

Oracle® Cloud

CLI Reference for Oracle Mobile Hub



Release 20.2.3

E99925-05

August 2020

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Cloud CLI Reference for Oracle Mobile Hub, Release 20.2.3

E99925-05

Copyright © 2018, 2020, Oracle and/or its affiliates.

Primary Author: Patrick Keegan

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

	Preface	
	Audience	v
	Documentation Accessibility	v
1	Overview	
	Syntax	1-1
	Global Options	1-1
2	Setup	
3	Commands	
	api	3-1
	api:create	3-1
	api:curl	3-2
	api:delete	3-2
	api:deploy	3-3
	api:fetch	3-3
	api:list	3-3
	api:scaffold	3-4
	api:security	3-4
	api:serve	3-5
	api:update	3-6
	backend	3-6
	backend:add	3-6
	backend:create	3-7
	backend:delete	3-7
	backend:list	3-7
	backend:remove	3-7
	bot	3-8
	bot:service:create	3-8

bot:service:list	3-9
env	3-9
env:add	3-9
env:delete	3-10
env:list	3-10
env:login	3-10
policy	3-10
policy:set	3-10

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

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

<pre>-b <backend_name>/<backend-version> --backend <backend_name>/<backend-version></pre>	(Required) Associate the API with the named backend. If such a backend doesn't exist, create it.
<pre>-e <environment_alias> --env <environment_alias></pre>	Associate the API with the given environment. (You must be logged in to that environment.)
<pre>-r <raml_file> --raml <raml_file></pre>	Path to the RAML file on which to base the API.
<pre>-t <template_name> --template <template_name></pre>	Generate the API's scaffolding based on the named template. Right now, the following template is available: <ul style="list-style-type: none">• bot

Examples

Create an API called <code>FixItFastCustomer</code> (version 1.0).	<pre>omce api:create FixItFastCustomer/1.0</pre>
--	--

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

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

<code>-b <backend_name>/<backend-version></code> <code>--backend <backend_name>/<backend-version></code>	(Required) Target backend.
<code>-e <environment_alias></code> <code>--env <environment_alias></code>	Target environment. (You must be logged in to that environment.)
<code>-u <Auth_method></code> <code>--auth <Auth_method></code>	Authorization type for the API. Valid values: <ul style="list-style-type: none"> • anonymous • basic • oauth Defaults to <code>anonymous</code> if not otherwise specified.

api:delete

Delete an API.

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

Options

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

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

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

api:fetch

Download the default implementation for the API.

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

Options

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

api:list

```
omce api:list
```

Options

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

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

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

api:security

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

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

Options

<code>-a</code> <code>--anonymous</code>	Enable anonymous user access.
<code>-e <environment_alias></code> <code>--env <environment_alias></code>	Target environment for the API. (You must be logged in to that environment.)
<code>-o <roles></code> <code>--roles <roles></code>	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

<pre>-a -auto</pre>	Automatically create debugger gateway API.
<pre>-b <backend_name>/ <backend_version> --backend <backend_name>/ <backend_version></pre>	(Required) Target backend.
<pre>-c <container_name> --container <container_name></pre>	Source path for the custom code container
<pre>-d --debug</pre>	Enable node debugger
<pre>-e <environment_alias> --env <environment_alias></pre>	Associate the API with the given environment. (You must be logged in to that environment.)
<pre>-m <Node_version> --compat <Node_version></pre>	<p>Node version compatibility:</p> <ul style="list-style-type: none"> • 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. <p>The default value is stored in CCC_DefaultNodeConfigurationValue.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • 6.10 • 8.9 • 12.16
<pre>-n <Node_installation_home> --node <Node_installation_home></pre>	Location of node installation on your system.

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

api:update

once api:update <api-name>/<api-version>

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

backend

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

backend:add

Associate an artifact with a backend.

once backend:add <backend_name>/<backend_version>

Options

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

Examples

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

backend:create

Create a backend.

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

Options

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

backend:delete

Delete a backend.

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

Options

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

backend:list

List available backends.

```
omce backend:list
```

Options

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

backend:remove

Remove an API from a backend.

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

Options

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

<code>--api <api_name>/<api_version></code>	
<code>-e <environment_alias></code> <code>--env <environment_alias></code>	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.	<code>omce backend:remove FiFBackend/1.0</code> <code>--api FiFAPI/1.0</code>
--	--

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

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

<code>-u <auth_method></code>	(Required) Authorization type for the API.
<code>--auth <auth_method></code>	Valid values:
	<ul style="list-style-type: none"> • anonymous • basic

Examples

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

bot:service:list

List the custom component services associated with a bot.

```
omce bot:service:list bot_name
```

Options

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

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>
```

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

<code>--client <client_ID></code>	
<code>-s <client_secret></code> <code>--secret <client_secret></code>	(Required) Your team member client secret.
<code>-t <token_endpoint></code> <code>--token-endpoint <token_endpoint></code>	(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

<code>-e <environment_alias></code>	Environment to log into.
<code>--env <environment_alias></code>	

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

<code>-e <environment_alias></code>	Target environment.
---	---------------------

```
--env <environment_alias>
```
