
ORACLE®

Oracle Digital Assistant

The Complete Training

Custom Component Development with Mobile Hub

ORACLE

Image courtesy of pixabay.com

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.

Topic agenda

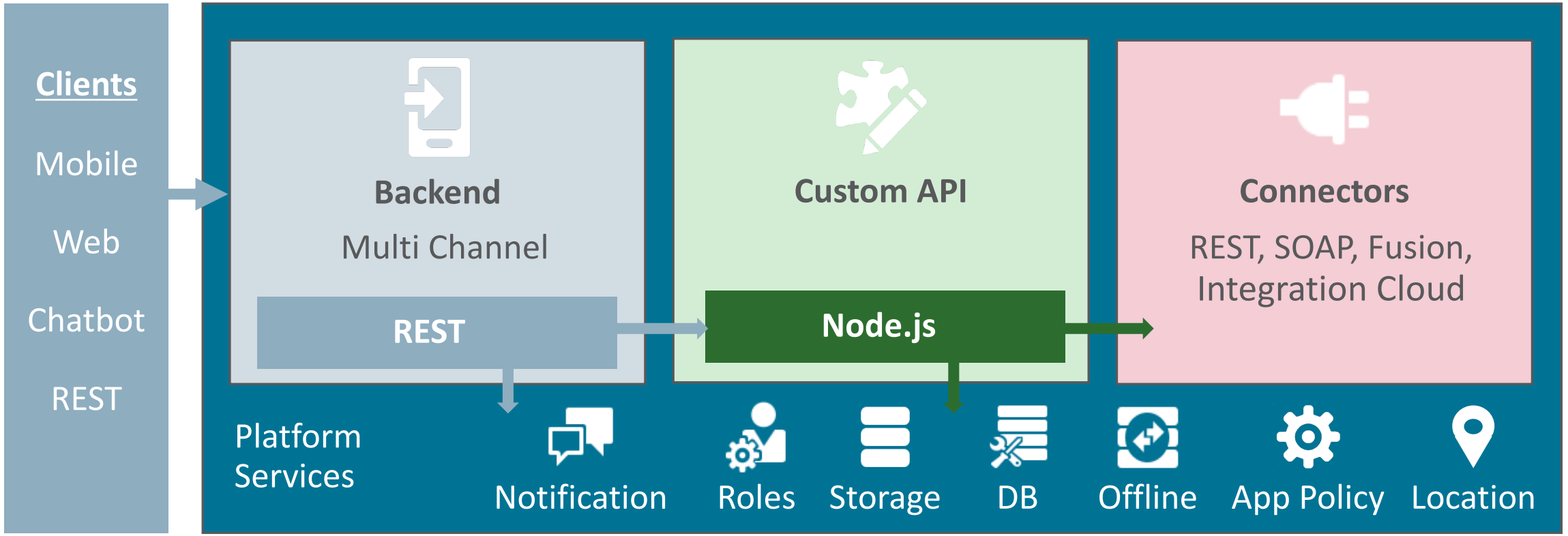
1. Mobile Hub introduction
2. Custom component services in Mobile Hub
3. Building custom components in Mobile Hub
4. Backend integration
5. Local development and debugging

Topic agenda

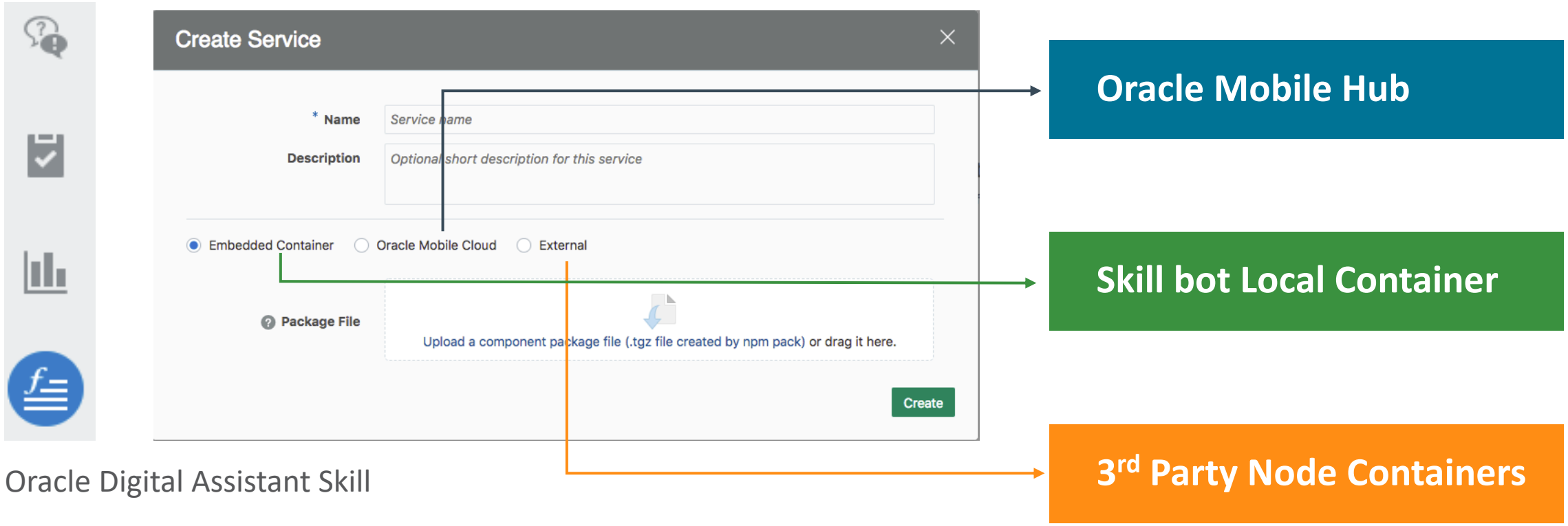
1. Mobile Hub introduction
2. Custom component services in Mobile Hub
3. Building custom components in Mobile Hub
4. Backend integration
5. Local development and debugging

Oracle Mobile Hub

Multi channel backend



Custom component service deployment options



Oracle Digital Assistant Skill

Mobile Hub benefits

- Multi channel backend service
  - API sharing between web, mobile and bot applications
  - Secure API access
  - Payload shaping
  - Platform services: storage, analytics, database, location, push etc.
- API and API Implementation versioning
- Declarative REST, SOAP and Fusion Apps connectors
- Single point of administration and maintenance
- Diagnostics

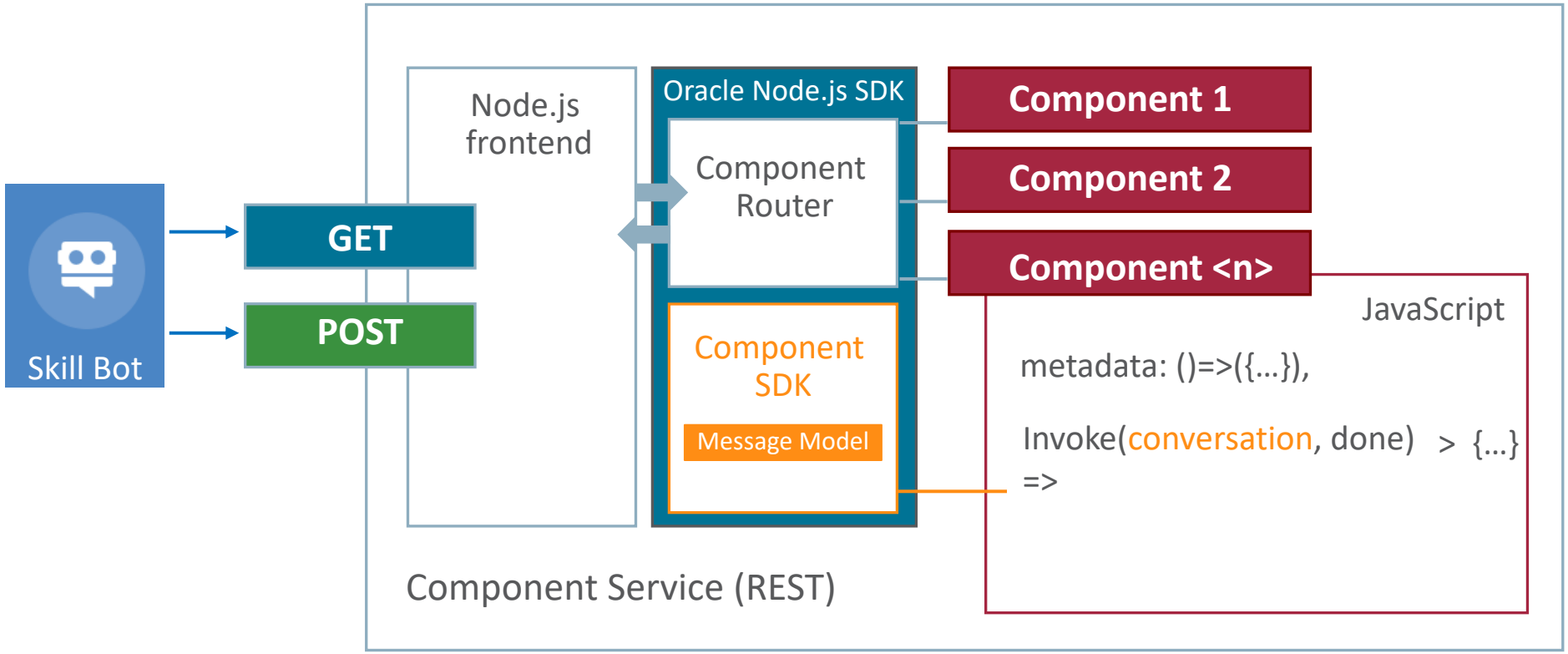
Topic agenda

1. Mobile Hub introduction
2. Custom component services in Mobile Hub
3. Building custom components in Mobile Hub
4. Backend integration
5. Local development and debugging

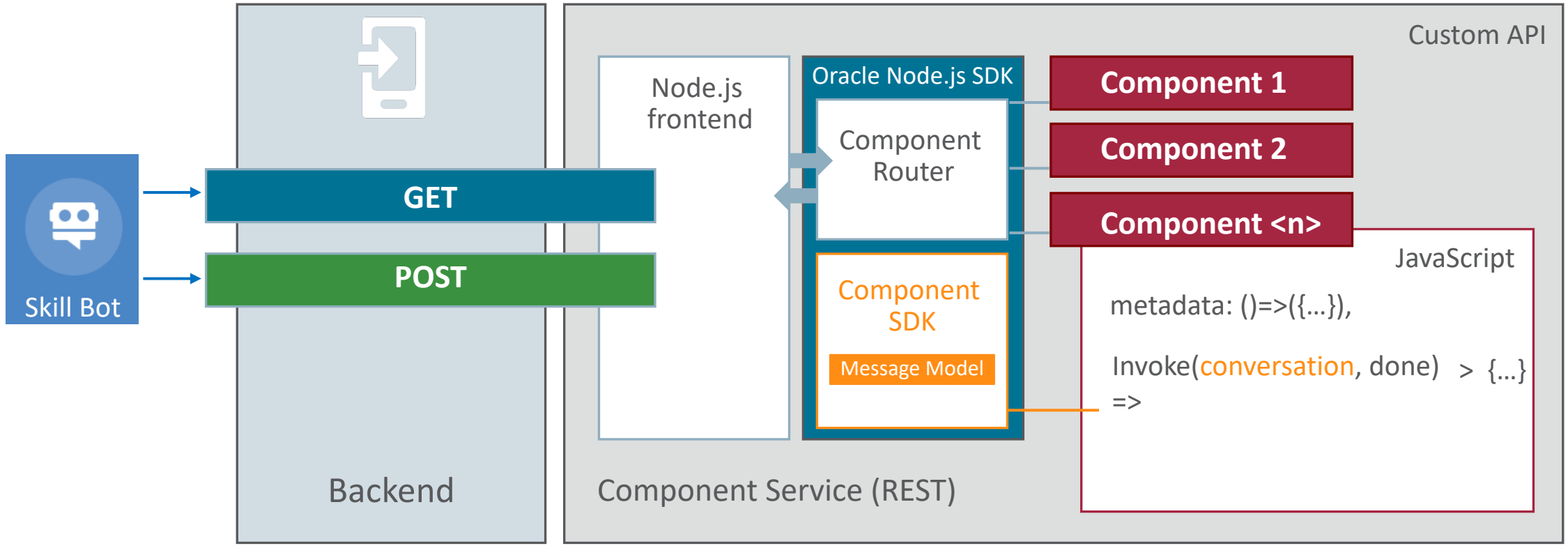
Custom component service development in Mobile Hub

- Custom component service built as Custom API
  - Component Service exposed through a Backend
  - Access to Mobile Hub services and SDK
  - Leverages Mobile Hub connector framework (REST, SOAP, Fusion, ICS)
- Component service API and Implementation versioned in Mobile Hub
  - Node programming
- Logs and diagnostic information saved in backend analytics

Custom component service architecture



Custom component service architecture in Mobile Hub



Topic agenda

1. Mobile Hub introduction
2. Custom component services in Mobile Hub
3. **Building custom components in Mobile Hub**
4. Backend integration
5. Local development and debugging

Installing Oracle Node.js SDK

- Global installation provides the command line to create custom components
- Requires Node and Node Package Manager (NPM) to be installed
- Open a terminal window and type

MAC / Linux

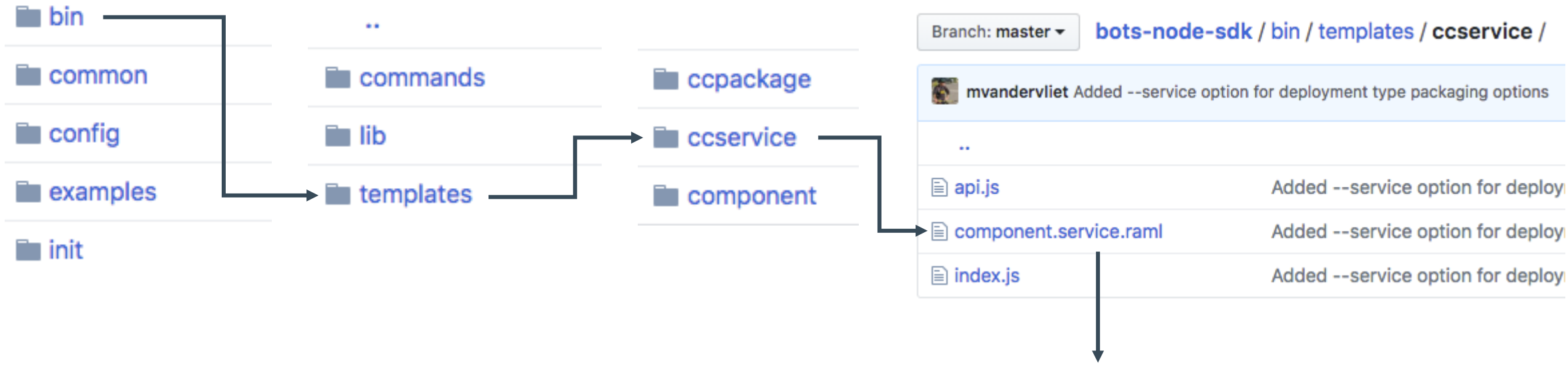
```
sudo npm install –g @oracle/bots-node-sdk
```

Windows

```
npm install –g @oracle/bots-node-sdk
```

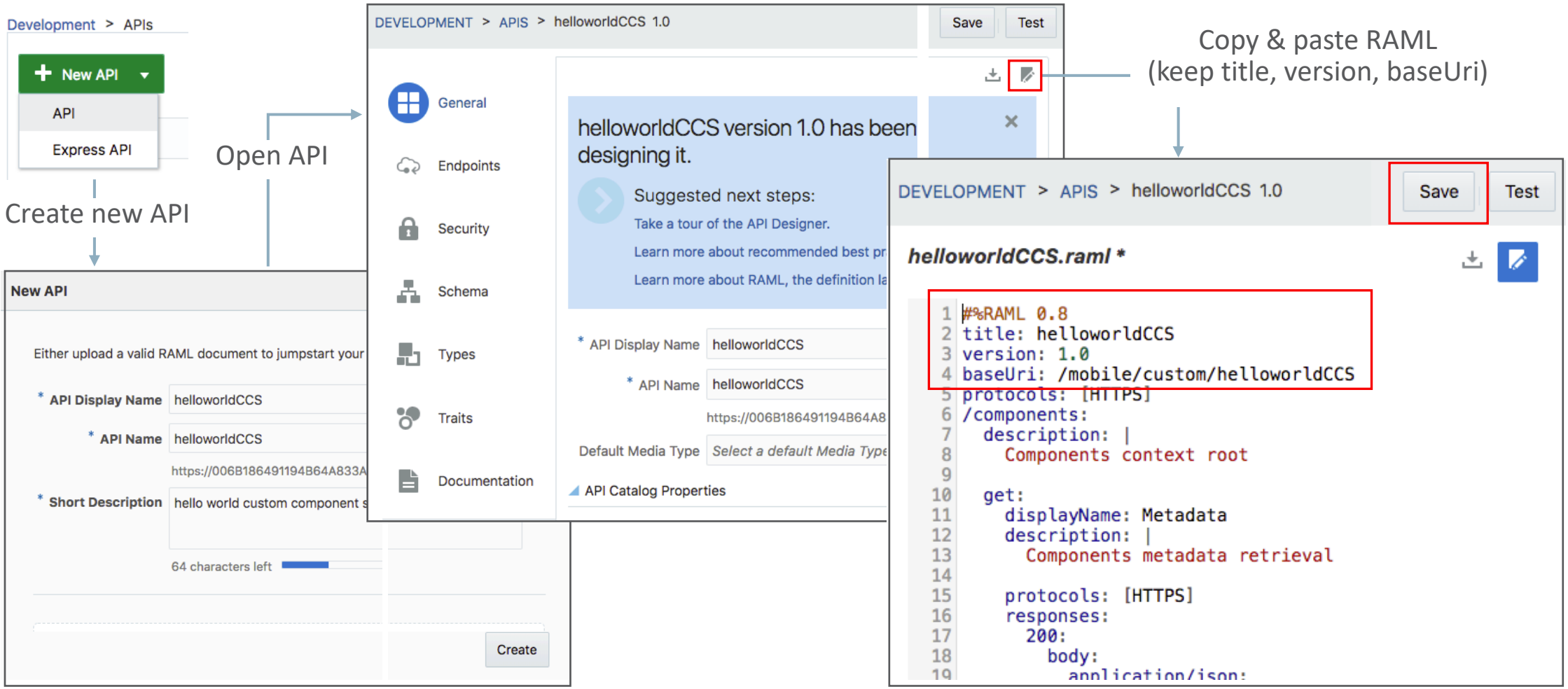
Downloading the custom API template

- Oracle Mobile Hub Custom API starter template is available in Oracle Bots Node.js SDK on GitHub
  - https://github.com/oracle/bots-node-sdk
  - Defines GET and POST methods required for Bot custom component services

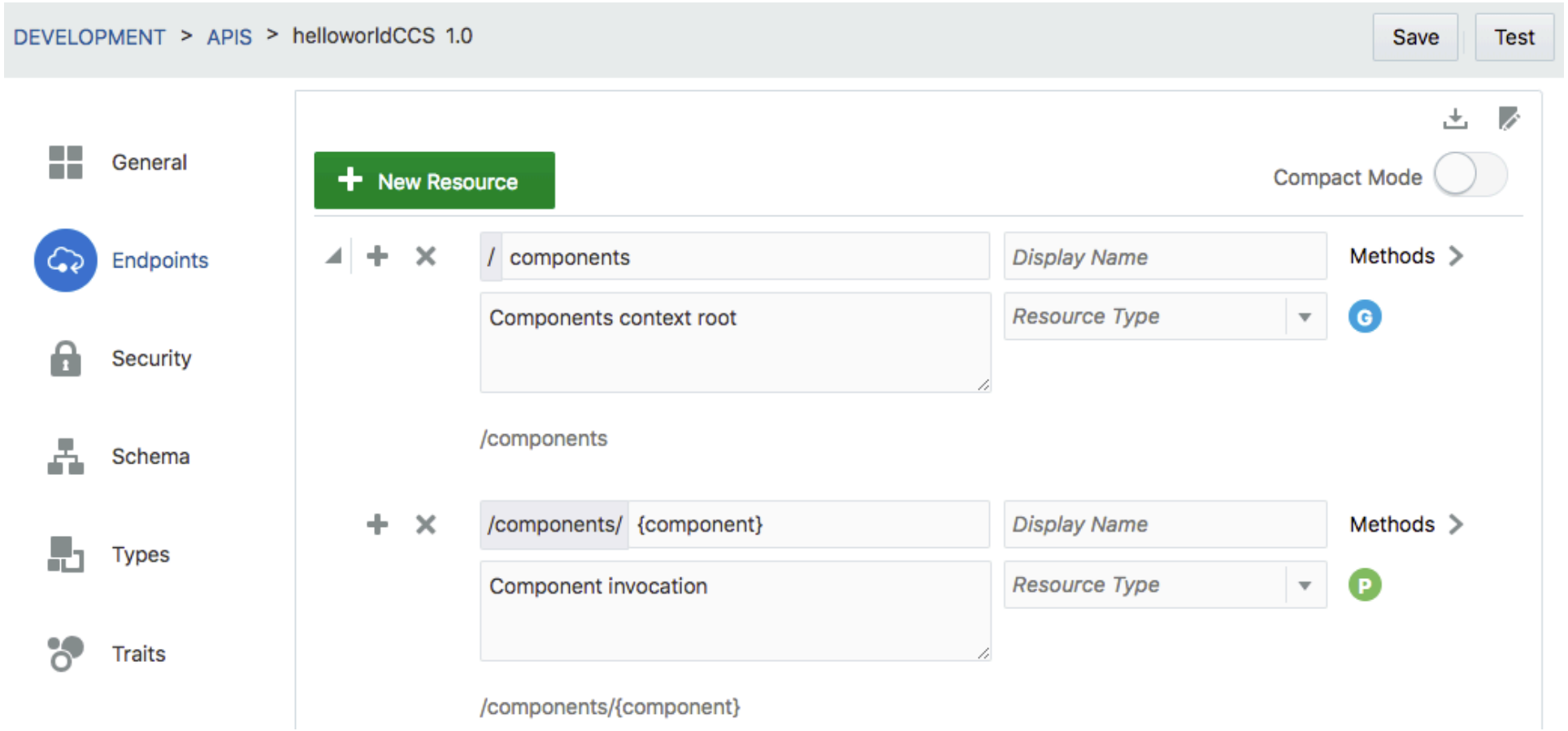


https://github.com/oracle/bots-node-sdk/blob/master/bin/templates/ccservice/component.service.raml

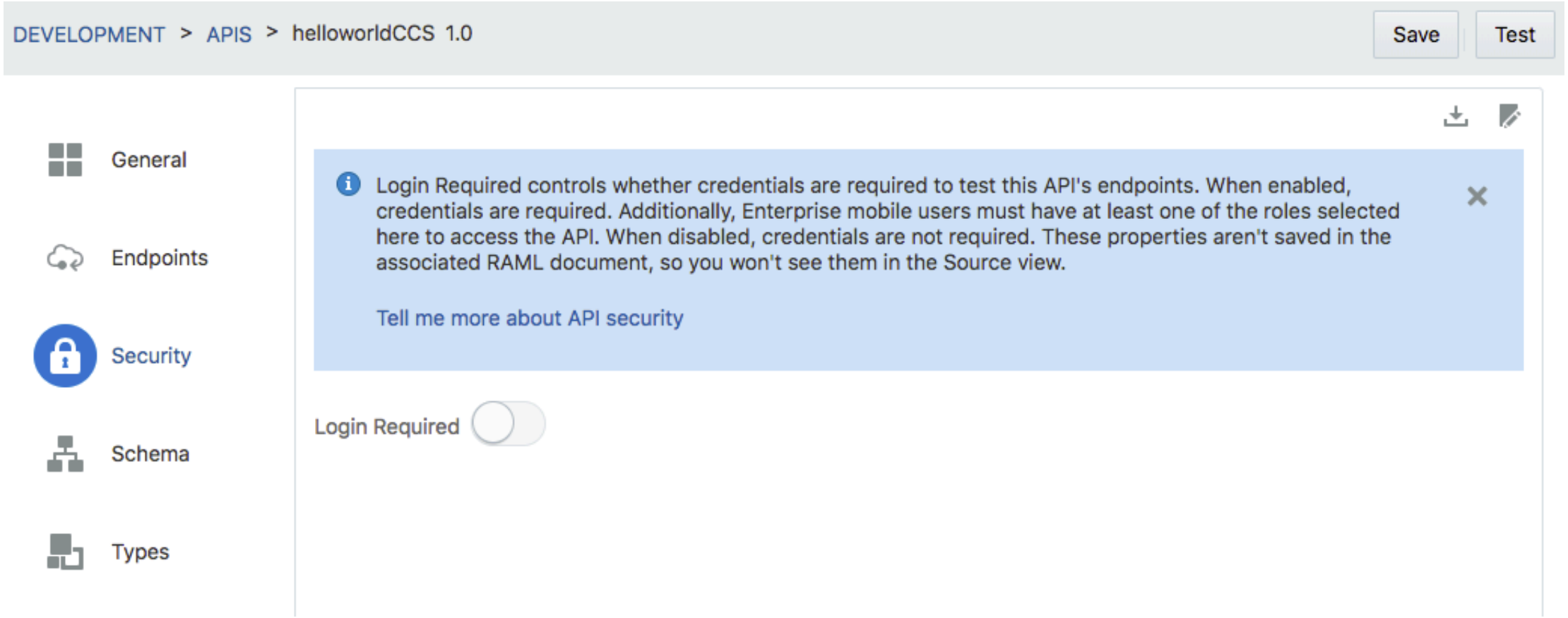
Creating a custom component service API



Custom component service endpoints



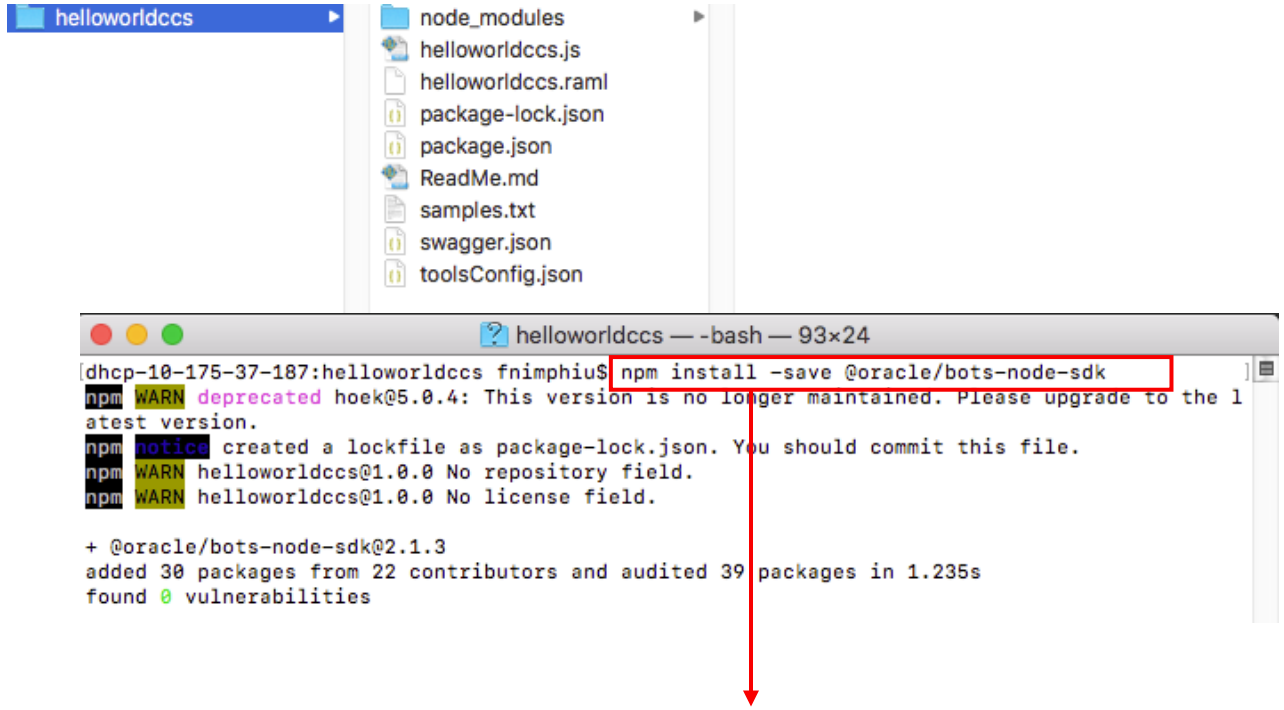
Disable login requirement



Downloading the custom API scaffold

Setting up the local development environment

- Unzip the downloaded scaffold
- Open command line and navigate into custom API root folder
  - Folder that contains package.json
- Install Oracle Bots Node.js SDK locally
  - Provides custom component SDK
  - Handles component request routing




```
npm install @oracle/bots-node-sdk
```

For better code organization**, create a "components" folder** and then a folder for each custom component you build therein

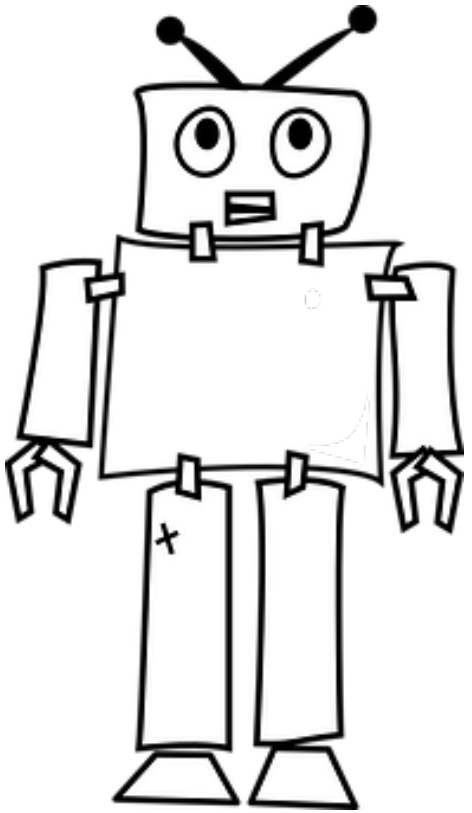
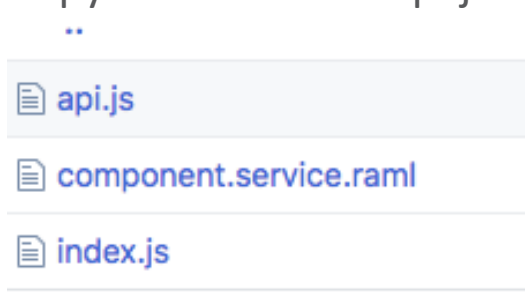


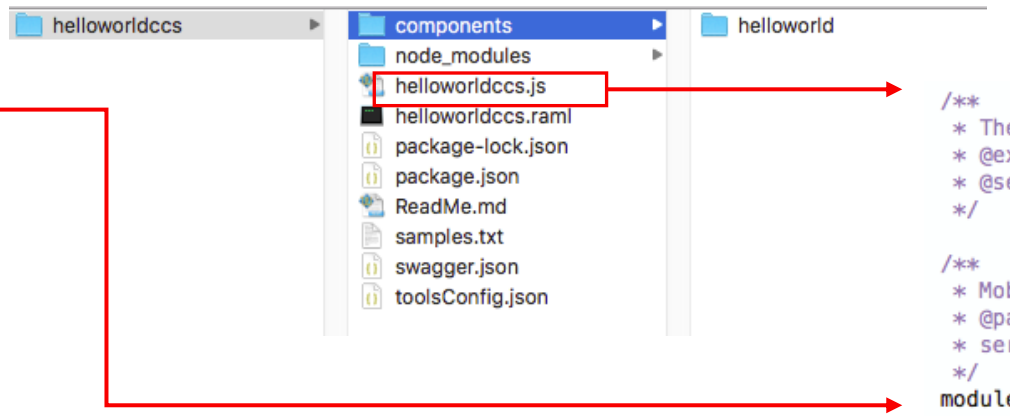
Image courtesy of pixabay.com

Editing the custom API

https://github.com/oracle/bots-node-sdk/blob/master/bin/templates/ccservice/api.js

Copy Content from api.js





Copy Content

```
/**
 * The ExpressJS namespace.
 * @external ExpressApplicationObject
 * @see {@link http://expressjs.com/3x/api.html#app}
 */

/**
 * Mobile Cloud custom code service entry point.
 * @param {external:ExpressApplicationObject}
 * service
 */
module.exports = function (service) {

  const OracleBot = require('@oracle/bots-node-sdk');
  OracleBot.init(service);

  // implement custom component api
  OracleBot.Middleware.customComponent(service, {
    baseUrl: '/mobile/custom/helloworldCCS/components',
    cwd: __dirname,
    register: [
      './components'
    ]
  });

};
```

Edit baseUrl, cwd, register properties

Custom component service code explained

```
module.exports = function (service) {

  const OracleBot = require('@oracle/bots-node-sdk');
  OracleBot.init(service);

  // implement custom component api
  OracleBot.Middleware.customComponent(service, {
    baseUrl: '/mobile/custom/helloworldCCS/components',
    cwd: __dirname,
    register: [
      '. /components'
    ]
  });

};
```

Node module definition

Load command for Bot Node.js SDK

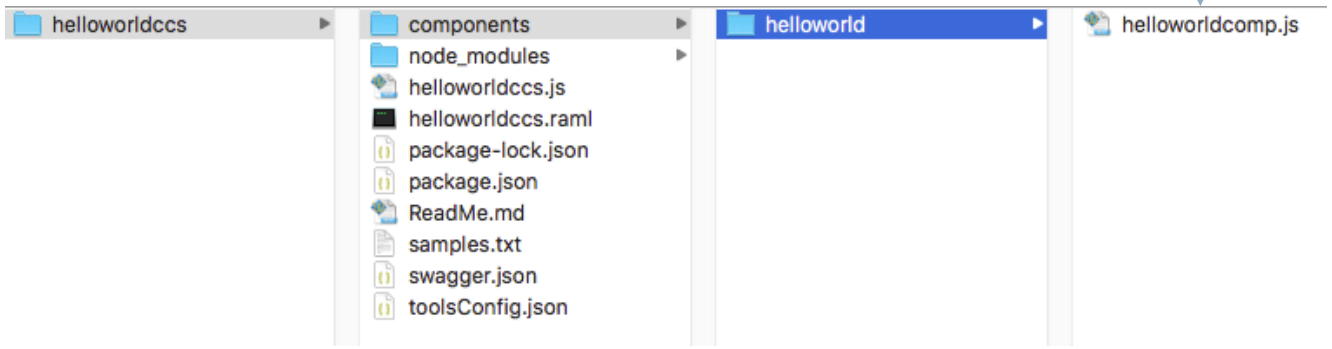
Request to custom component router

REST URI to invoke custom component (should match your API URI)

Relative folder custom components are Searched in

Creating a custom component

```
fnimphiu-orcl:helloworldccs fnimphiu$ bots-node-sdk init component --name helloworldcomp components/helloworld
Writing file: /Users/fnimphiu/Downloads/helloworldccs/components/helloworld/helloworldcomp.js
------------------------------------------------------------
Added Custom Component: 'helloworldcomp'
------------------------------------------------------------
fnimphiu-orcl:helloworldccs fnimphiu$
```




```
bots-node-sdk init component --name helloworldcomp components/helloworld
```

Generated custom component file & code

helloworldcomp.js

```
'use strict';

module.exports = {
  metadata: () => ({
    name: 'helloworldcomp',
    properties: {
      human: { required: true, type: 'string' },
    },
    supportedActions: ['weekday', 'weekend']
  }),
  invoke: (conversation, done) => {
    // perform conversation tasks.
    const { human } = conversation.properties();
    // determine date
    const now = new Date();
    const dayOfWeek = now.toLocaleDateString('en-US', { weekday: 'long' });
    const isWeekend = [0, 6].indexOf(now.getDay()) > -1;
    // reply
    conversation
      .reply(`Greetings ${human}`)
      .reply(`Today is ${now.toLocaleDateString()}, a ${dayOfWeek}`)
      .transition(isWeekend ? 'weekend' : 'weekday');

    done();
  }
};
```

Component invocation name

Component properties

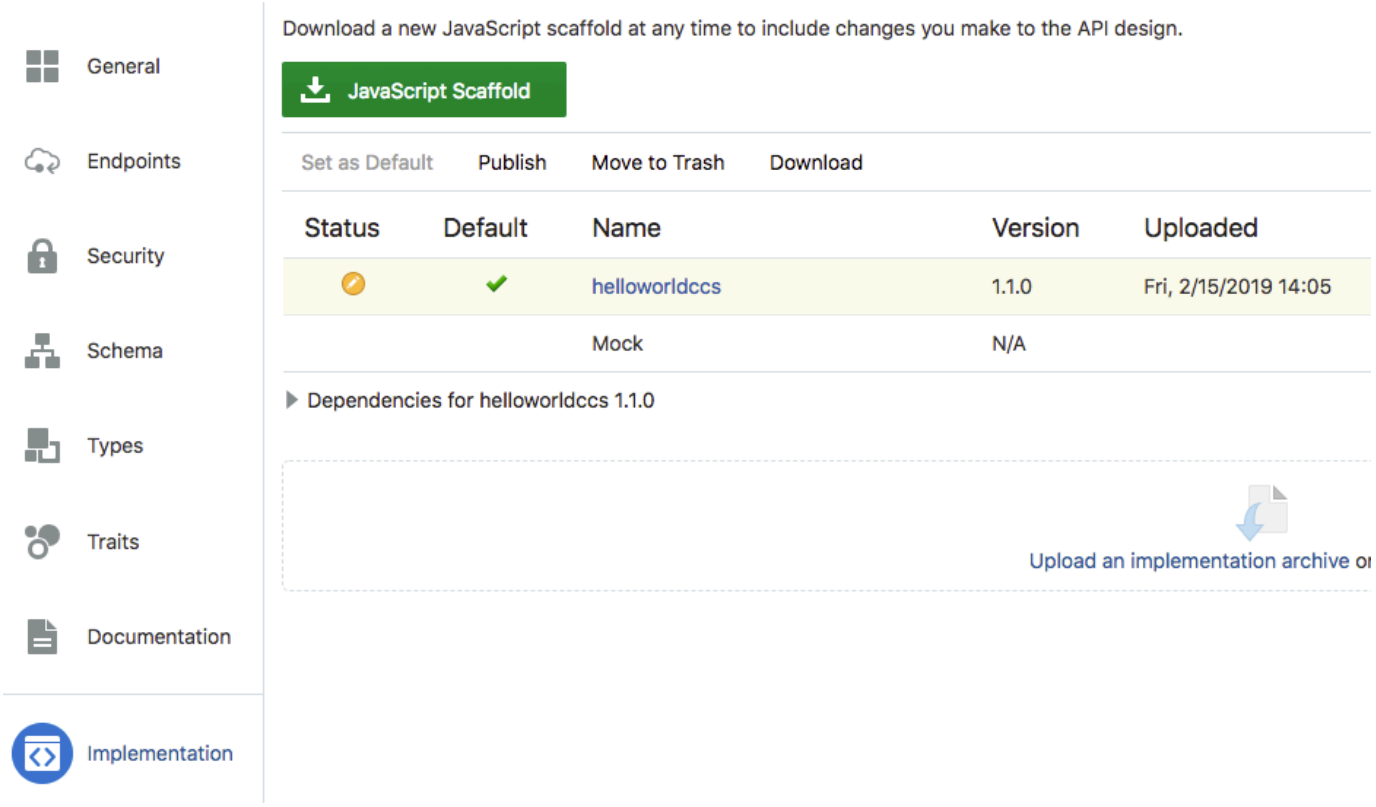
Action transitions

Function invoked at runtime

Callback that must be called at the end

Deploying the custom component service to Mobile Hub

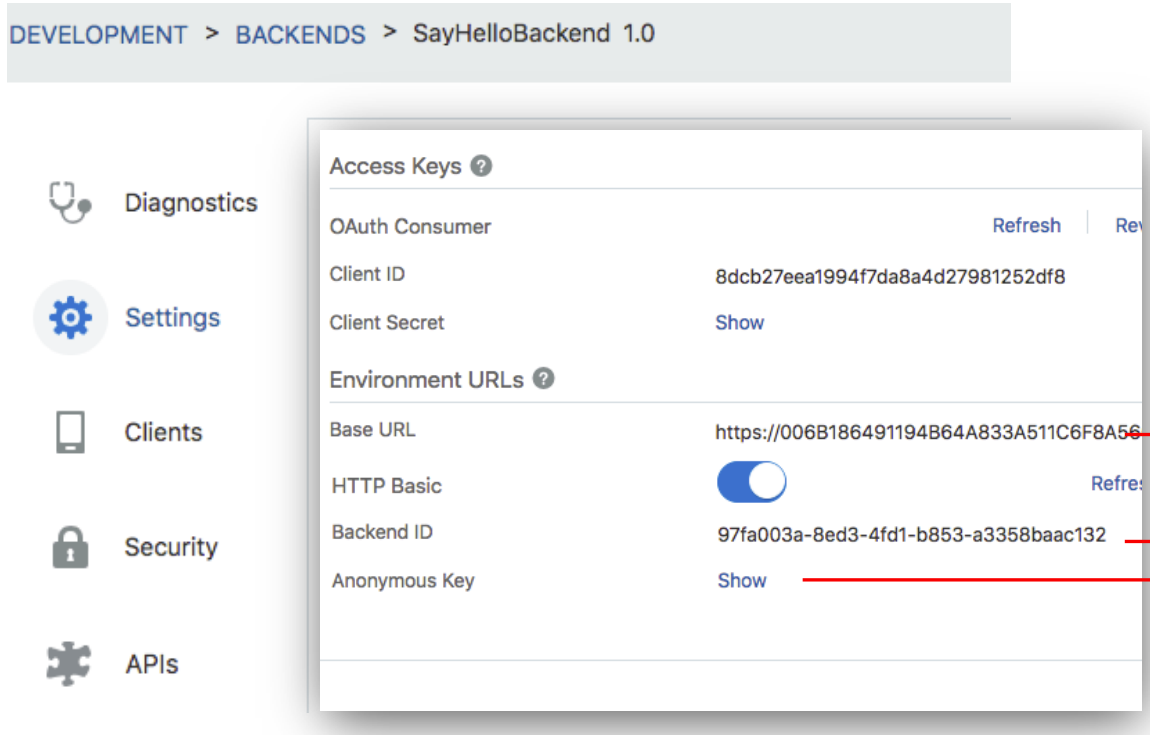
- compress project root folder to a *zip*-file
- Upload *zip-file* as custom API implementation
- Use embedded tester in Oracle Mobile Hub to test GET method
- Expose custom API on Mobile Hub backend



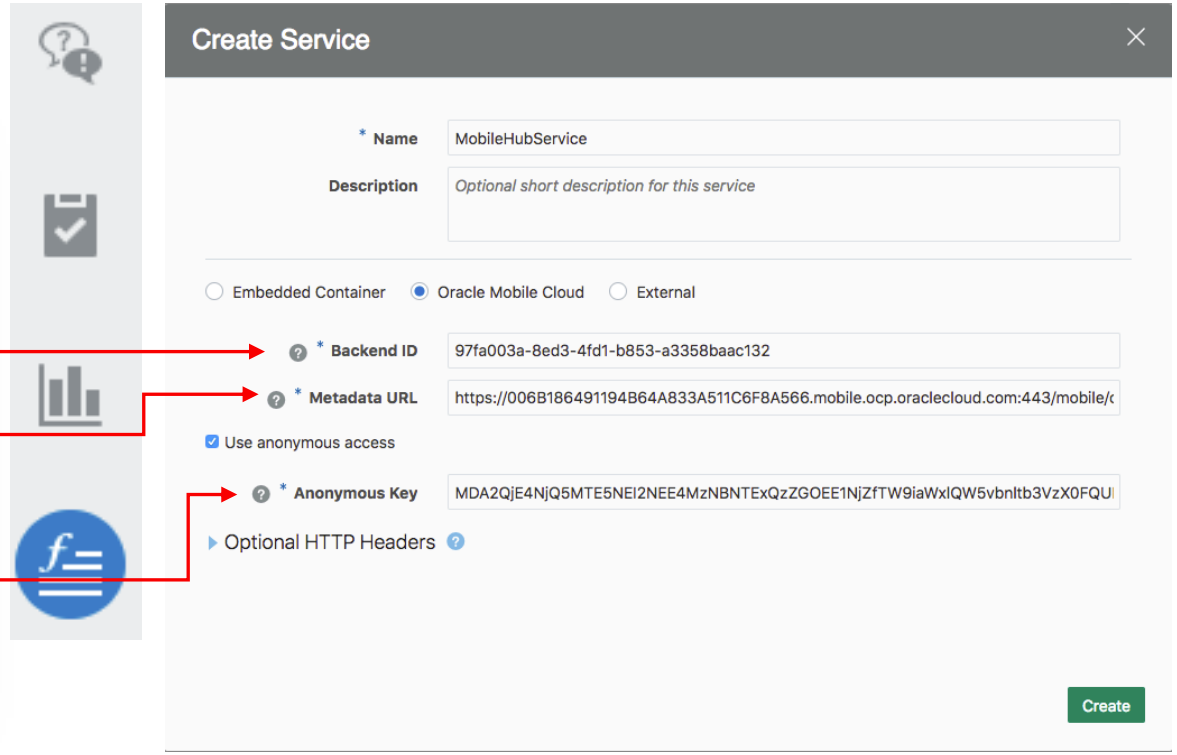
'Deployment' to Oracle Digital Assistant

Component registration in Oracle Digital Assistant skill

Oracle Mobile Hub



Skill



Topic Agenda

1. Mobile Hub introduction
2. Custom component services in Mobile Hub
3. Building custom components in Mobile Hub
4. **Backend integration**
5. Local development and debugging

In addition to multi-channel support, **backend integration is a good argument for using Oracle Mobile Hub** to create custom components in

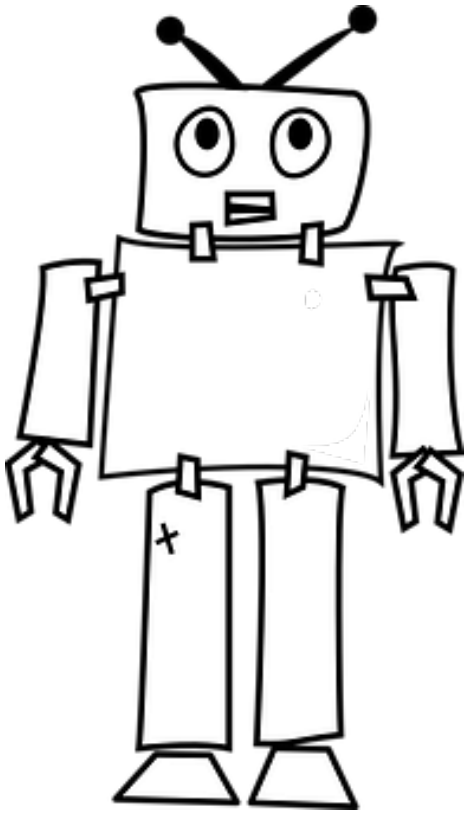
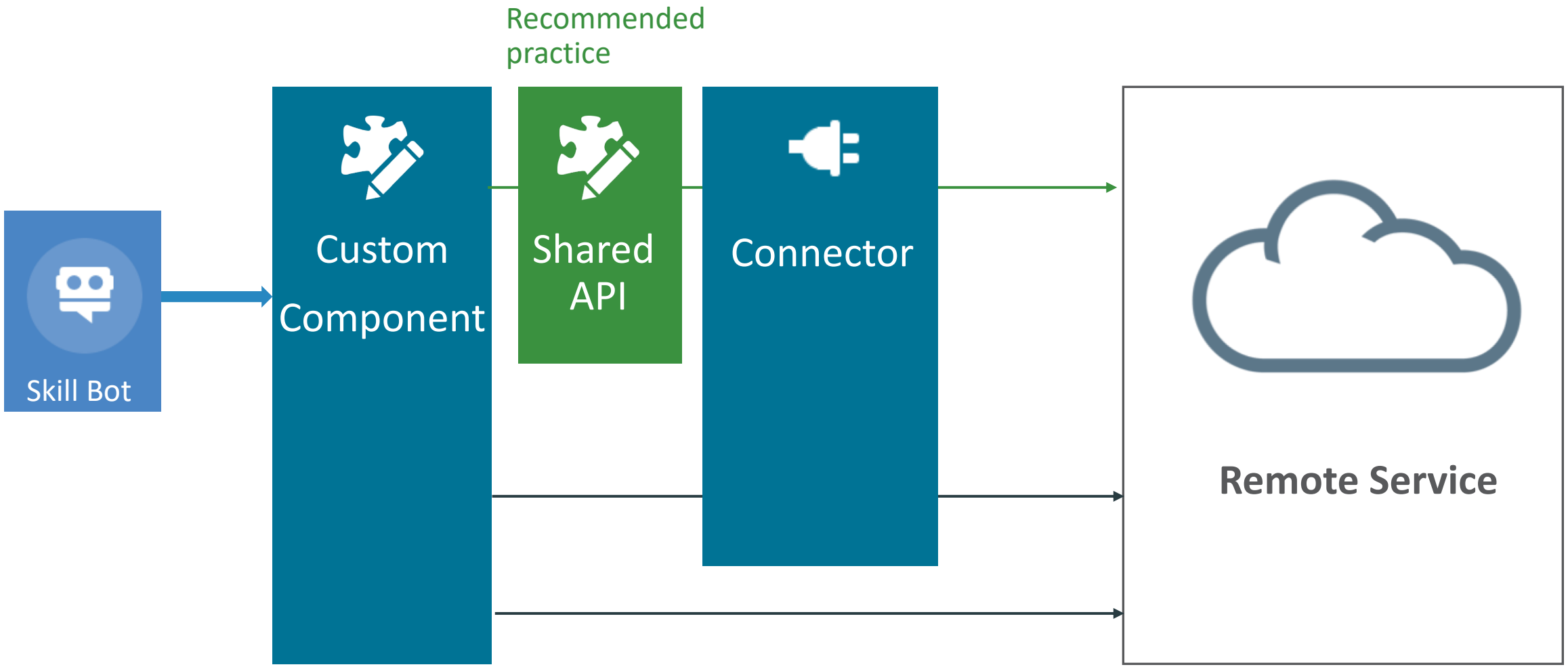


Image courtesy of Farbentek at FreeDigitalPhotos.net

Mobile Hub backend integration options



You can **access** the **Oracle Mobile Hub SDK through** the **Custom Component SDK** to access Mobile Hub custom APIs, platform APIs and Connectors

```
conversation.oracleMobile.<function>
```

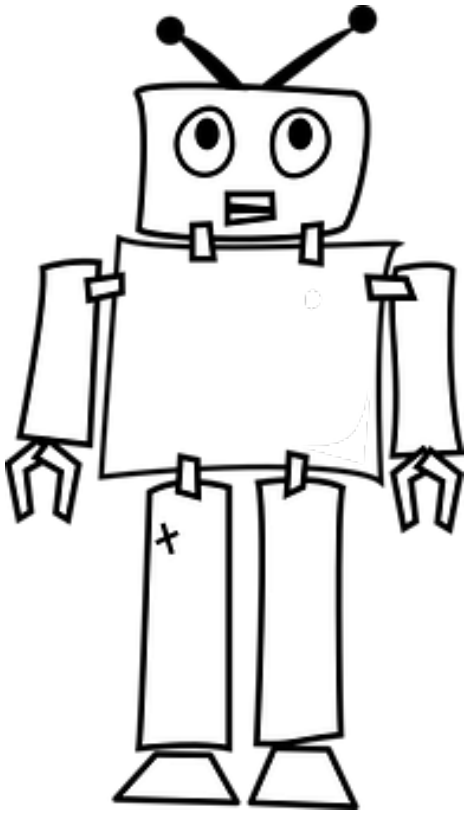


Image courtesy of Farbentek at FreeDigitalPhotos.net

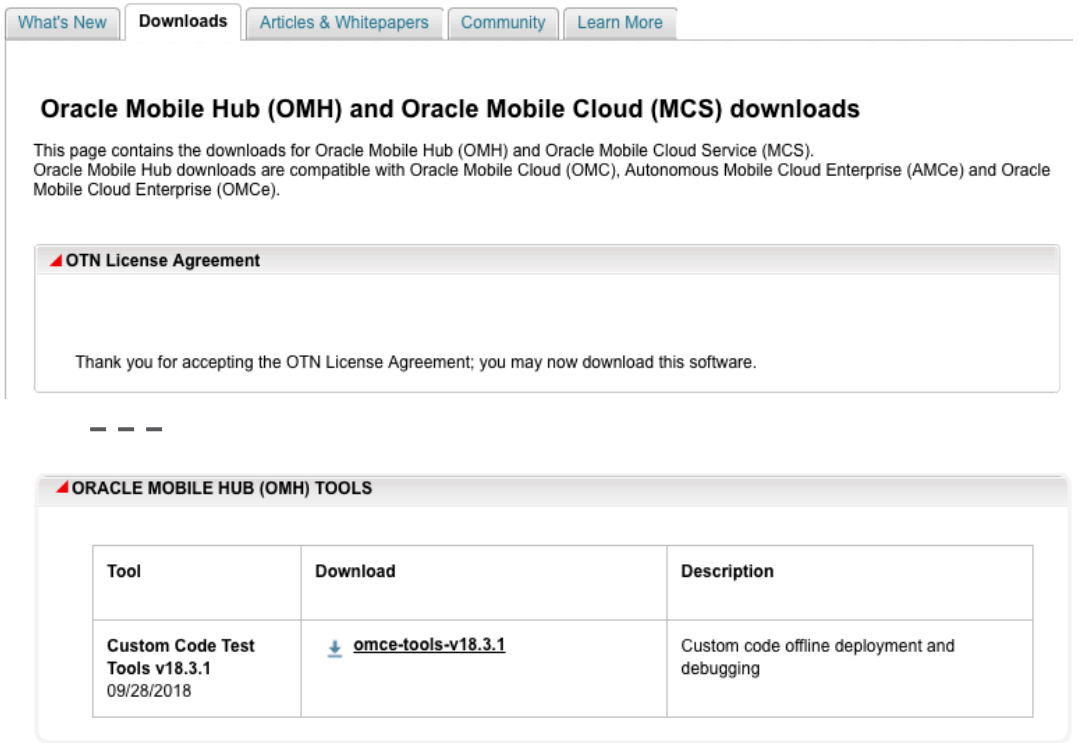
Topic agenda

1. Mobile hub introduction
2. Custom component services in Mobile Hub
3. Building custom components in Mobile Hub
4. Backend integration
5. Local development and debugging

Custom component debugging with Oracle Mobile Hub

- Install Oracle Custom Code Test Tool
  - Download from OTN
  - Follow instructions in readme
- Configure backend with code test tool reference
  - Code test tool proxy installed as custom API
- Have local copy of component service custom API
  - Configure toolsConfig.json with backend access
- Start Local Debugging
  - Code test tool command

https://www.oracle.com/technetwork/topics/cloud/downloads/mobile-cloud-service-3636470.html



Mobile Hub debugging architecture

Debug with JS Editor like
MS Visual Studio Code

omce-tools download
from OTN

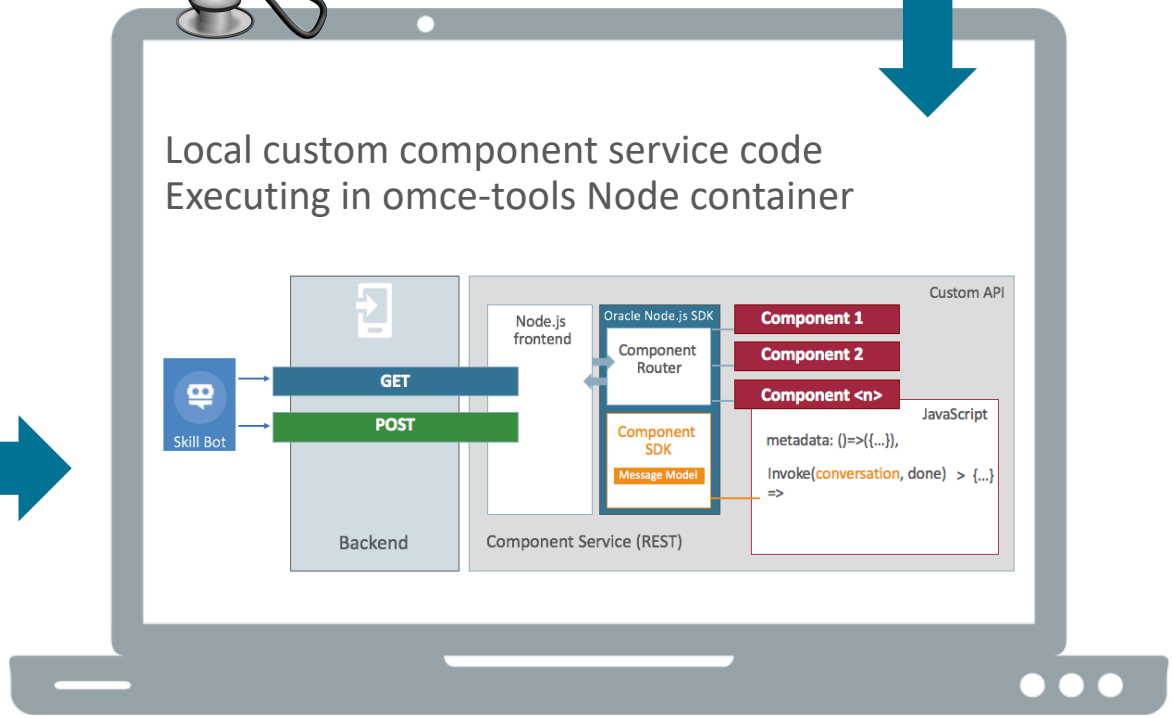
Internet

Access Dependent
Services

Oracle Mobile Hub

Oracle Digital Assistant
(Skillbot)

Tunnel
(e.g. Ngrok)



Integrated Cloud

Applications & Platform Services


ORACLE®

Oracle Digital Assistant Hands-On

TBD