
ORACLE®

Oracle Intelligent Bots Advanced Training

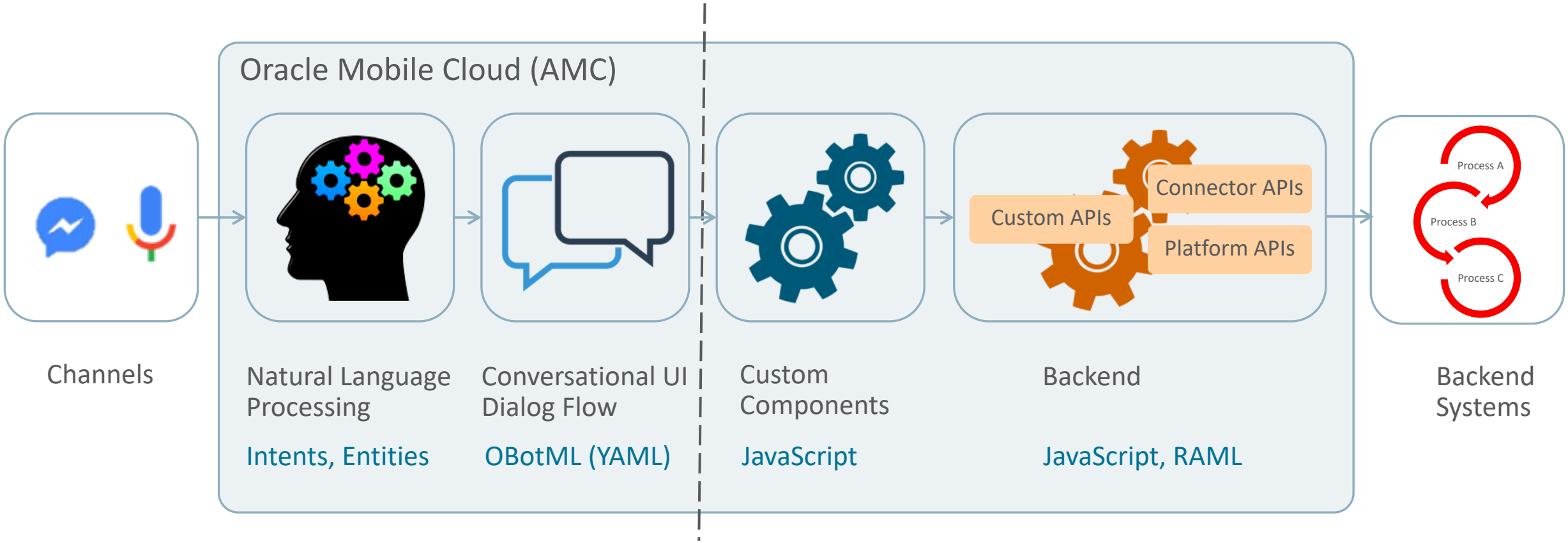
Custom Component Local Development & Debugging

ORACLE

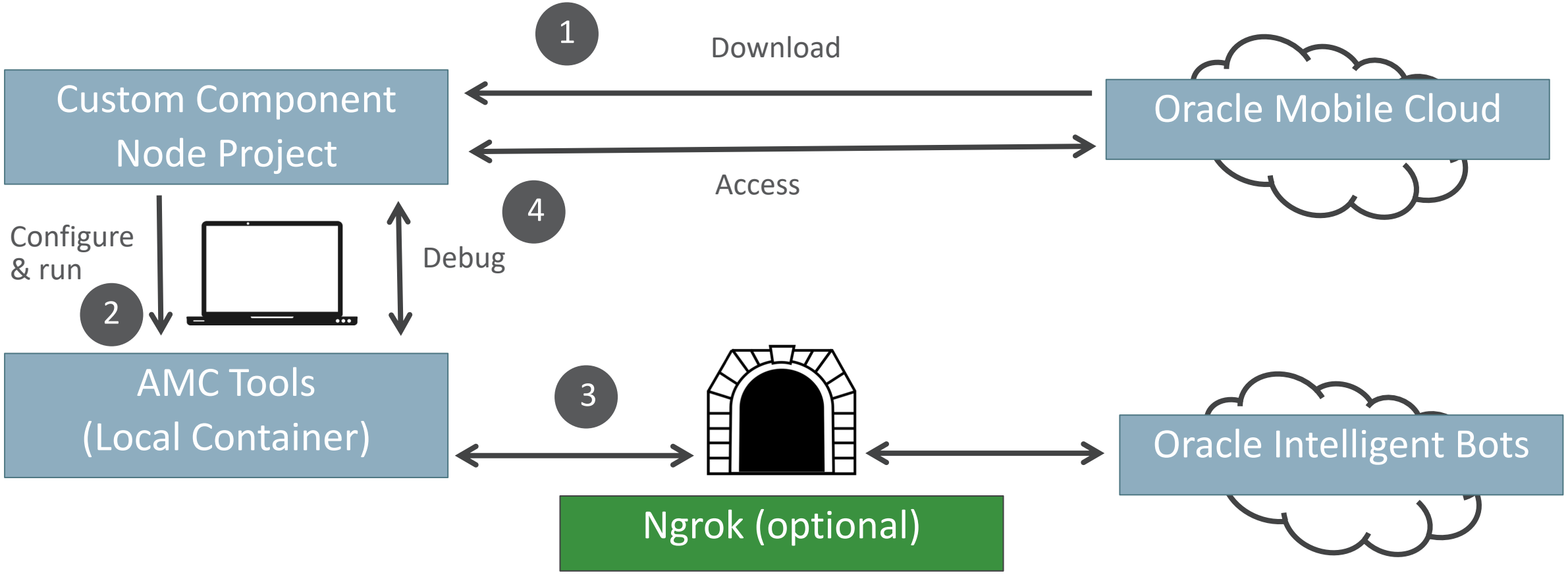
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Custom Components In Oracle AMC



AMC / Bot Local Development & Debugging Architecture



Topic Agenda

1. Install Custom Code Test Tools
2. Setup for Custom Component Project
3. Run Custom Component in Local Container from Oracle Intelligent Bots
4. Debug Custom Component in Local Container

Local Development & Debugging

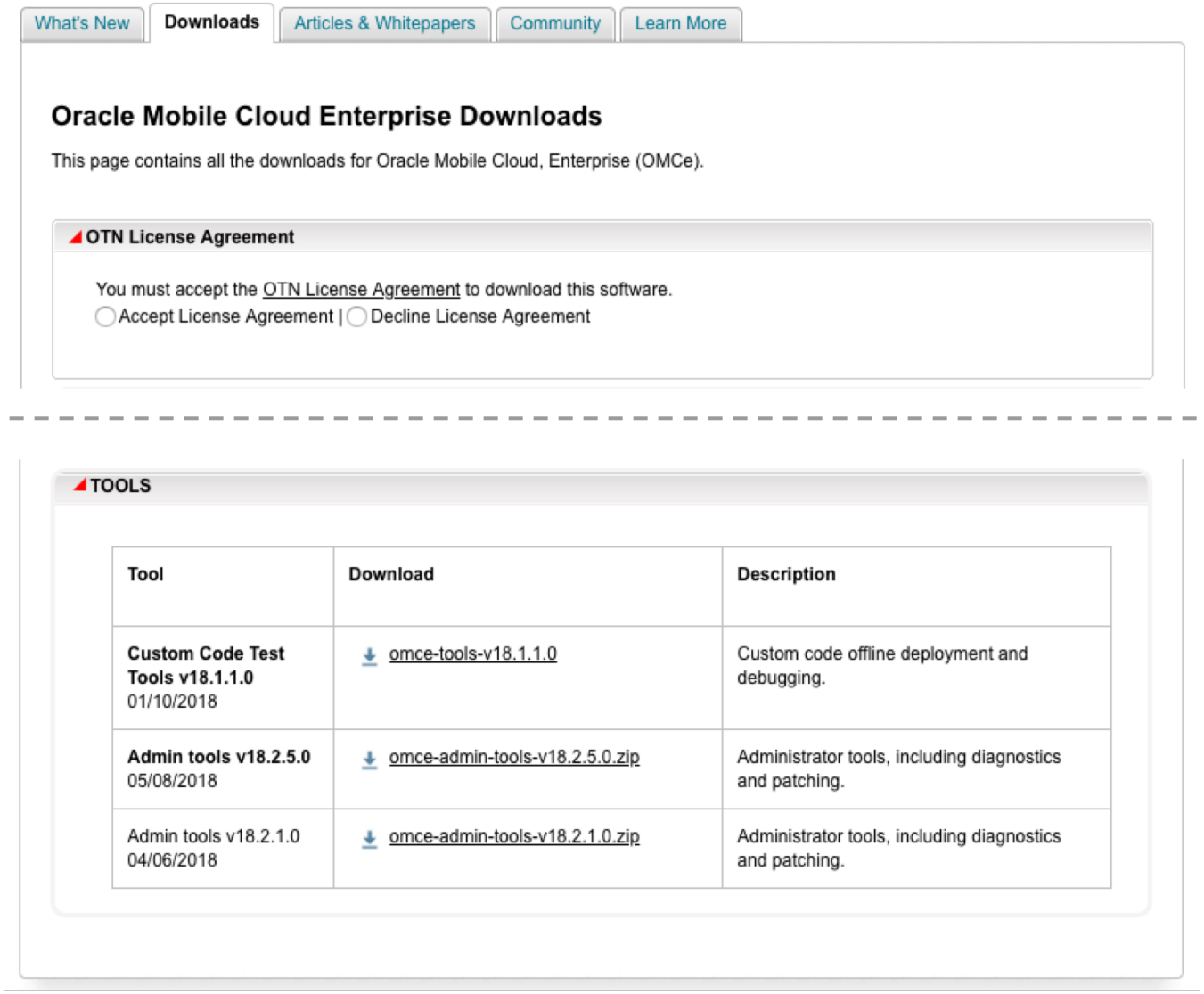
Install Custom Code Test Tools

Oracle Custom Code Test Tools

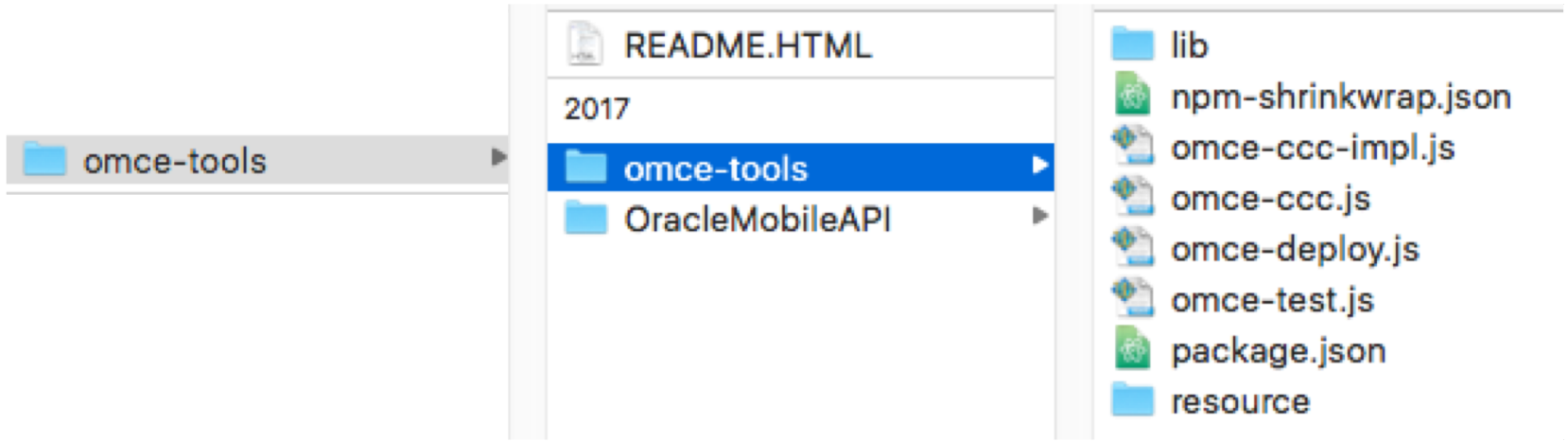
- Set of tools to locally test Custom API
  - Local node container provides AMC integration
- Tools
  - npm module that includes command-line tools
    - Starting a local custom code container
    - Run tests
    - Deploying an API to AMC
  - OracleMobileAPI to proxy AMC API calls from local node container to AMC
    - Needs to be uploaded as custom API to AMC

Custom Code Test Tools

- Download from OTN
  - Custom Code Test Tools
- Need to accept license agreement
- Chose most recent version
- Unzip download on local machine



Unzipped Custom Code Test Tools



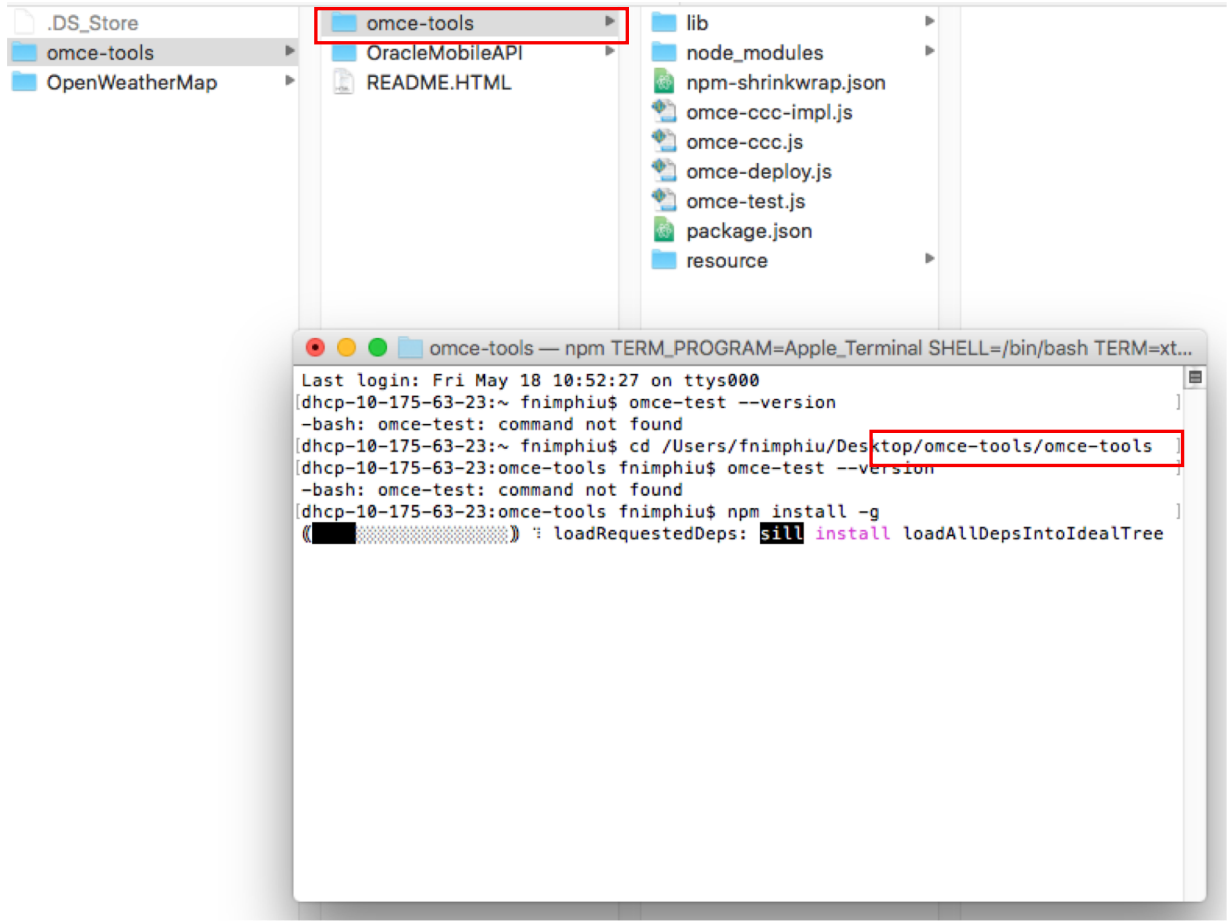
- omce-tools
  - Contains the actual command line
  - Starts the local node container
- OracleMobileAPI
  - AMC custom API acting as a proxy

You install the OracleMobileAPI **only once** per AMC instance. The custom code test tool also only needs to be installed only once for a device.

Image courtesy of Farbentek at FreeDigitalPhotos.net

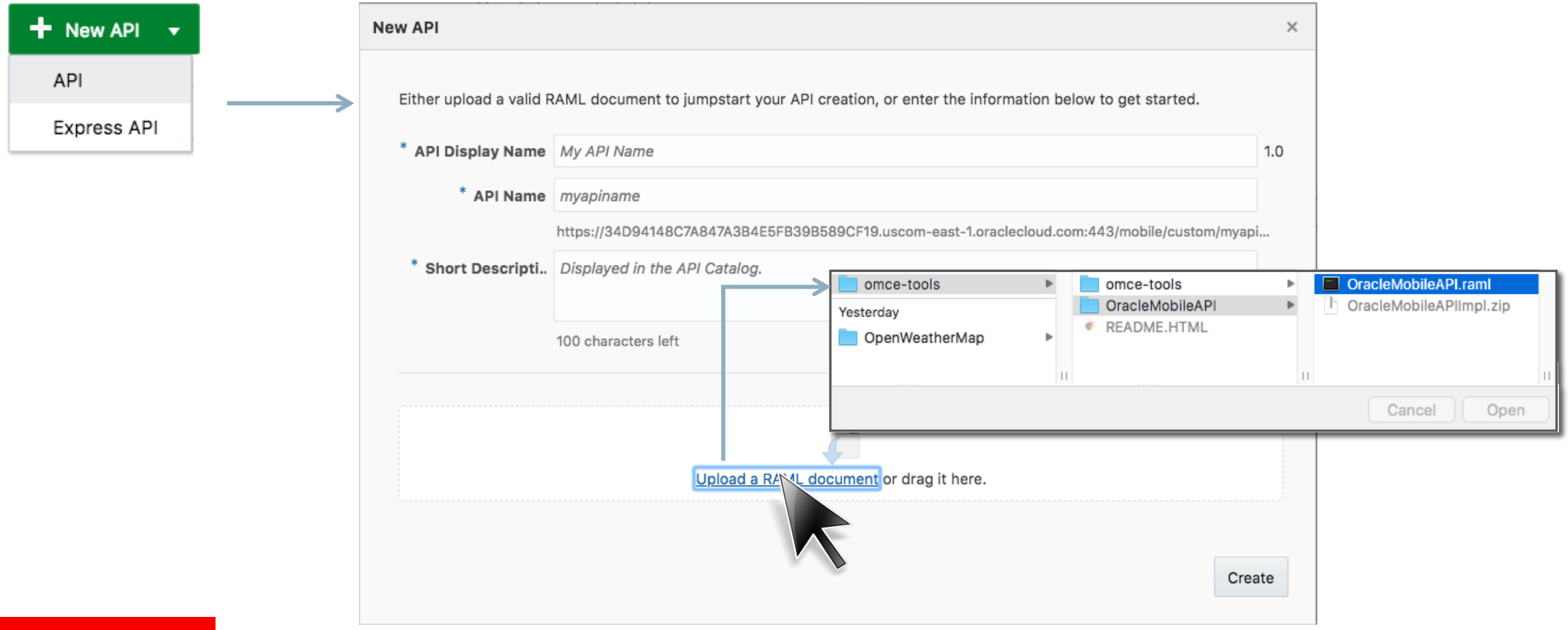
AMC Tools Installation

- Ensure you have Node package manager (npm) installed
- In terminal window, switch to omce-tools directory
- Issue *npm install -g*
  - Installation takes a minute or two
  - If installation fails try with "sudo" command
- Run *omce-test -- version* to verify installation success



Uploading the OracleMobileAPI Proxy to AMC

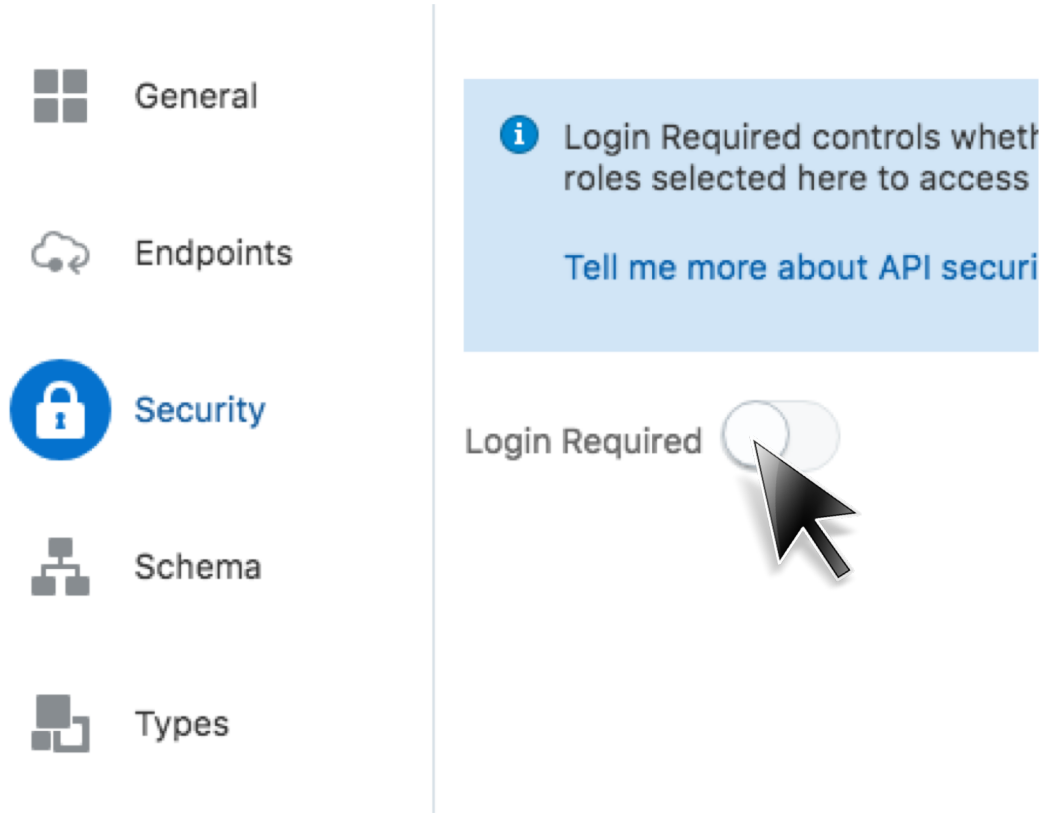
Create Custom API from RAML File



Uploading the OracleMobileAPI Proxy to AMC

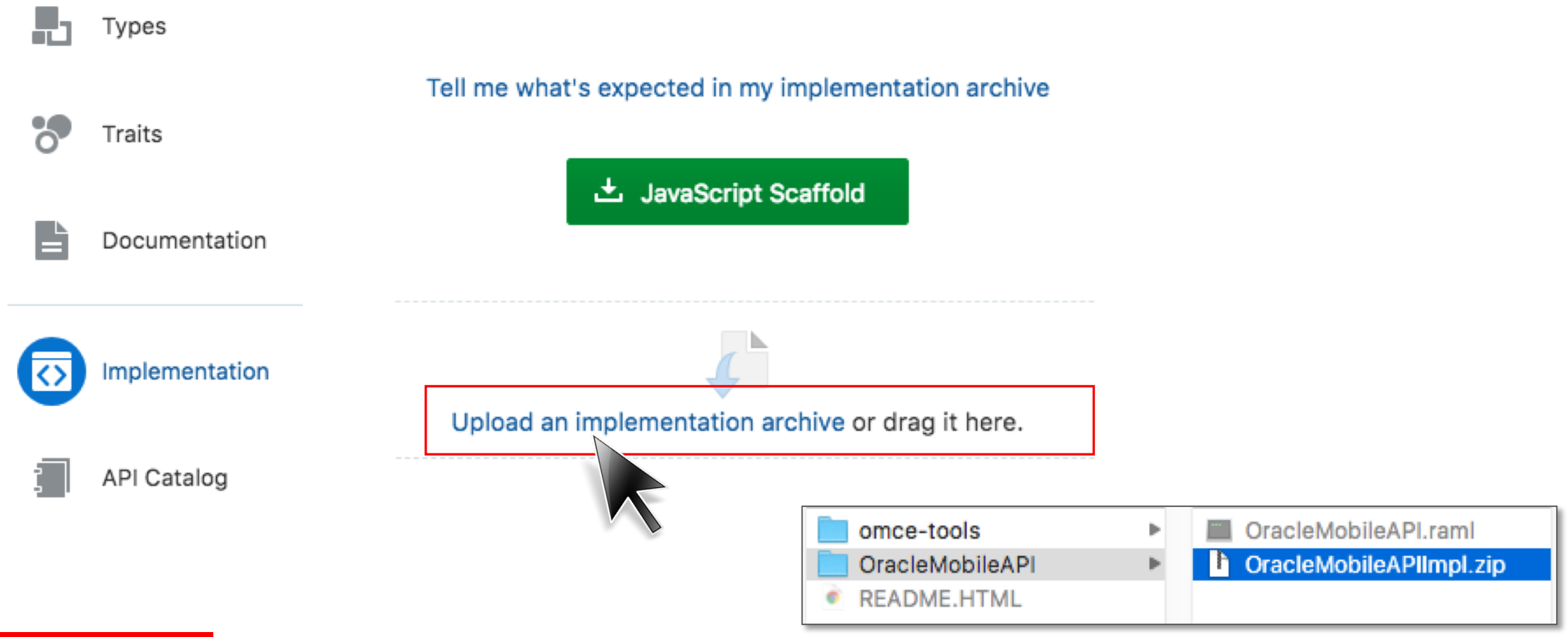
Disabling Named User Authentication

- Disable named user account authentication by unsetting the **Login Required** toggle
- Save your change
- Still, anonymous authentication will be required



Uploading the OracleMobileAPI Proxy to AMC

Upload Proxy Implementation Code

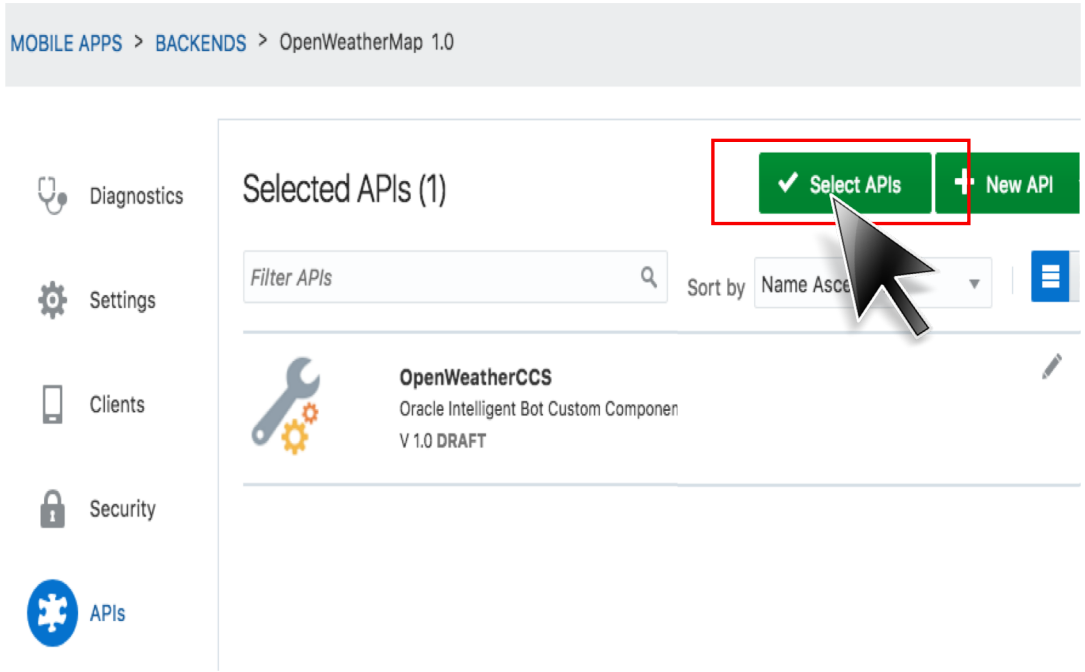


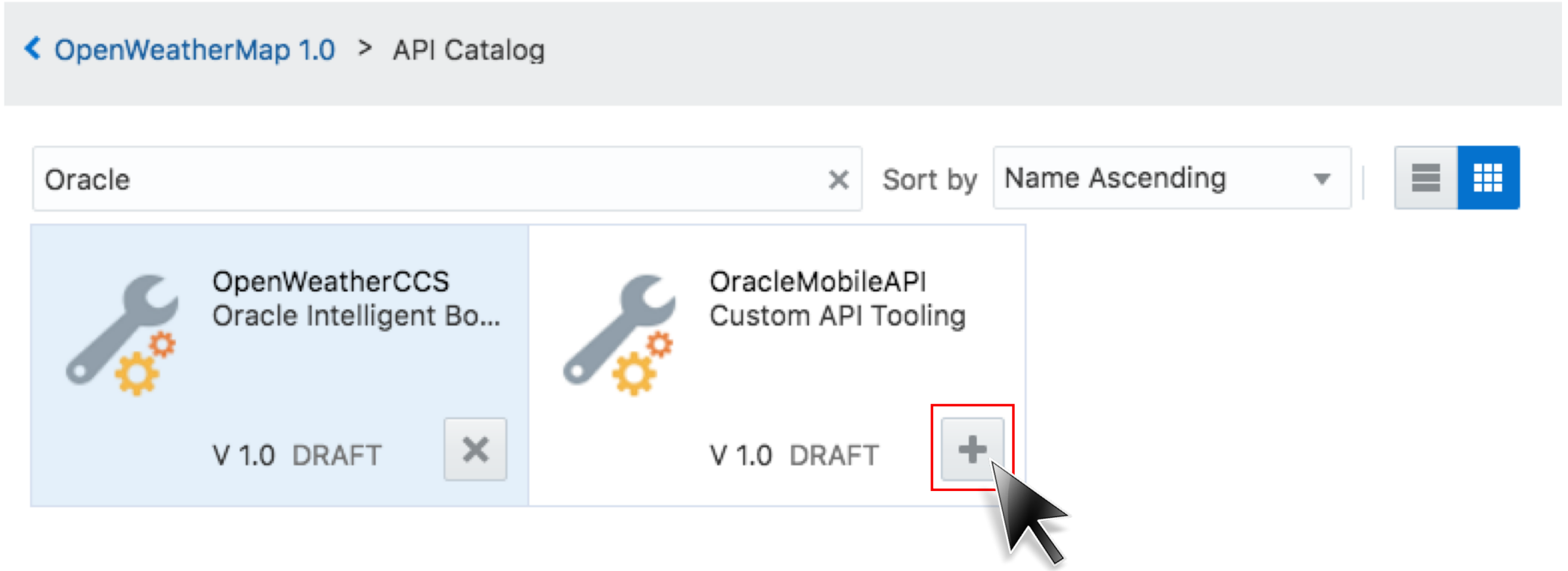
Local Development & Debugging

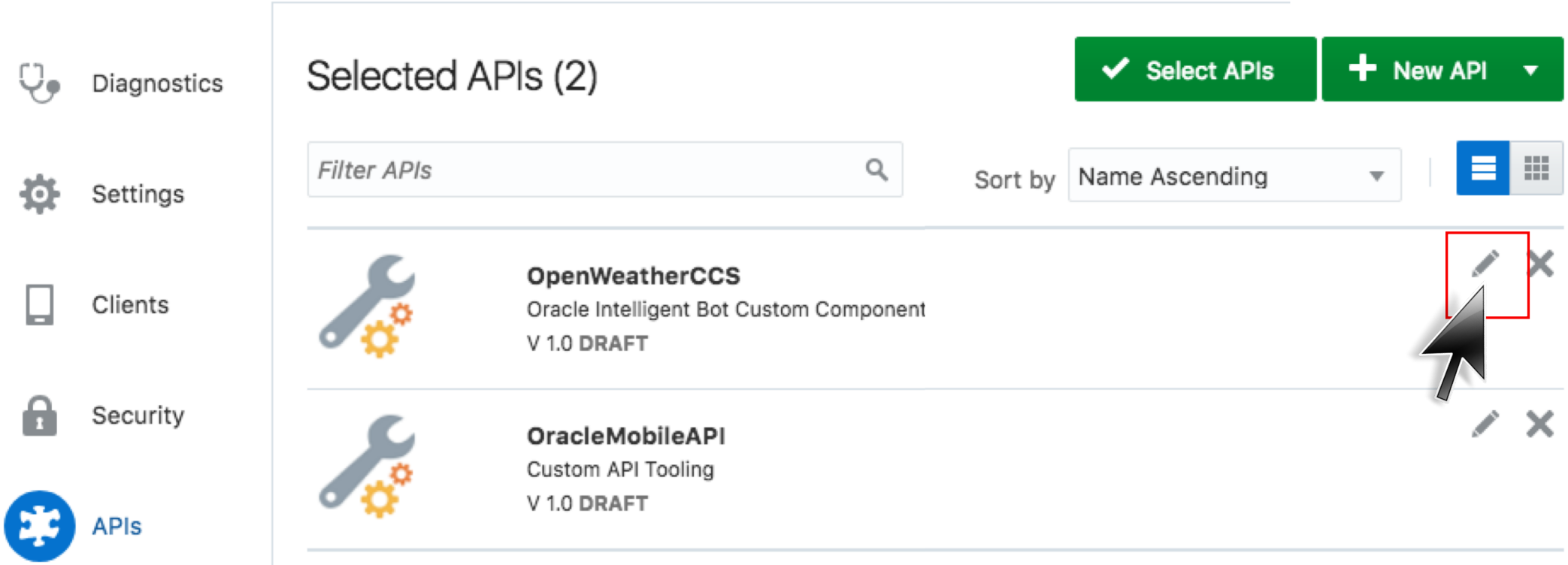
Setup for Custom Component Project

Adding the Oracle Mobile API to Your Backend

- Go to bot the mobile backend that exposes the bot custom component service
  - Open the mobile backend in the AMC dashboard
- Select the "APIs" menu item
- Select "Select APIs" to add the Oracle Mobile API to the mobile backend

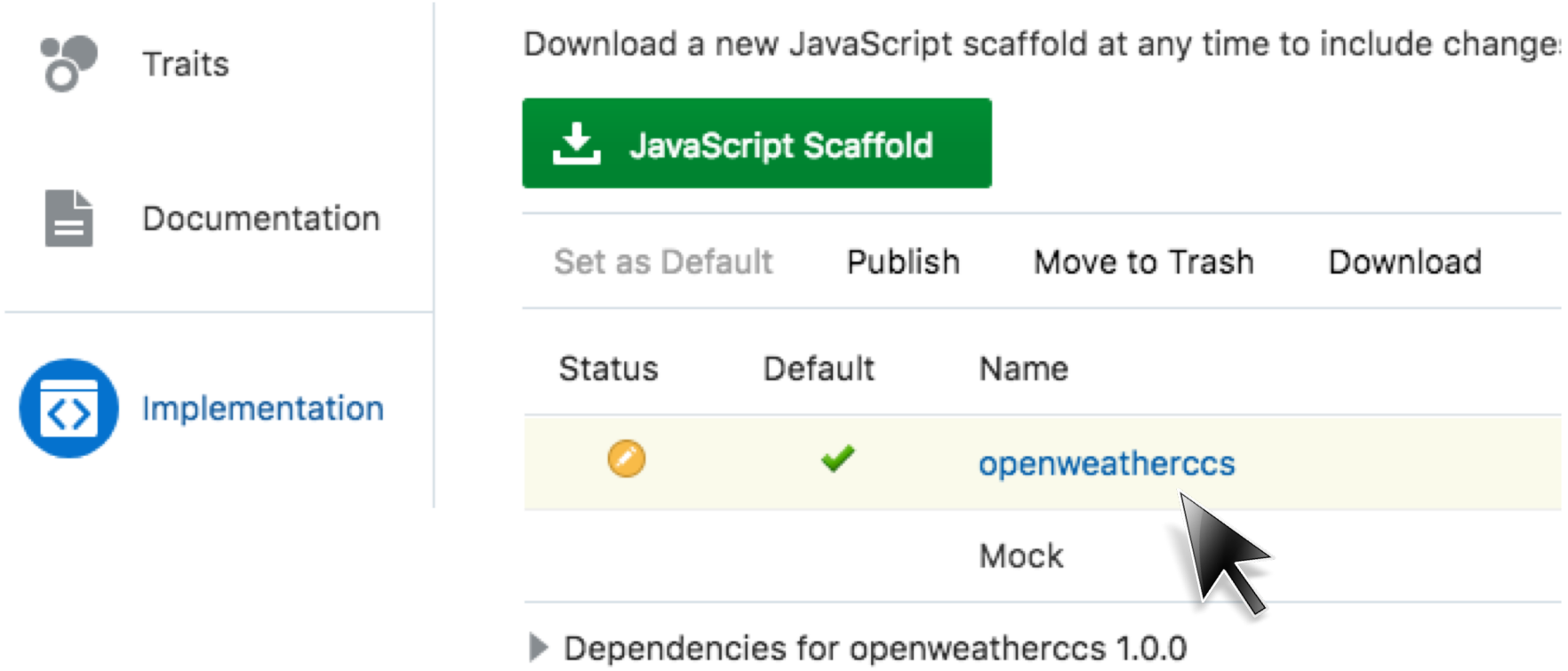






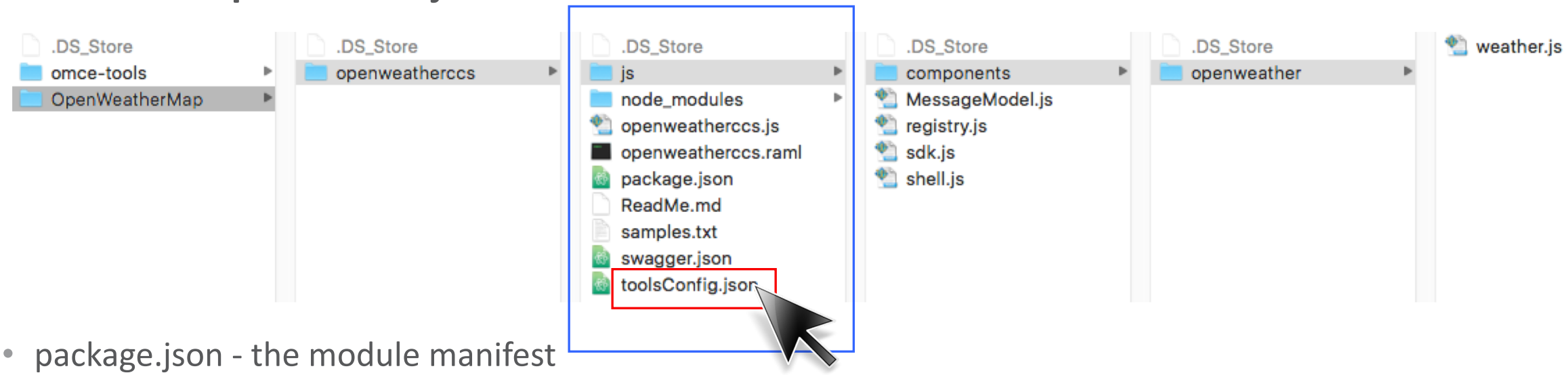
Download Custom Component Service

Implementation Scaffold



OpenWeatherMap Custom Component Project

Custom Component Project



- package.json - the module manifest
- <api name>.js - your starter implementation
- <api name>.raml - the API definition in RAML format
- swagger.json - the API definition in Swagger format
- toolsConfig.json - metadata needed by the tools

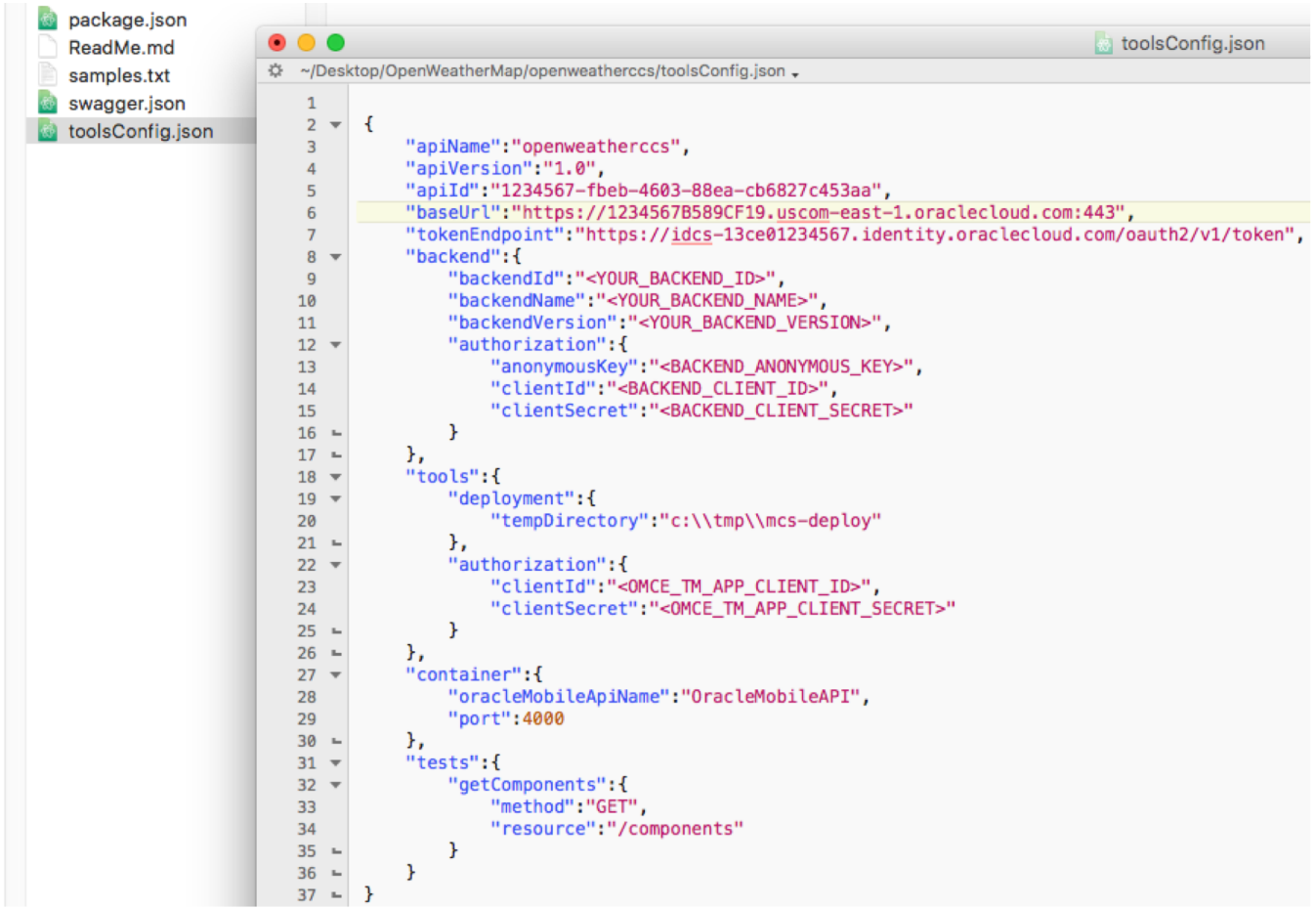
toolsConfig.json format is **different for migrated** Custom API projects

Handling Migrated Custom API Projects

- toolsConfig.json has a different format in Oracle MCS and Oracle AMC
  - Format may change in the future too
- To update the toolsConfig.json file
  - Go to custom API in AMC
  - Download and unzip API implementation
  - Download new scaffold
  - Copy toolsConfig.json to unzipped implementation
  - Re-upload implementation
    - But not before setting it up for debugging

Configuring the toolsConfig.json File

- Configure toolsConfig.json with settings from Mobile Backend
  - baseUrl
  - backendId
  - backendName
  - backendVersion
  - anonymousKey
  - clientId
  - clientSecret



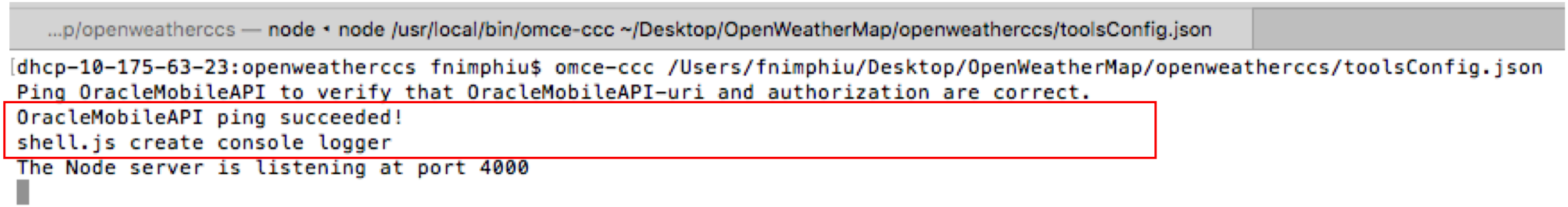
Local Development & Debugging

Run Custom Component in Local Container from Oracle Intelligent Bots

Starting the Local Container for Testing

Command:

omce-ccc <path to>/toolsConfig.json



Query Components Hosted by Custom Component Service

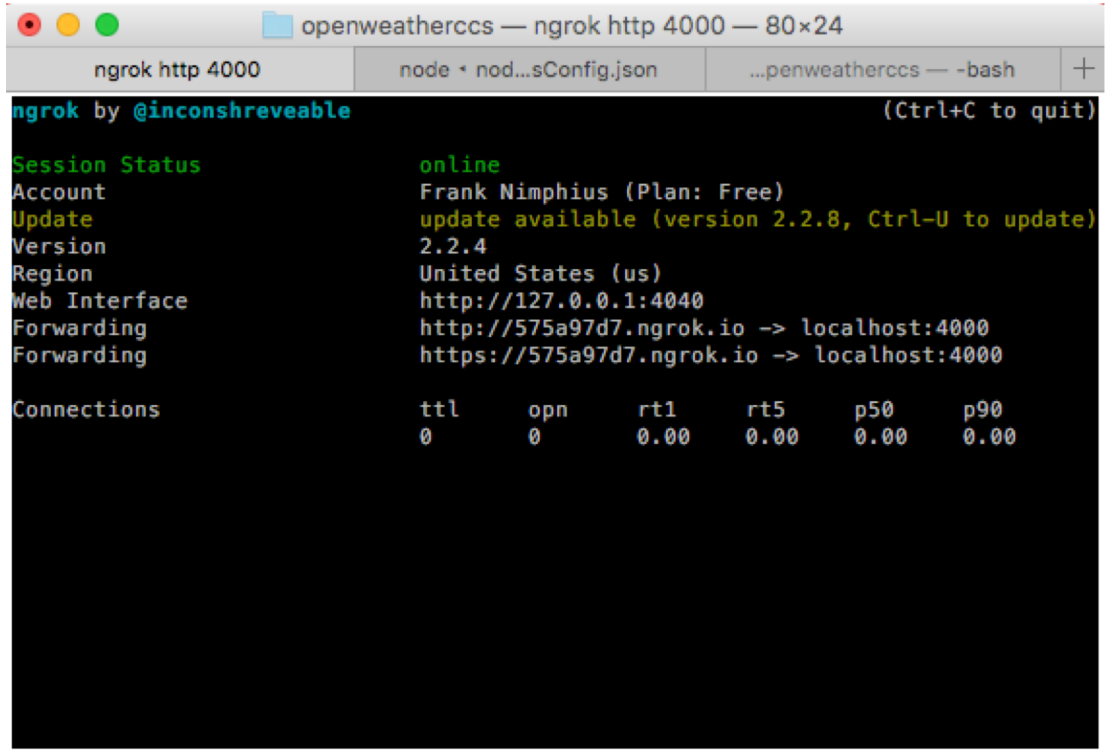
Command:

omce-test <path to>/toolsConfig.json getComponents

```
~/Desktop/OpenWeatherMap/openweatherccs — -bash    ...p/openweatherccs — node • node /usr/lo
fnimphiu-mac:openweatherccs fnimphiu$ omce-test /Users/fnimphiu/Desktop/OpenWeatherMap/openweatherccs/toolsConfig.json getComponents
Executing test getComponents. Method: GET, URL: http://localhost:4000/mobile/custom/openweatherccs/components
Response body:
{"version":"1.1","components":[{"name":"fn.weathercc","properties":{"city":{"type":"string","required":true},"country":{"type":"string
Test getComponents completed successfully with status 200
fnimphiu-mac:openweatherccs fnimphiu$
```

Using ngrok

- Exposes local services to public internet
  - Secure tunnel
- Setup
  - Install ngrok from https://ngrok.com/
  - In a terminal window, type *ngrok http 4000* and press the enter key
- (Optional) Inspect traffic in browser
  - http://localhost:4040

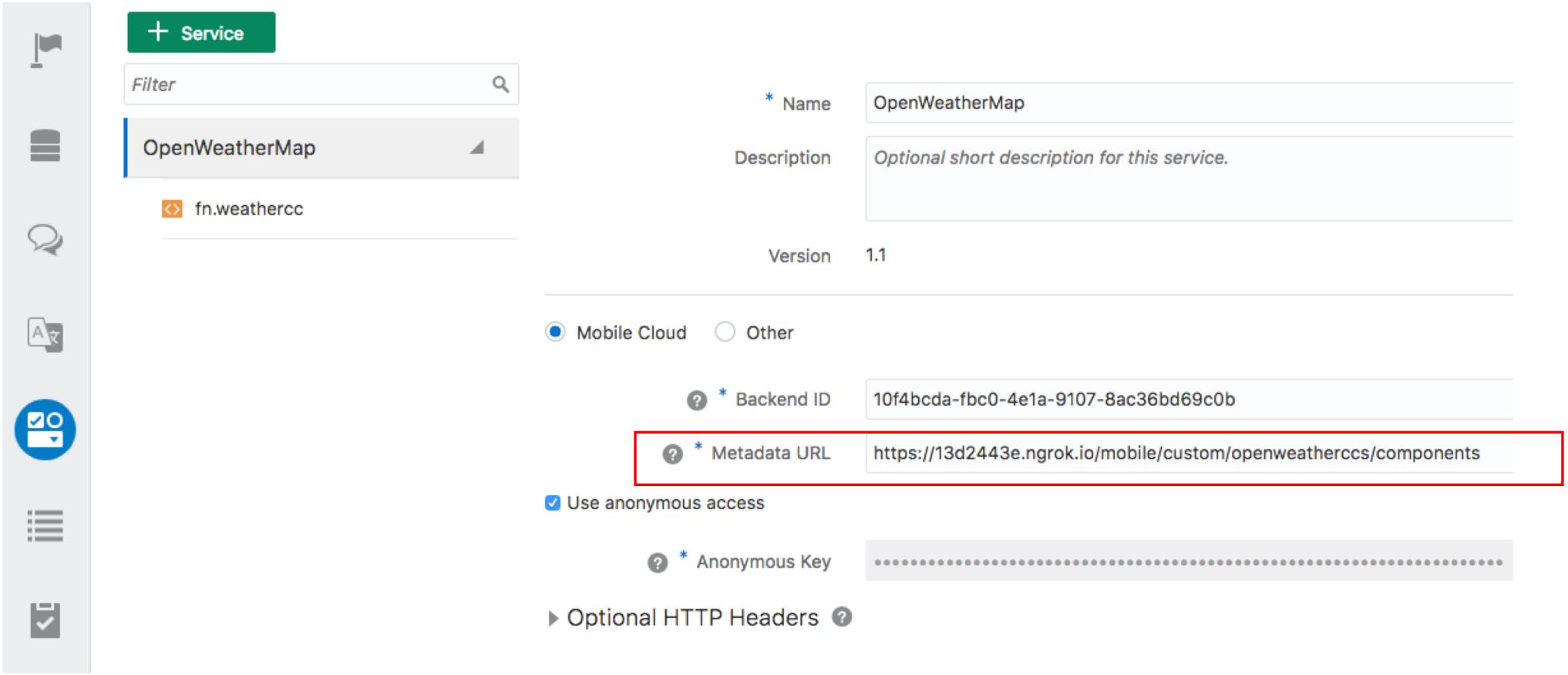


Running NGROK From Behind a Proxy

- Get the external IP address of your proxy
  - http://www.whatismyproxy.com/
- Issue an export statement in a terminal window
  - export https_proxy=http://<external ip>:80
  - export http_proxy=http://<external ip>:80
- Launch ngrok
  - ./ngrok http <port>

Configure Custom Component in Oracle Intelligent Bots

Using the locally deployed custom component



If your component does anything useful already, you can test it in the bot.

Local Development & Debugging

Starting the Local Container for Debugging

Starting the Local Container for Debugging

Command:

omce-ccc <path to>/toolsConfig.json --debug [--verbose]

--**debug**

Provides a URL you can use to open a debug session in Google Chrome.

--verbose

If you have errors or warnings, will show examples of the missing property in addition to a description of the property

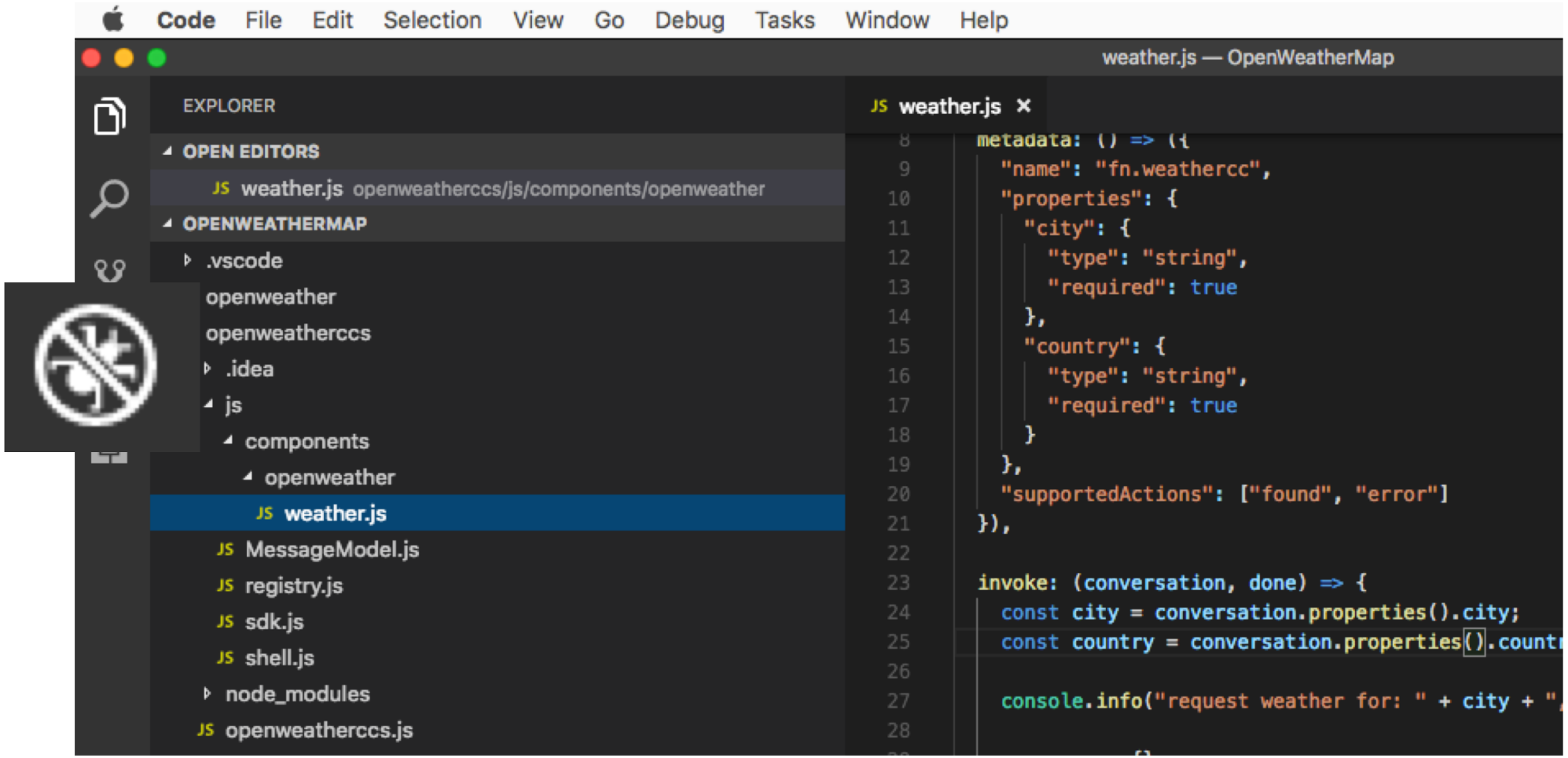
Starting the AMC Local Container in Debug Mode

```
dhcp-10-175-22-71:/ fnimphiu$ omce-ccc /Users/fnimphiu/Desktop/OpenWeatherMap/openweatherccs/toolsConfig.json --debug
Debugger listening on ws://127.0.0.1:9229/07cc0cf1-465e-4944-9ea2-750ea17cc30b
For help see https://nodejs.org/en/docs/inspector

Warning: Configuration property "proxy" is undefined

To display help and examples associated with warnings, use the --verbose option
Ping OracleMobileAPI to verify that OracleMobileAPI-uri and authorization are correct.
OracleMobileAPI ping succeeded!
shell.js create console logger
The Node server is listening at port 4000
```

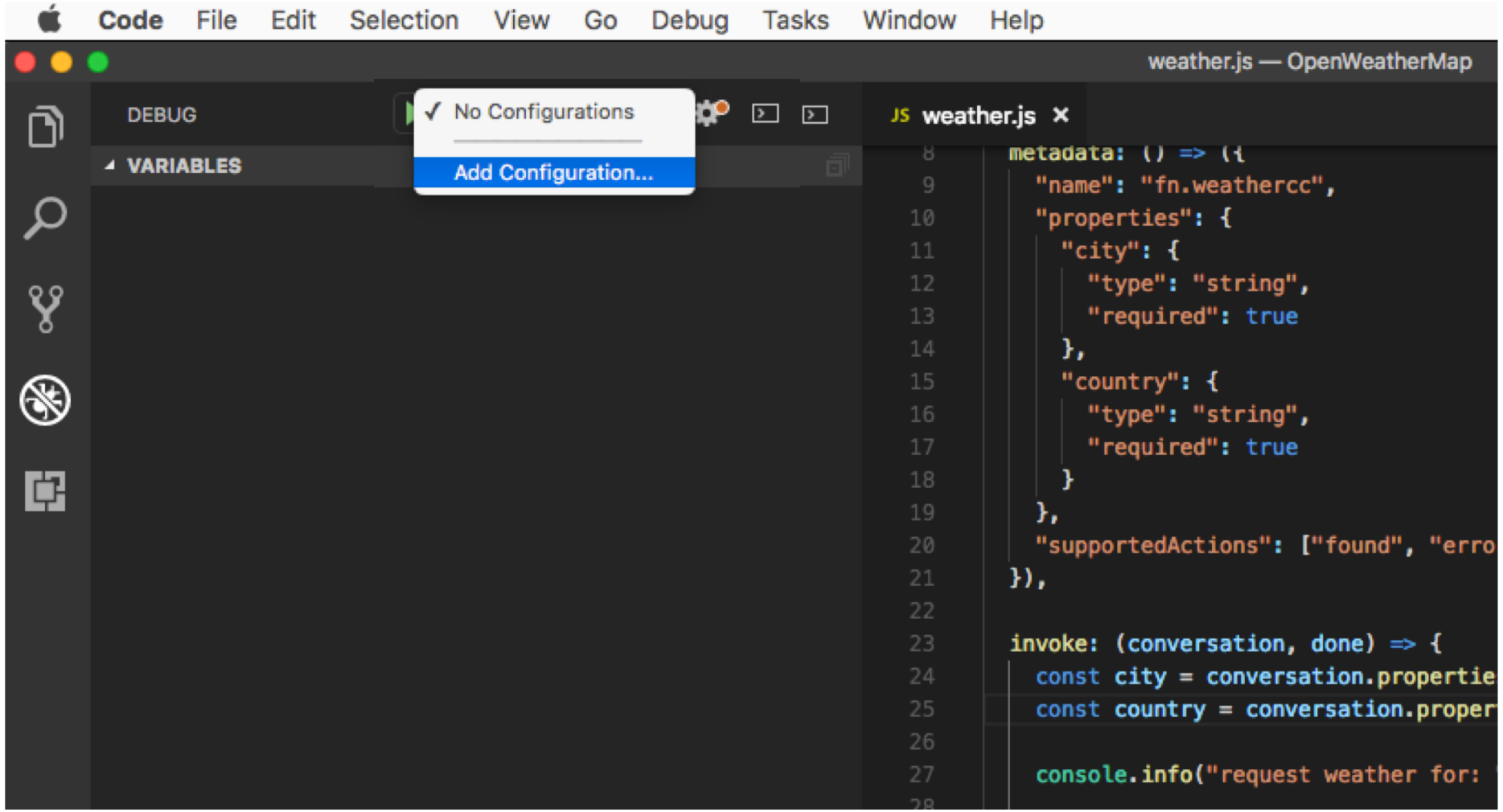
Local Development & Debugging

Debugging with Microsoft Visual Studio Code

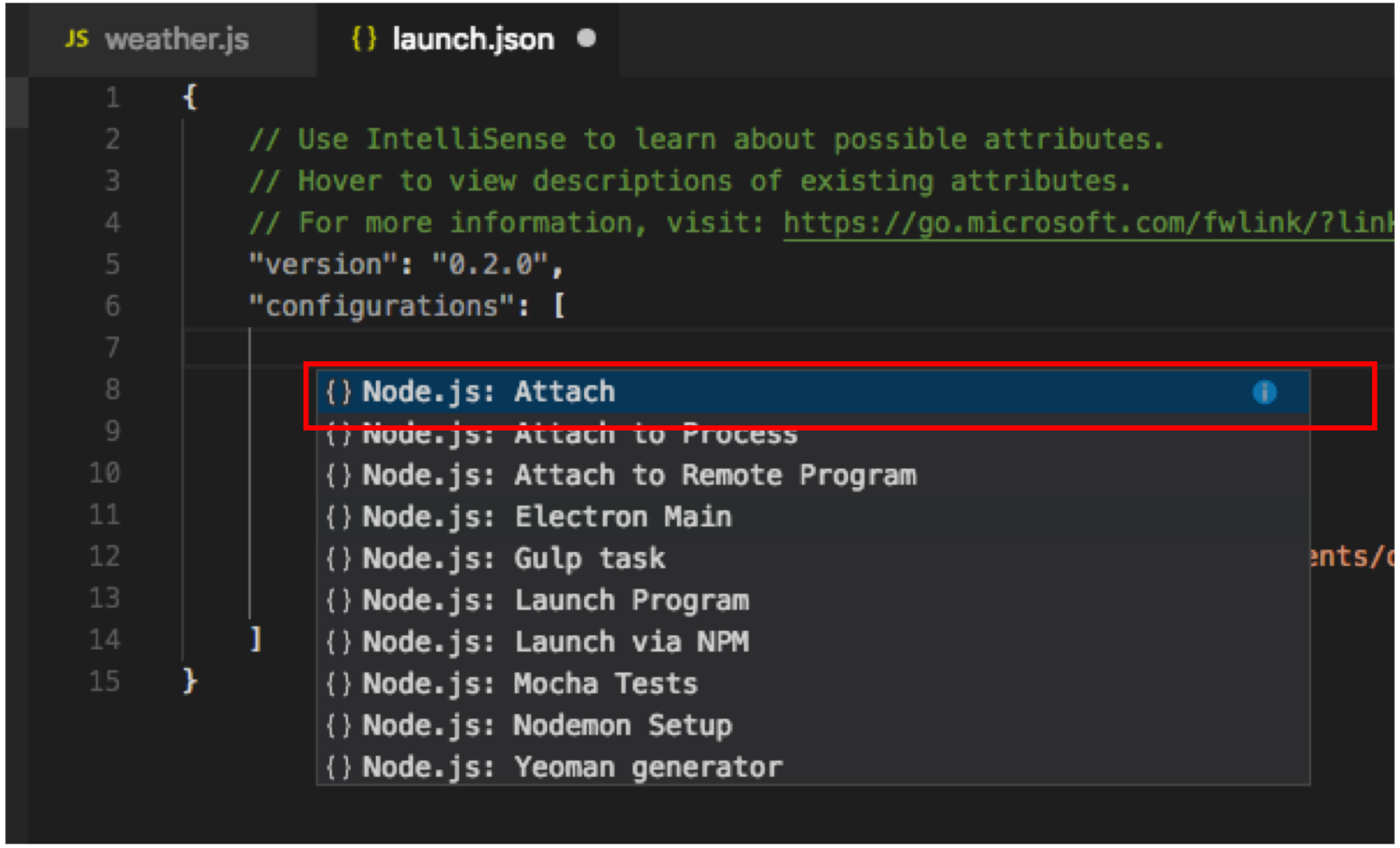
Debugging With Microsoft Visual Studio Code IDE



Debugging With Microsoft Visual Studio Code IDE

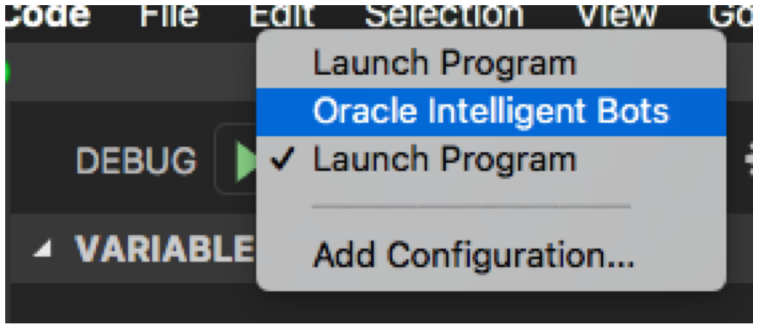


Debugging With Microsoft Visual Studio Code IDE

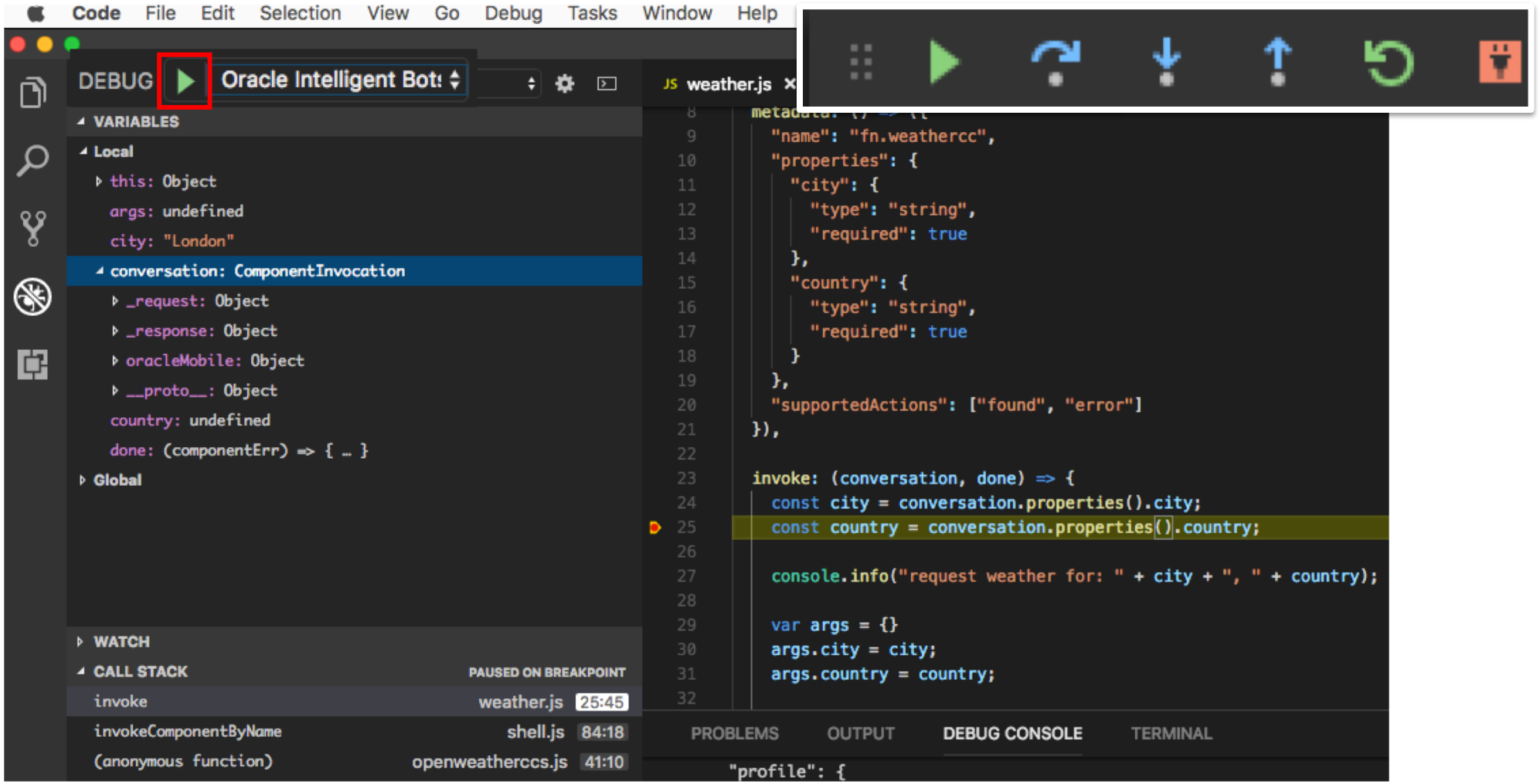


You can **assign a custom name** to the configuration

```
{
"type": "node",
"request": "attach",
"name": "Oracle Intelligent Bots",
"port": 9229
},
```



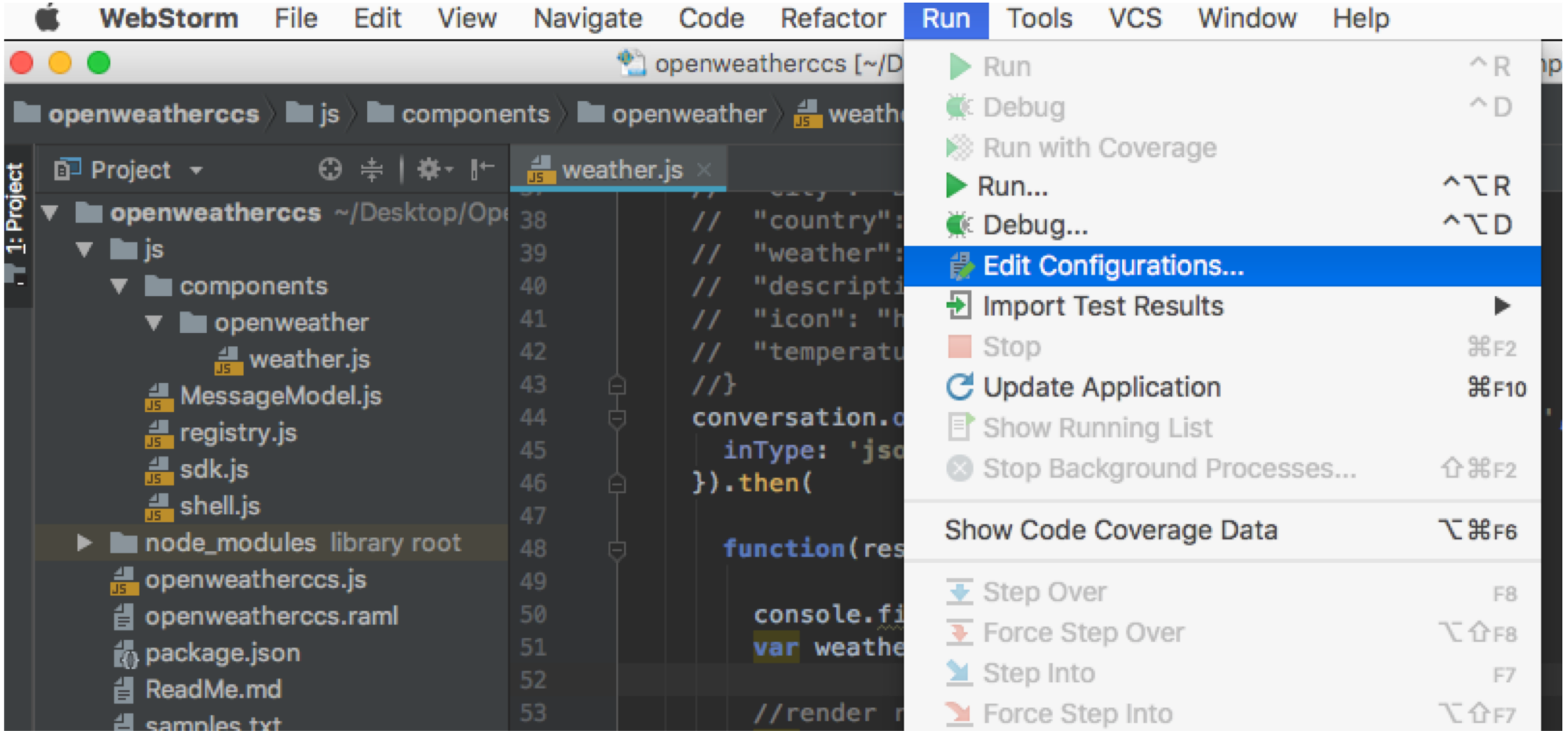
Debugging With Microsoft Visual Studio Code IDE



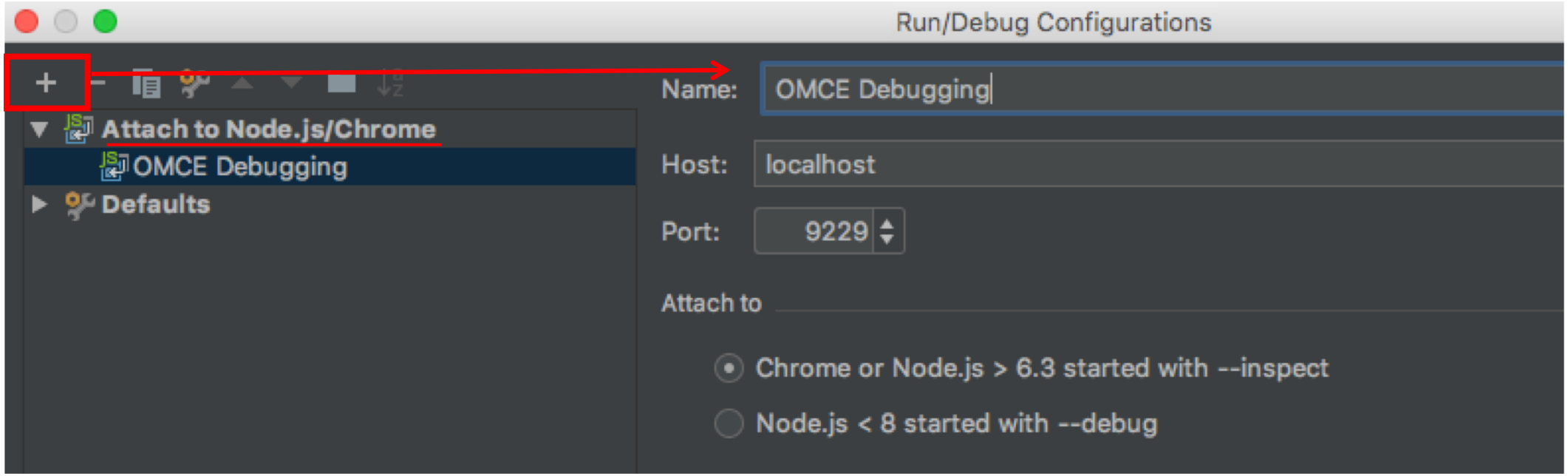
Local Development & Debugging

Debugging with IntelliJ Webstorm

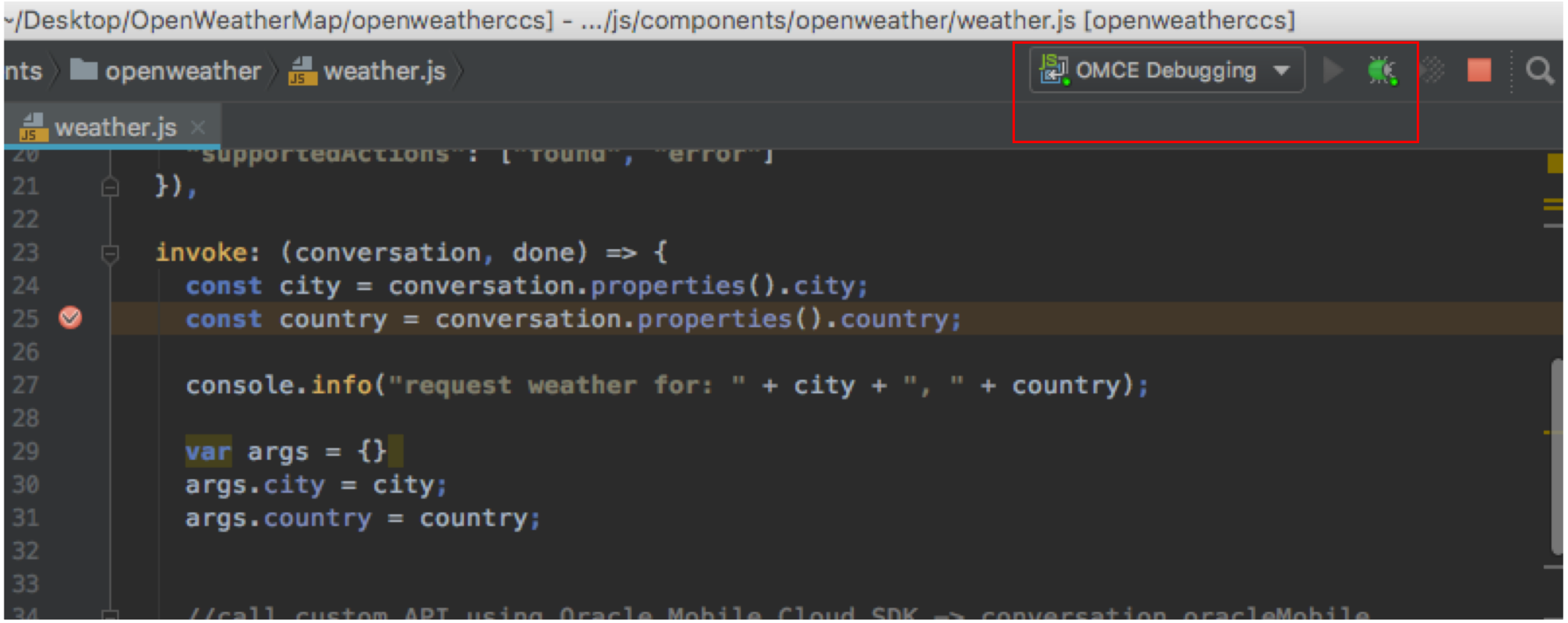
Debugging With IntelliJ Webstorm IDE



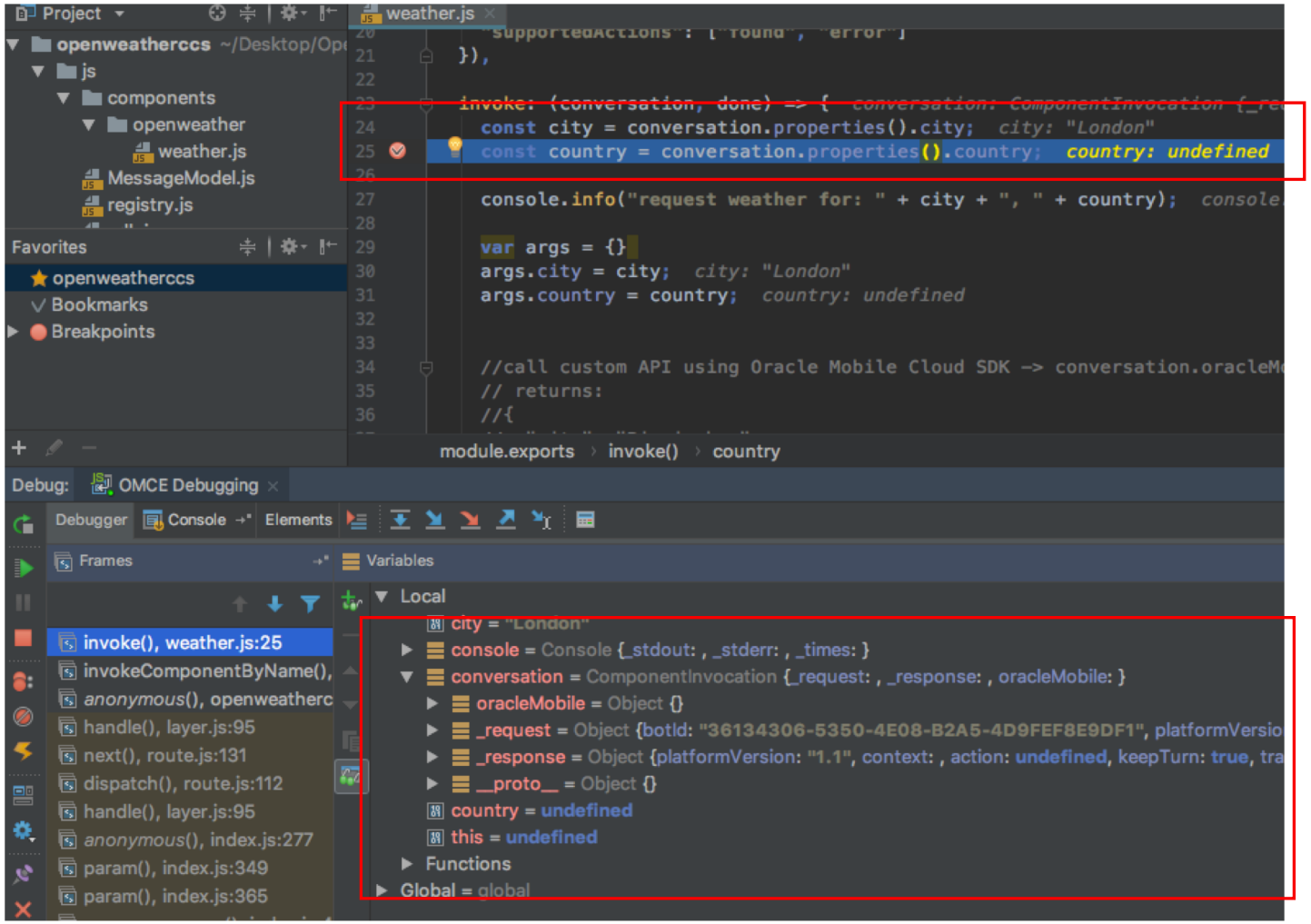
Debugging With IntelliJ Webstorm IDE



Debugging With IntelliJ Webstorm IDE



Debugging With IntelliJ Webstorm IDE



Discussion

Lets assume **a custom component service project hosts 3** custom **components**. **How many toolsConfig.json files do you need?**

And why?

Advanced Bot Training Hands-On

Lab 5a: Setup Local Debugging


ORACLE®