**Oracle Cloud Test Drive for PaaS and IaaS**

Lab 1.0

API Platform Cloud Service API First Lab

Contents

[1 Lab Scenario 2](#_Toc498958249)

[1.1 Logging into the Apiary.io 2](#_Toc498958250)

[1.2 The Apiary.io Environment 3](#_Toc498958251)

[1.2.1 Apiary Editor 3](#_Toc498958252)

[1.2.2 Interactive Documentation 4](#_Toc498958253)

[1.2.3 Mock Server 4](#_Toc498958254)

[1.2.4 API Inspector 4](#_Toc498958255)

[2 Desinging and Documenting an API with Apiary 4](#_Toc498958256)

[2.1 Create a New API Project 4](#_Toc498958257)

[2.2 Edit your API definition 5](#_Toc498958258)

[2.2.1 Edit the Sample API definition 5](#_Toc498958259)

[3 Apiary Interactive Documentation 8](#_Toc498958260)

[3.1 Update the API design with additional documentation 8](#_Toc498958261)

[3.2 Review the API Documentation 8](#_Toc498958262)

[4 Apiary Mock Server 9](#_Toc498958263)

[4.1 Using the Mock Server to Test your API 9](#_Toc498958264)

[4.2 Access the Mock Service directly 9](#_Toc498958265)

[5 Apiary API Inspector 10](#_Toc498958266)

[5.1 Use the API Inspector to review API requests 10](#_Toc498958267)

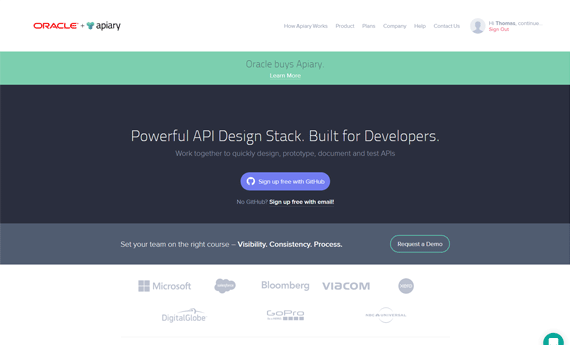
Introduction to the Lab and Logging-in

# Lab Scenario

The objective of this lab is to give you a comprehensive overview of the Oracle Apiary Cloud Service as an API Developer.

Oracle Apiary makes it easy for teams to work together to quickly design, prototype, document and test APIs. API first development, although API developers and consumers to collaborate and validate the design of the API before it is fully developed. This removes the iterative cycle of API development, dissatisfactions from the consumer, re-design while the enterprise adopts and adapts to changing requirements.

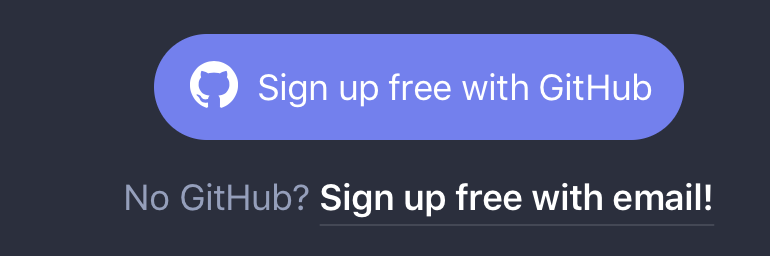
In this lab you will see how easy it is to create, document and test an API using the Oracle Apiary Cloud Service.



**Apiary.io Portal**: This portal is used by App developers, testers, architects, product managers, clients, and partners, allowing effective collaboration, bringing unique perspectives to the design of your APIs.

It provides a dashboard for web-based API Blueprint management, with complete control and visibility to project Role-based access, documentation, API editor, mock server for testing and is pre-integrated with GitHub.

## Logging into the Apiary.io

Access Apiary is free for individual developers but you must signup for an account. (no credit card needed)

Open your browser and navigate to: <http://apiary.io>

Click Signup for free: with GitHub or with email

|  |  |
| --- | --- |
| For GitHub\* | Email Only |
| /var/folders/3c/n2d_jp3x4nn9p8vdr15jsxsh0000gn/T/DIS_357.png /var/folders/3c/n2d_jp3x4nn9p8vdr15jsxsh0000gn/T/DIS_358.png | /var/folders/3c/n2d_jp3x4nn9p8vdr15jsxsh0000gn/T/DIS_354.png |
| Enter a username, email address and password for new accounts. Simply sign in for existing GitHub accounts | |

\*For existing GitHub users, this process will link your GitHub account with Apiary and share your GitHub login credentials for both services.

Check your email for the confirmation email and then complete the registration and login process.

After your first login, the Apiary service will present a short tutorial to get you understand Apiary and get you started, you may review this or continue with the Lab Exercise.

## The Apiary.io Environment

Apiary is focused on solving the fundamental task of improving API development and streamlining the business process of how work gets done.

The API Designer / Developer works with the Apiary tools to create teams who collaborate, design and document APIs, test & inspect mock versions of API services.

The Lab will focus on 4 areas of the Apiary tools:

* Apiary Editor
* Interactive Documentation
* Mock Server
* API Inspector

Let’s go through each component quickly. There is nothing to do here yet.

### Apiary Editor

The Apiary Editor is the foundation of your API design. Apiary Editor supports [API Blueprint](https://apiblueprint.org/) and [Swagger](https://help.apiary.io/api_101/swagger) API Description languages. The Apiary Editor shows a preview of your documentation while providing instant feedback to ensure correct syntax in your API document.

Your API document will be parsed with warnings and errors as you type. These errors will be displayed both in the editor header and on the lines where the warnings and errors exist.

### Interactive Documentation

Apiary interactive documentation is an interactive representation of your API Description for you to not only read and write, but to be a place where you can interact with your API—even before you’ve built it.

### Mock Server

The Mock Server allows you to quickly preview your API and share it with others. It allows you to try out your API as you design it, giving immediate feedback along the way in how it may be used.

The Mock Server accomplishes this by listening for requests as you’ve defined them in your blueprint. When a request is received to your Mock Server for a URL you’ve defined, the corresponding response for that request will be returned.

### API Inspector

The API Inspector to monitors requests to the Mock Server and captures both the requests and responses and allows you to view them.

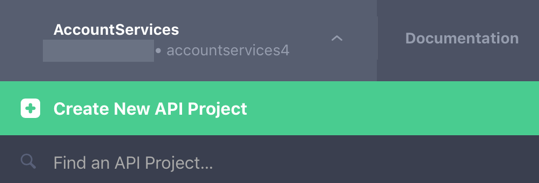
Next, let’s start to implement our scenario.

Lab Execution

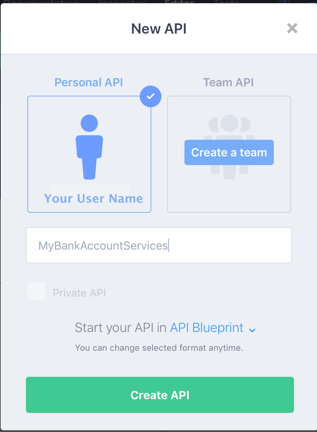
# Desinging and Documenting an API with Apiary

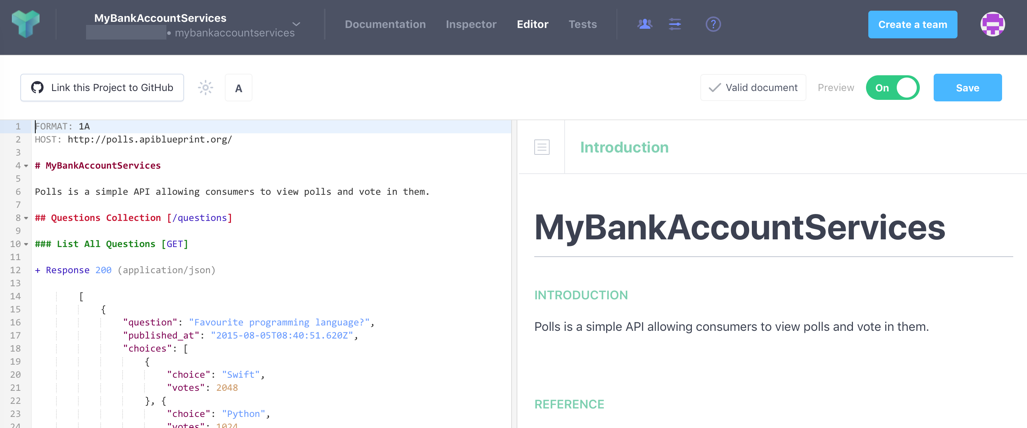
In this first stage, you will design a simple API using the Apiary Editor. You will create a project, which will also create a new GitHub project, if your accounts have been linked. Then you will edit the sample API to design and document your API.

## Create a New API Project

If you’ve already doen so, Log into [Apriary.io](http://apiary.io/).

1. Click on the Account Services menu item in the top left corner
2. Select Create a New API Project



1. Enter **MyBankAccountServices** for the API project name
2. Click Create API
3. You will now see your API project opened in the Apiary Editor with the code editor and documentation preview windows.

## Edit your API definition

We have selected to use the [API Blueprint](https://apiblueprint.org/) format, which is a open source project, to describe the API. This format and the Apiary editor will use the metadata and resource description to document and build the mock service of your API.

The API Blueprint uses levels to document your API, the first level uses one '#' which will become the API Name. For example:

# Polls

A headings section is used for Resources with two '##', a description and URI. For example:

## Question Collection [/questions]

Subheadings are used for actions with three '###', a description and HTTP method, followed by usage text and the Response(s)

You can refer to the [URI Template](https://help.apiary.io/api_101/uri-templates/) for additional details on structuring your API.

### Edit the Sample API definition

In the Code Editor

1. Edit the Host: and change the URI
   1. From: **http://polls.apiblueprint.org/**
   2. To: **http://accountservices.apiblueprint.org/**
2. Edit the API description as follows:
   1. From: **Polls is a simple API allowing consumers to view polls and vote in them.**
   2. To: **MyBankAccountServices is a simple API to demonstration the use of Apiary and the concept of API Design First.**
3. Edit the API resouce
   1. Change the Resourse:
      1. From: ## Questions Collection [/questions]
      2. To: **## My Bank Account Balance [/accountBalance]**
   2. Notice the documentation preview is being updated automatically as you change the design
4. Edit the API subheading and Response as follows:
   1. Change the subheading:
      1. From: ### List All Questions [GET]
      2. To: **### Get Bank Account** **[**GET**]**
   2. Change the json of the Response as following:
   3. From:

{

"question": "Favourite programming language?",

"published\_at": "2015-08-05T08:40:51.620Z",

"choices": [

{

"choice": "Shift",

"votes": 2048

}, {

"choice": "Python",

"votes":1024

}, {

"choice": "Objective-C",

"votes": 512

}, {

"choice": "Ruby",

"votes": 256

}

]

}

* 1. To:

{

"accountNr" : "12345678",

"bankName" : "Allied Irish Banks plc.",

"branch" : "Artane Roundabout, Dublin 5",

"custFirstName" : "YourFirstName",

"custLastName" : "YourLastName",

"custEmail" : "YourEmailAddress",

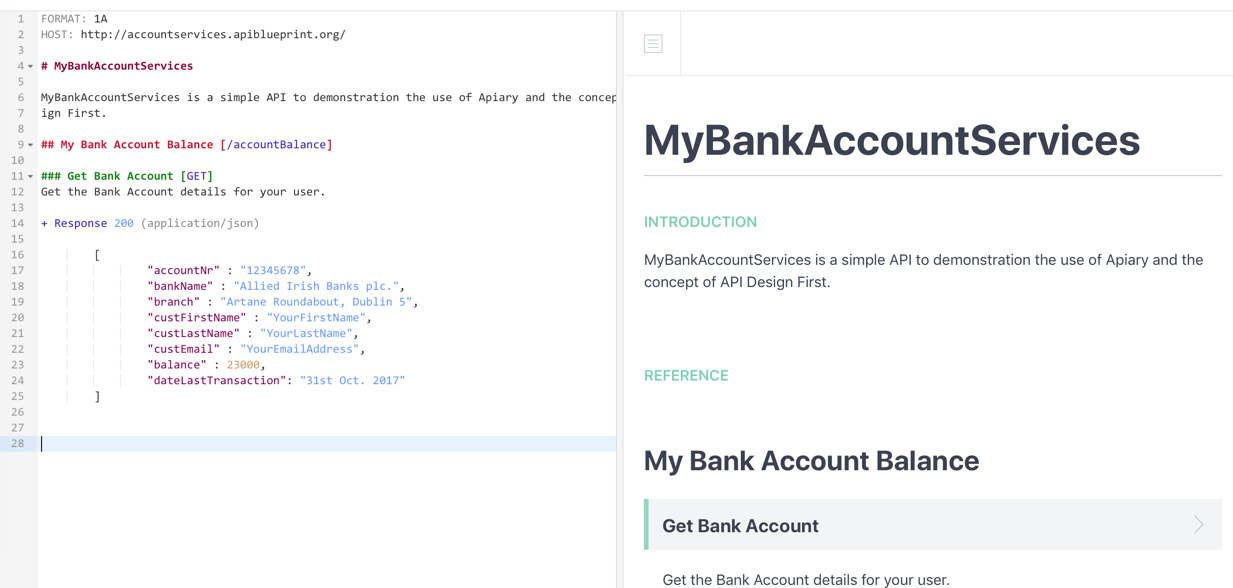
"balance" : 23000,

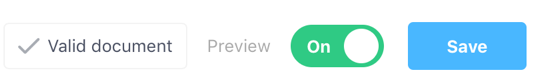
"dateLastTransaction": "31st Oct. 2017"

}

You may copy and paste this section but don’t forget to update the custFirstName, custLastName, and custEmail values with your information.

1. Delete the remaining Resource and Response sections. Your API code and documentation should look as follows, without any Warnings or Errors from the validator





1. Click on Save

# Apiary Interactive Documentation

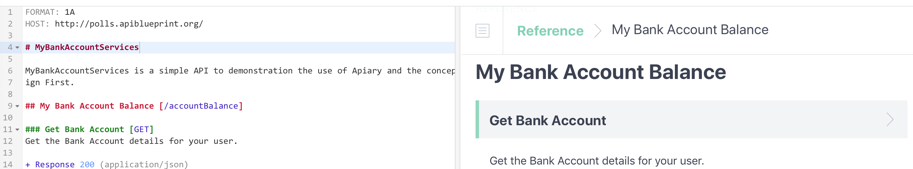
As you have seen the is generated automatically as the API is being designed based on the tagging of sections and the provided descriptions.

The interactive documentation contains two main columns: the human and machine columns. These two columns provide the separation that is important for reading the documentation and actually using tooling to interact with it.

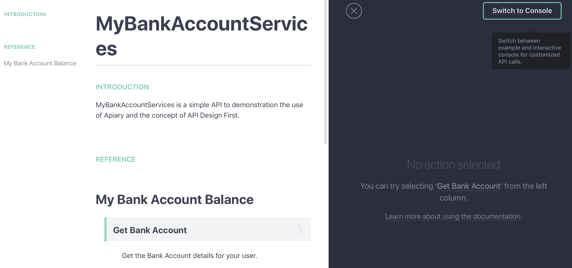
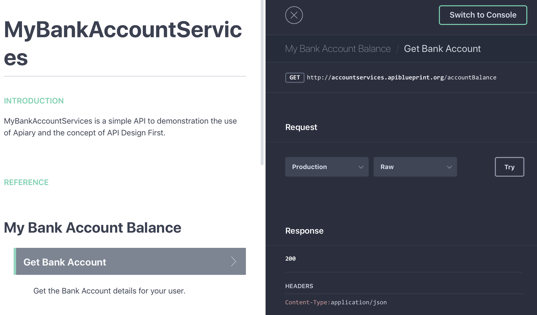
The Human Column provides visibility into the long-form description of your API. When you write your API Description, the goal is to write it as you would any other document you mean for other humans to read and consume.

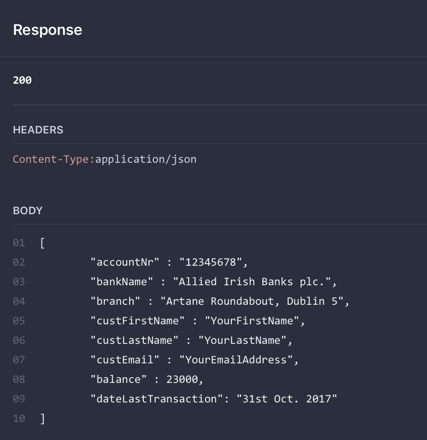
The Machine Column displays the information that clients and servers will be interested in when interacting with your API or Apiary’s mock and debugging servers.

## Update the API design with additional documentation

1. Update the API design and add a description for the Get Bank Account resource [GET] action
2. Type the following under the [GET ] action
   1. Get the Bank Account details for your user.
3. Preview the updated documentation
4. Click on Save

## Review the API Documentation

1. Navigate to the API Documentation
   1. Click on Documentation from the menu
   2. If shown, click leave page
2. Review the Introduction and Reference sections in the Human Column
3. Select the Get Bank Account action this will update the Machine column with the actual Resource, Request and Response for the API.
   1. Scroll down to preview the Response

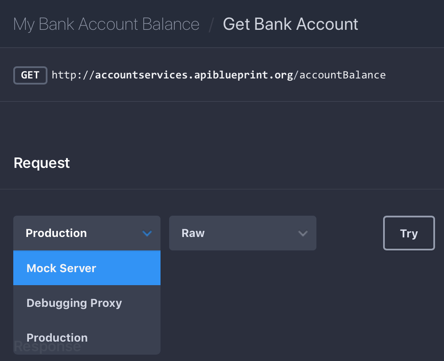
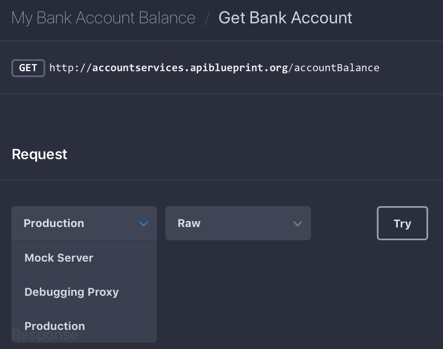


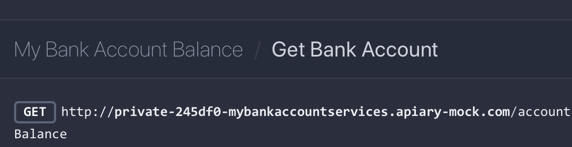
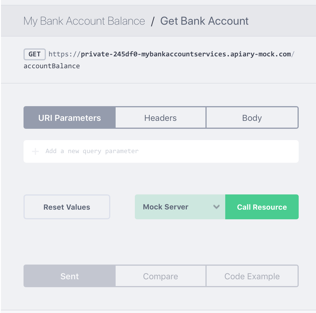
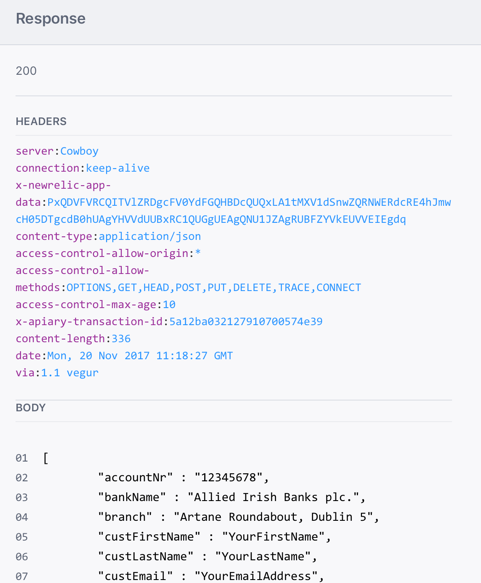
# Apiary Mock Server

In the design phase of an API, it is helpful to experiment with how you feel your API should work and iterate on this process quickly. In the past, this involved writing code as a prototype and setting up a server to run it. This takes time and directly effects the creativity behind the design process.

The Mock Server allows you to try out your API as you design it, giving immediate feedback along the way in how it may be used.

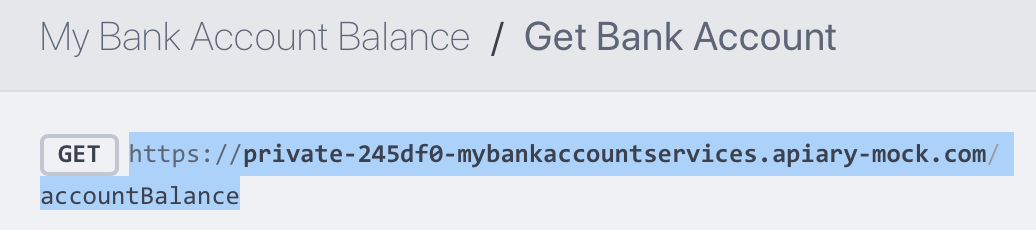
## Using the Mock Server to Test your API

The Mock Server is automatically created each time you publish your API Description. This means the only thing you have to do to get started is to write your API Description, include specific requests and responses for your resources, and click publish.

1. From the Documentation menu, ensure your request action is selected.
2. Change the Request from Production to Mock Server
3. Notice that the request URI has changed to the Mock server
4. Click Try
5. This will change from the Example to the Console View where you may test the API. As this API does not require any input parameters, we can simple invoke it without any parameters.
6. Click **Call Resource**
7. The Mock Server will return the first response from the API
   1. Scroll down to review the response

## Access the Mock Service directly

You may invoke the Mock Service directly in your browser as well by using the API URI for the Mock Server and your API service resource.

1. Copy the service URL for the GET http request:
   1. Sample - https://private-245df0-mybankaccountservices.apiary-mock.com/accountBalance

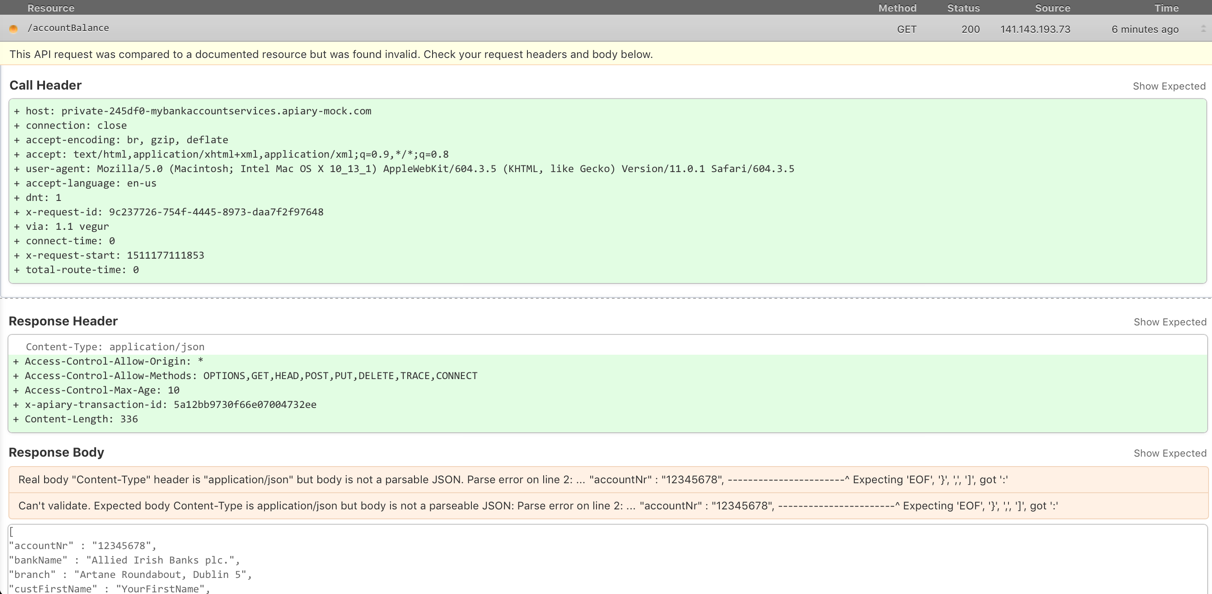


1. Paste this in your Browser and submit the request
2. Review the API response

# Apiary API Inspector

Any request to or response from the Mock Server will be logged by the Apiary Debugging Server and made available in the API Inspector. This means that you can use your favorite tool for interacting with your API (such as curl or Paw) and Apiary will log that traffic.

## Use the API Inspector to review API requests

1. Navaigate to the API Inspector
   1. Click on Inspector from the menu
   2. If shown, click leave page
2. You should see atleast two requests for your API resource
3. Select one of the resource request to view the Request and Response

Congratulations you have completed the APIPCS API Desigin First lab.