

Hackathon Lite

Setup and Development Guide



group team public mobile execute event features



Goals For This Session

By the end of this session, Hackathon participants should be able to...

1. Setup their workstation for Hackathon development
2. Design, develop, and deploy API's and mock services
3. Call your mock services in a live web service environment

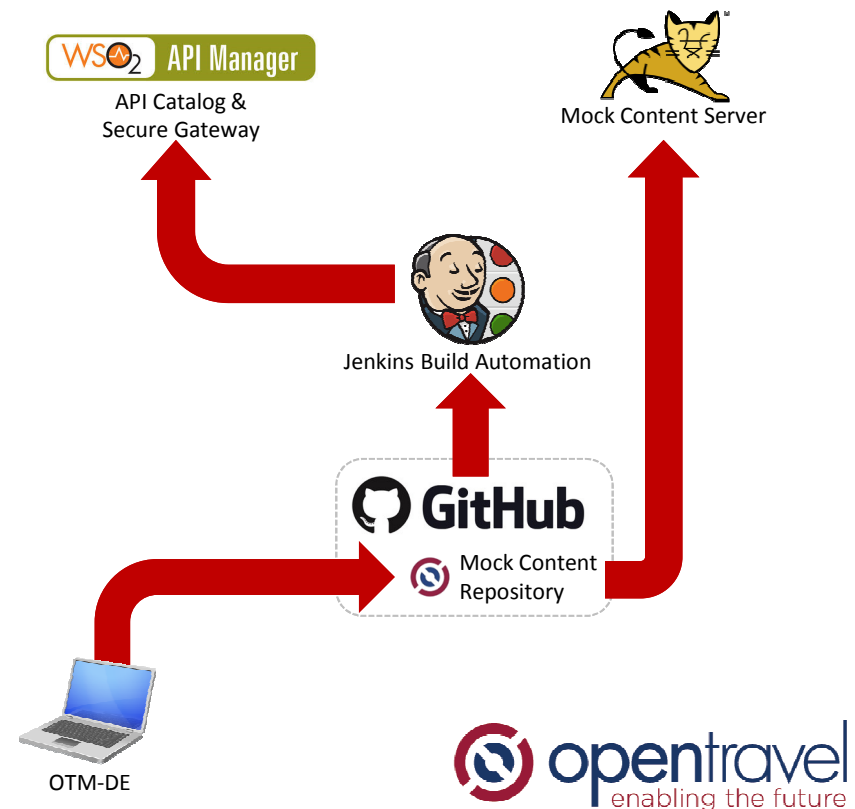


Hackathon Lite Environment

DevOps environment that leverage build automation to quickly design, deploy, and publish RESTful mock services

- OTM-DE – Designer application for modeling information and API's using OTA 2.0 technology
- GitHub Repository – Maintains OTM projects and mock content for all participants
- Jenkins – Coordinates publication of OTM resources to the API Catalog & Secure Gateway
- Mock Content Server – Tomcat web service to returns mock responses in response to live API calls
- WSO2 API Manager – Open source API Catalog and Secure Service Gateway

All of these technologies are free and publicly available to everyone who wants to use them!



Workspace Setup

System Requirements (Java 7 or later, Git, OTM-DE & Postman [recommended])

- Java 7 (or later) SE
 - Provides a Java run-time for the OTM applications
 - Available for free download (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
- Git Client
 - Source control and remote access to GitHub repositories
 - Available for free download (<https://git-scm.com/downloads>)
- OTM-DE Application
 - Creation and editing of OTM models
 - Latest Stable Build available on the ODN (<http://opentraveldevelopersnetwork.com/model-designer-app>)
- OTM Example Helper
 - Provides fine-grained support for substitution group selection in XML and JSON examples
 - Latest Stable Build available on the ODN (<http://opentraveldevelopersnetwork.com/model-designer-app>)
- Postman
 - Recommended for testing of REST services (other alternatives available)
 - Available for Free Download (<http://www.getpostman.com/>)

Workspace Setup (continued)

1. Get Access to the GitHub Repository

- All Hackathon participants will require a GitHub login (visit <https://github.com> to sign up)
- Contact Stephen.Livezey@travelport.com for access to the Hackathon workspace repository

2. Cloning the Mock Content Repository

- Create a local clone (copy) of the repository by running the following from a command window:

```
> git clone https://github.com/OpenTravel-Forum-2016/otaforum-mock-content.git
```

3. Create a context work area

- Multiple teams will be sharing the same GitHub repository. Each team will create a separate context folder to hold their own models and mock responses
- Simple create a new directory under the root folder of your local copy of the GitHub repository (no spaces in the directory name)

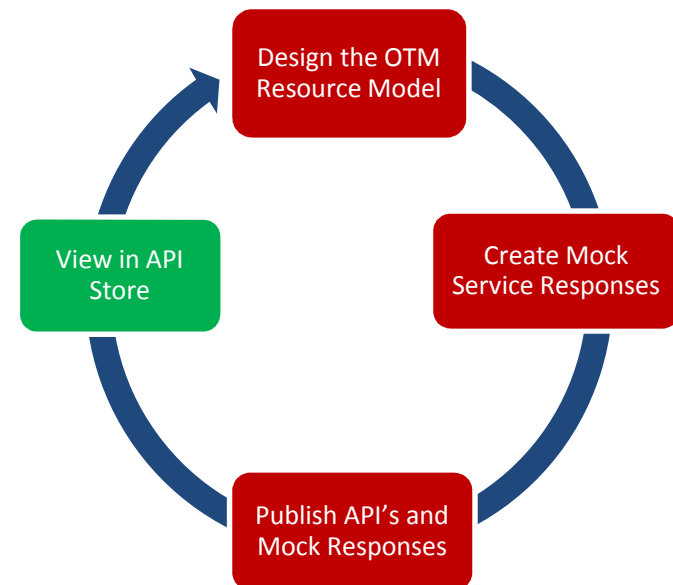
4. Creating an API Store Login

- Register at the WSO2 API Store (<https://opentravelforum.net:9443/store>)
- Click the “Sign Up” link and fill out the required information

Developing Mock Services

Mock service development requires only a few simple steps:

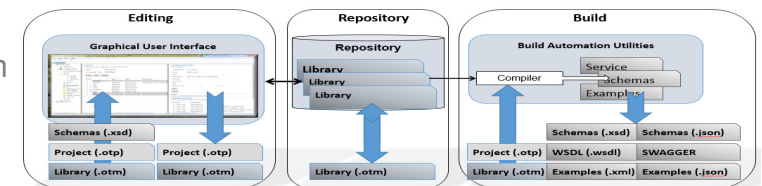
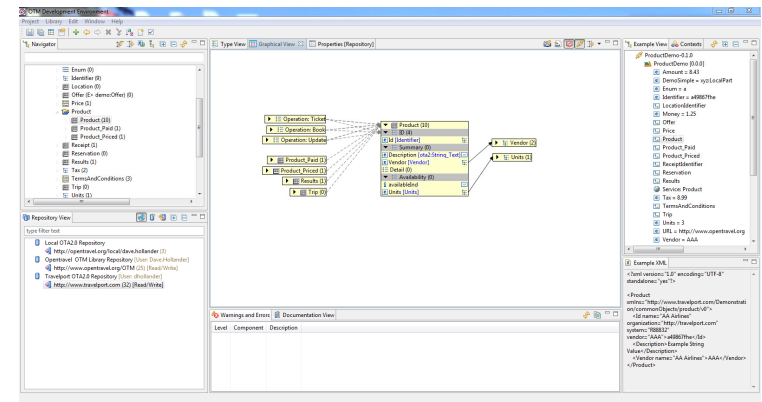
- *Design the OTM Resource Model* using the OTM-DE application
- *Create Mock Service Responses*
 - Run Update Context to generate mock folders and sample responses in your local workspace
 - Edit / Add Mock Responses to refine the content and structure of your mock responses
- *Publish API's & Mock Responses*
 - Deliver Updates to GitHub to make them available in the API Catalog and the Mock Content Server
 - Rebase Work Area from GitHub when you want to pick up changes from other participants
- View in API Store within minutes of delivering your work to the GitHub Repository



Design an OTM Resource Model

RESTful API's are designed using OTM Resources...

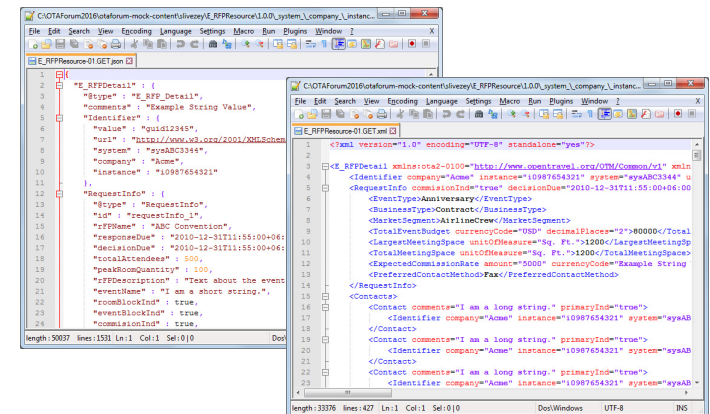
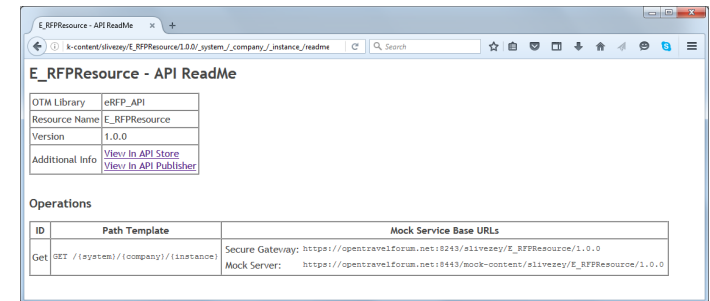
- Business Objects – Model objects to be identified as RESTful entities
- Resources – The API's or actions that can be performed on those entities
- Models can be created from scratch or build upon existing models in the OpenTravel OTM repository
- A separate webinar for the Hackathon Lite event will cover more detail about designing OTM models and RESTful API's
- Special Instructions
 - Save your project (.otp) files somewhere in your context folder
 - New OTM models should be saved in the context folder, rather than adding them into the repository as managed libraries
 - To avoid naming conflicts in the API Store, each team should add a unique prefix to the name of each resource they define



Create Mock Service Responses

Step 1: Run the `UpdateContext.bat` script

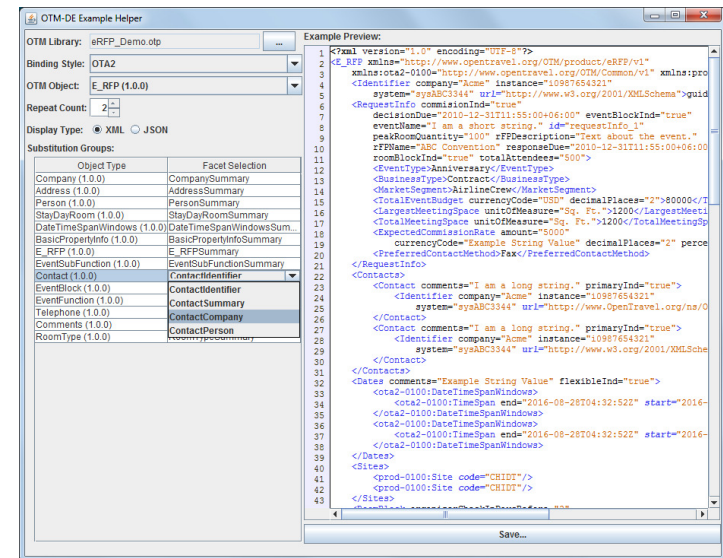
- Located in the root folder of the mock content repository
- A popup window will allow the user to select their context directory
- Several artifacts will be generated automatically in your context folder:
 - New directories that map to the API path templates
 - Sample response files in XML and/or JSON formats
 - ReadMe.html files with basic information and links about each API operation
- Special Note:
 - Folder names that start and end with underscores (“_”) represent wildcards in an API’s path template



Create Mock Service Responses (continued)

Step 2: Edit / Add Mock Response Files

- The `UpdateContext.bat` script creates sample response files, but these may not be sufficient to demonstrate a mock API
- The OTM Example helper allows users to easily define the structure of a message that contains one or more substitution groups
- Once a response message is well-formed, developers can easily hand-edit the mock response details as desired
- Special Notes:
 - The naming convention for response files is:
filename.<HTTP Method>.<Format> (e.g. MyMock-01.POST.json)
 - Existing files will not be changed or overwritten when the script runs
 - If multiple files are provided for the same API path, method, and file format, the mock server will round-robin through those responses
 - Parameter substitution will be performed for any path/query parameters defined in the response body
Example: `<id>{param1}</id>` will become `<id>foo</id>` where param1="foo"



Publish API's and Mock Responses

Publishing mock API's from your local workspace is easy...

- Simply run the `Deliver.bat` script located in the root folder of the mock content repository
 - Publishes mock responses to the live web service
 - Publishes API's to the WSO2 API Store

To download recent changes from other participants:

- Run the `Rebase.bat` script located in the root folder of the mock content repository

For the Git experts...

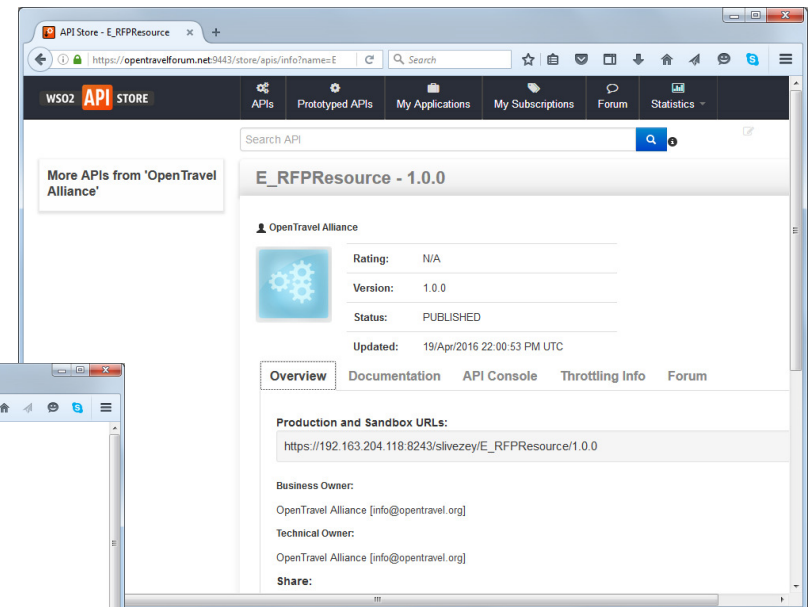
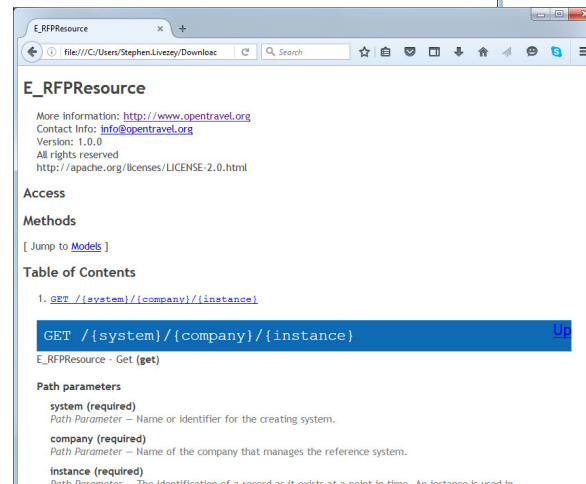
- `Deliver.bat` is equivalent to:
 - > `git stash`
 - > `git pull origin`
 - > `git stash pop`
 - > `git push origin`
- `Rebase.bat` is equivalent to:
 - > `git stash`
 - > `git pull origin`
 - > `git stash pop`



View in the API Store

Within minutes of delivering your updates to GitHub, your API's will be available in the WSO2 API Store!

- API's are based upon Swagger definitions generated by the OTM-DE compiler
- HTML reference is published as part of the API documentation
- XML & JSON schemas also published as documentation



Calling Mock Services

Invocation of Hackathon mock services can be accomplished by two methods:

- Calling via the Secure Gateway
 - HTTPS connections with OAuth2 authentication protocol
 - Closely simulates a production-like environment for consuming applications
- Calling the Mock Content Server directly (optional)
 - HTTPS connections directly to the mock content server
 - Sufficient to demonstrate the function and use of an API

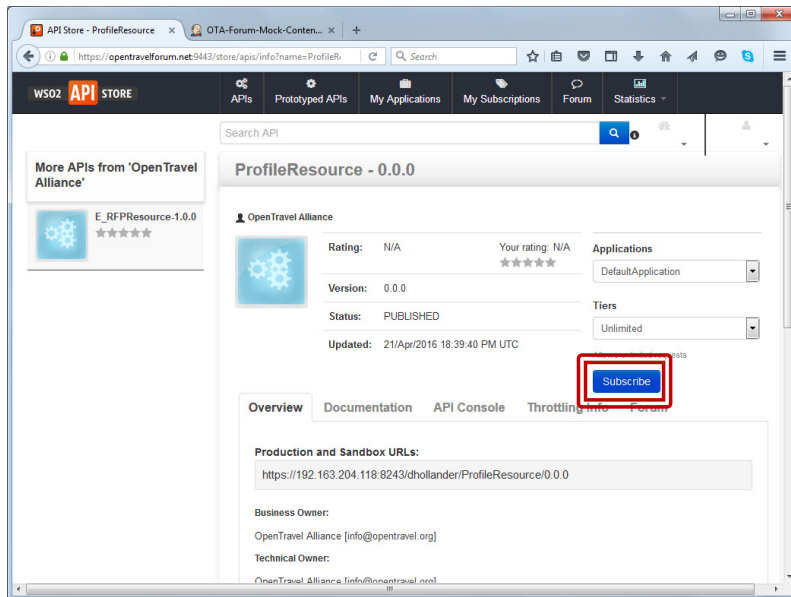


Calling via the Secure Gateway

- Subscribing to the API

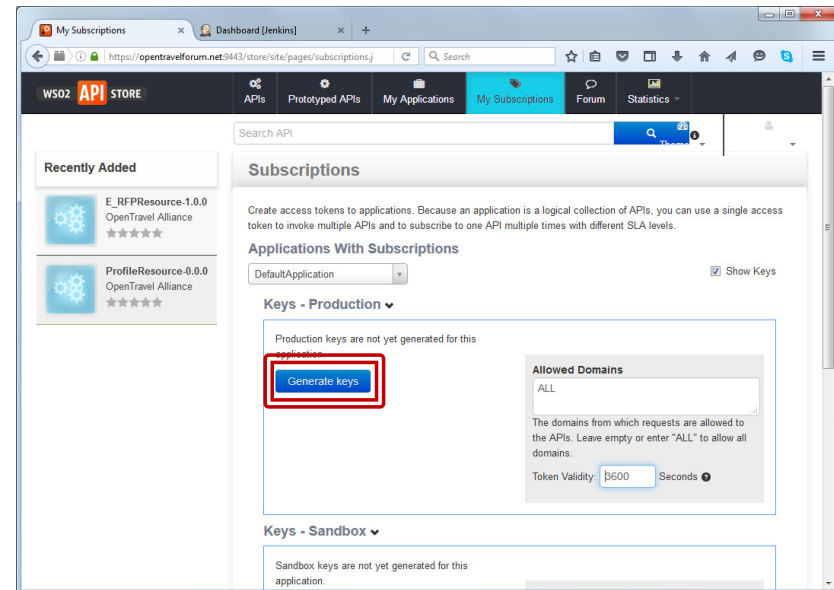
Step 1: Click the "Subscribe" Button

This will authorize you to call the API



Step 2: Generate a Production Access Token

This will create the Client Key & Secret required for OAuth2



Calling via the Secure Gateway (continued)

- Obtaining an OAuth2 Access Token

Sample Token Request (curl):

```
curl -k -d "grant_type=password&username=<userid>&password=<password>" -H  
"Authorization: Basic  
SGZFbDFqSlBkZzV0YnRyeGhBd3liTjA1UUdvYTpsNmMwYW9MY1dSM2Z3ZXPiaGM3WG9HT2h0NUFh"  
https://opentravelforum.net:8243/token
```

<userid> = Your API Store User ID
<password> = API Store Password

Basic Authentication Header:
<client-key>:<client-secret> (Base64 Encoded)

Sample Token Response:

```
{  
  "token_type": "Bearer",  
  "expires_in": 3600,  
  "refresh_token": "33c3be152ebf0030b3fb76f2c1f80bf8",  
  "access_token": "292ff0fd256814536baca0926f483c8d"  
}
```

OAuth2 Access Token

- Mock Service Invocation (curl example)

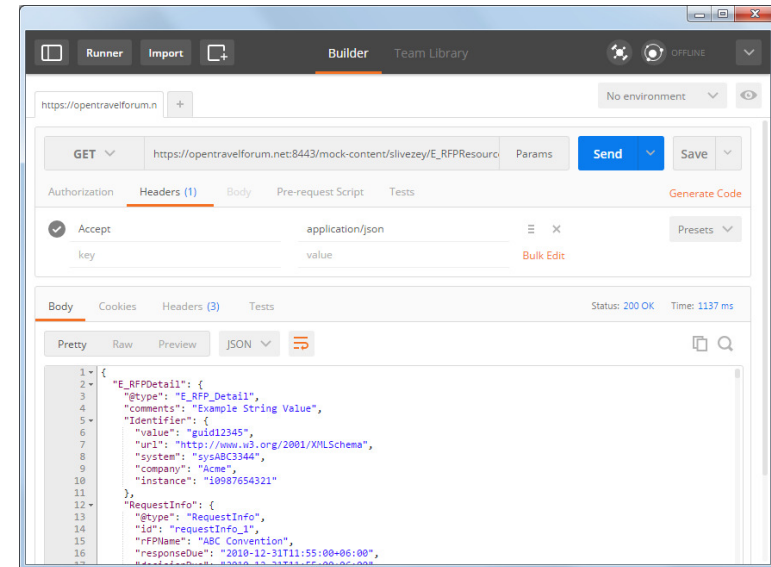
```
curl -H "Authorization: Bearer 292ff0fd256814536baca0926f483c8d"  
https://opentravelforum.net:8243/slivezey/E_RFPResource/1.0.0/myapi
```

Calling the Mock Content Server Directly

- Direct calls to the mock content server do not require authentication...

```
curl https://opentravelforum.net:8443/mock-content/slivezey/E_RFPResource/1.0.0/myapi
```

- For API's that support both XML and JSON, an `Accept` header can be used to specify the response format:



Review & Wrap-Up

The Hackathon Lite environment is intended to allow developers to quickly create and deploy API's and mock services.

During this session you have learned to...

1. Setup your local workstation for participation in the Hackathon Lite
2. Design, develop, and deploy mock services to a live DevOps environment
3. Call the mock services in a production-like environment



The real goal of the event is not to build mocks – but to show what we can do with those API's!