias*> --env <*environment_alias*>	Associate the backend with the given environment. (You must be logged in to that environment.)

Examples

Add the FiFAPI/1.0 API to the FiFBackend/1.0 backend.	`omce backend:add FiFBackend/1.0 --api FiFAPI/1.0`

backend:create

Create a backend.

`omce backend:create <backend_name>/<backend_version>`

Options

`-e <environment_alias>` `--env <environment_alias>`	Associate the backend with the given environment. (You must be logged in to that environment.)

backend:delete

Delete a backend.

`omce backend:delete <backend_name>/<backend_version>`

Options

`-e <environment_alias>` `--env <environment_alias>`	Target environment containing the backend you are deleting. (You must be logged in to that environment.)

backend:list

List available backends.

`omce backend:list`

Options

`-e <environment_alias>` `--env <environment_alias>`	Environment containing the backends that you want listed. (You must be logged in to that environment.)

backend:remove

Remove an API from a backend.

`omce backend:remove <backend_name>/<backend_version> --api <api_name>/<api_version>`

Options

`-a <api_name>/<api_version>`	(Required) API to remove from the backend.

`--api <api_name>/<api_version>`	
`-e <environment_alias>` `--env <environment_alias>`	Environment containing the backend that you are removing an artifact from. (You must be logged in to that environment.)

Examples

Remove the `FiFAPI/1.0` API from the `FiFBackend/1.0` backend.	`omce backend:remove FiFBackend/1.0 --api FiFAPI/1.0`

bot

Use this set of commands to manage your bots.

bot:service:create

Add a custom component service to a bot so that the bot can access to the custom components of a given API.

```
omce bot:service:create <bot_name> --service
<custom_component_service_name> --backend <backend_name>/
<backend_version> FixItFast/1.0 --api <api_name>/<api_version> --auth
<auth_type>
```

Options

`-a <api_name>/<api_version>` `--api <api_name>/<api_version>`	(Required) API containing the custom components that the bot needs.
`-b <backend_name>/<backend_version>` `--backend <backend_name>/<backend_version>`	(Required) Backend that the bot is associated with.
`-e <environment_alias>` `--env <environment_alias>`	Target environment for the bot. (You must be logged in to that environment.)
`-m <environment_alias>` `--mobile-env <environment_alias>`	Use a backend and/or API that is deployed to a different environment.
`-s <custom_component_service_name>` `--service <custom_component_service_name>`	(Required) Custom code service name.

–u *<auth_method>* --auth *<auth_method>*	(Required) Authorization type for the API. Valid values: • anonymous • basic

Examples

Add the FixItFast_AMCe component service to the FixItFastBot so that it can access the custom components of the FixItFast/1.0 API.	omce bot:service:add FixItFastBot --service-name FixItFast_AMCe --backend FixItFast/1.0 --api FixItFast/1.0 --auth anonymous

bot:service:list

List the custom component services associated with a bot.

omce bot:service:list *bot_name*

Options

-e *<environment_alias>* --env *<environment_alias>*	Target environment for the bot. (You must be logged in to that environment.)

env

Use this set of commands to manage environments that you are accessing through the CLI.

env:add

Add an environment.

omce env:add *<environment_alias_of_your_choosing>* --base-url *<base_URL>* --token-endpoint *<token_endpoint>* --client *<client_id>* --secret *<client_secret>*

-b *<base_URL>* --base-url *<base_URL>*	(Required) Base URL for the environment.
-c *<client_ID>*	(Required) Your team member client ID.

`--client <client_ID>`	
`-s <client_secret>` `--secret <client_secret>`	(Required) Your team member client secret.
`-t <token_endpoint>` `--token-endpoint <token_endpoint>`	(Required) OAuth token endpoint for your environment.

env:delete

Remove environment from list of environments you are working with in this session.

```
omce env:delete <environment_alias>
```

env:list

List environments that you have added in this session.

```
omce env:list
```

env:login

Log into an environment.

```
omce env:login
```

Options

`-e <environment_alias>` `--env <environment_alias>`	Environment to log into.

policy

Use this set of commands to manage policies in your environment.

policy:set

Set a value for an environment policy.

```
omce policy:set policy_name policy_value
```

Options

`-e <environment_alias>`	Target environment.

`--env <environment_alias>`	