Date :  13th Oct 2023

Appstore DATA ADAPTER (1.0.0)

# Overview

App Store data adapter is one of the best and fastest performing APIs for iTunes/Mac app store data. This is an unofficial App Store API offering a solution for developers who crave for a gateway for their App Store API calls.

It does not cache nor store any information in their servers, instead it scraps the app store in real-time when requested.

If the developer wants to fetch its iTunes/Mac app apps details in the app, this adapter is very helpful in fetching the iPhone/iPad/Mac apps data.

# Getting Started

## Prerequisites

* Volt Foundry
* ApiLayer license key - <https://apilayer.com/marketplace/app_store-api#documentation-tab>

## Importing the adapter

**To import the Data Adapter to Volt Foundry, do the following:**

1. Sign in to the  [HCL Foundry](https://manage.hclvoltmx.com/).
2. From the left navigation menu, select **API Management**.
3. In **API Management**, select **Custom Data Adapters**.  
   
4. Click **IMPORT** to import a custom data adapter.  
   
5. On the Import Data Adapter dialog box, click browser to import.  
   
6. Select **AppStoreAPI** zip file and click **IMPORT**.

After you import the data adapter, Volt Foundry opens a window that shows the metadata of the data adapter.A screenshot of a computer

Description automatically generated

After you import the data adapter, you can view it on the Custom Data Adapters page and use it to create services on Volt Foundry.



## [Creating an Integration service](javascript:void(0);)

After you import the data adapter into Volt Foundry, you can use it to create an Integration Service.

Follow the given steps to create an Integration service using the **AppStoreAPI** Adapter.

1. Sign in to the [HCL Foundry](https://manage.hclvoltmx.com/).
2. From the left navigation menu, select **API Management**.
3. In **API Management**, select **Integration**.  
   
4. To create a new service, click the **+** button or the **CONFIGURE NEW** button.  
   
5. On the Service Definition tab, select the service type as **AppStoreAPI**, and click **SAVE**.  
   A screenshot of a computer

   Description automatically generated

Alternatively, you can also create a Foundry app and create an Integration service inside it.

E. [Creating and Executing operations](javascript:void(0);)



After you create an integration service, you can create and execute operations using the service.

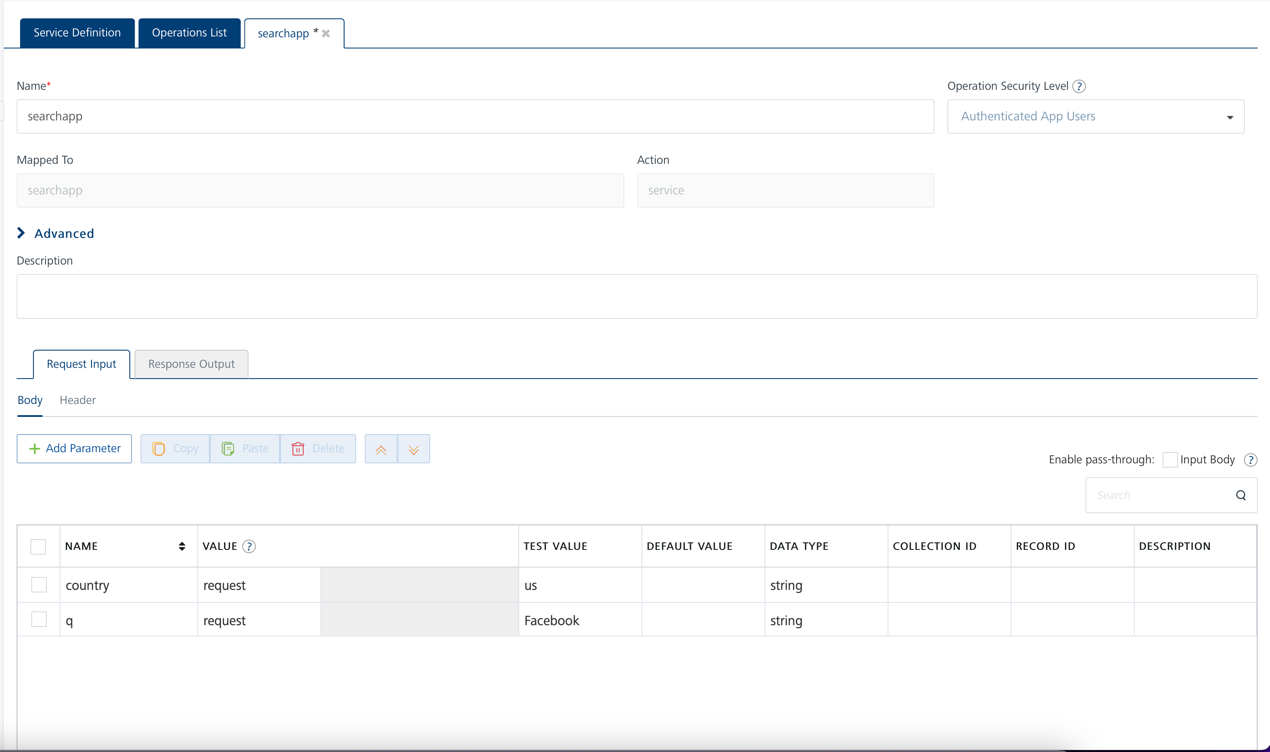
#### Creating an Operation

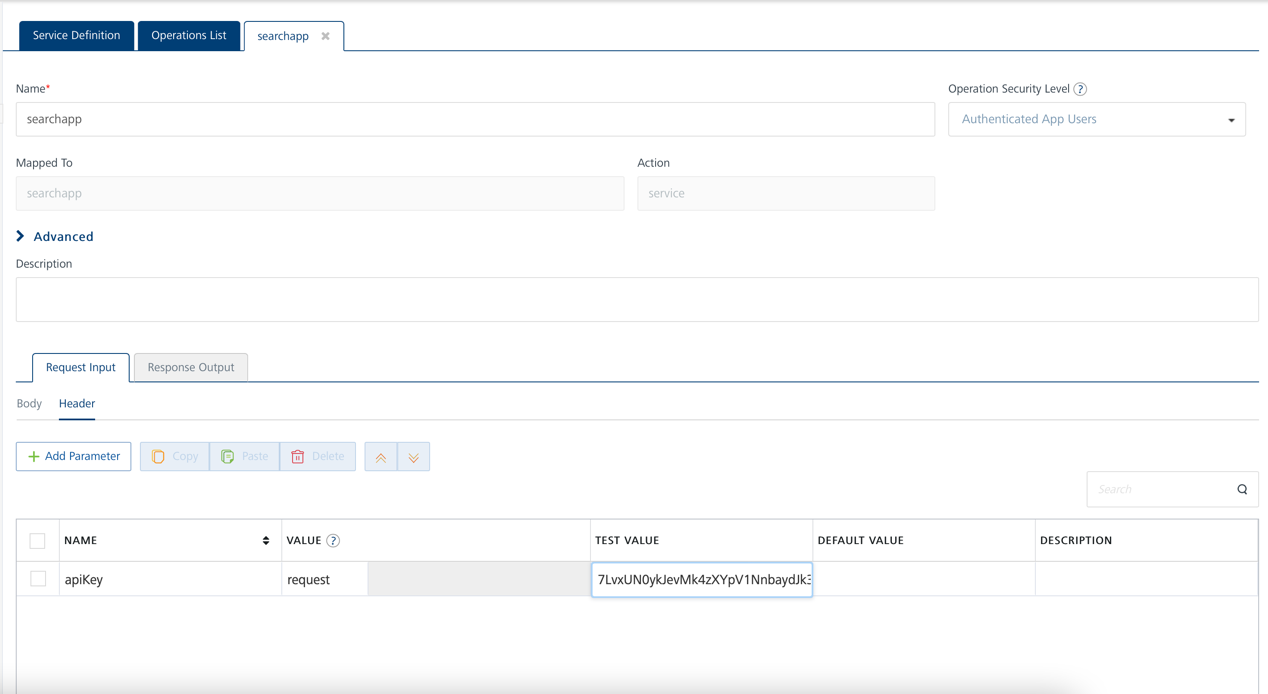
* In **API Management/Foundry app you created**, in the **Integration** section, select the service that you created.
* After you select the service, navigate to the **Operation List** tab.  
  
* From the drop down list, select an operation that you want to execute, and click **ADD OPERATION**.  
  A screenshot of a computer

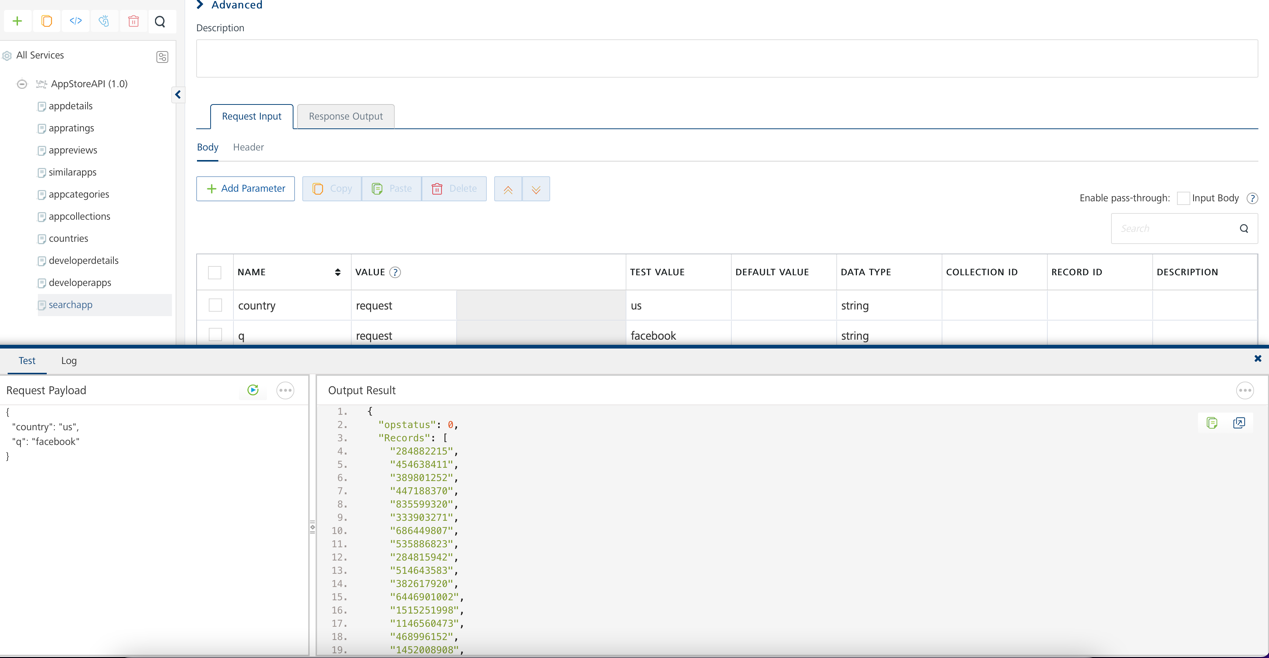
  Description automatically generated

#### Executing an Operation

* From the **Operations List** tab, in the **Configured Operations** section, select the operation you want to execute.
* A screenshot of a computer

  Description automatically generatedOn the Operation Page, in the Request Input tab, enter a TEST VALUE for all the fields.  
  



* Select a run-time environment and click **Save and Fetch Response** to get a response based on your inputs.  
    
  

## [Publishing your application](javascript:void(0);)

If you want to use the services in client applications, you need to publish an app to a run-time environment. You can create the service (as described above) in an application or import the service into an application and publish the application.

# References

## Endpoint Documentation

### Operations:

### i. appDetails

|  |  |
| --- | --- |
| **Description:** | Fetches information of any app using app id. |
| **Parameters:** | **Request Params:** i. country(required) - String - Two-character country parameter. To get a full list of supported countries check /countries endpoint.  ii. id(required) - String - The id of the app. **Header Params:**i. apiKey(required) - String - license key must be passed in the service header param |
| **Restrictions:** | NA |

### ii. appRatings

|  |  |
| --- | --- |
| **Description:** | Retrieves the ratings information of an application |
| **Parameters:** | **Request Params:** i. country(required) - String - Two-character country parameter. To get a full list of supported countries check /countries endpoint.  ii. id(required) - String - The id of the app. **Header Params:**i. apiKey(required) - String - license key must be passed in the service header param |
| **Restrictions:** | NA |

### iii. appReviews

|  |  |
| --- | --- |
| **Description:** | Retrieves the reviews of an application |
| **Parameters:** | **Request Params:** i. country(required) - String - Two-character country parameter. To get a full list of supported countries check /countries endpoint.  ii. id(required) - String - The id of the app. **Header Params:**i. apiKey(required) - String - license key must be passed in the service header param |
| **Restrictions:** | NA |

### iv. similarApps

|  |  |
| --- | --- |
| **Description:** | Returns the list of 'customers also bought apps' shown in the app's detail page |
| **Parameters:** | **Request Params:**  i. id(required) - String - The id of the app. **Header Params:**i. apiKey(required) - String - license key must be passed in the service header param |
| **Restrictions:** | NA |

### v. appCategories

|  |  |
| --- | --- |
| **Description:** | Gets the full list of categories to use for in the /list endpoint. |
| **Parameters:** | **Request Params:** NA **Header Params:**i. apiKey(required) - String - license key must be passed in the service header param |
| **Restrictions:** | NA |

### vi. appCollections

|  |  |
| --- | --- |
| **Description:** | Gets the full list of collections to use for in the /list endpoint. |
| **Parameters:** | **Request Params:** NA **Header Params:**i. apiKey(required) - String - license key must be passed in the service header param |
| **Restrictions:** | NA |

### vii. countries

|  |  |
| --- | --- |
| **Description:** | Gets the full list of countries |
| **Parameters:** | **Request Params:** NA**Header Params:**i. apiKey(required) - String - license key must be passed in the service header param |
| **Restrictions:** | NA |

### viii. developerDetails

|  |  |
| --- | --- |
| **Description:** | Retrieves the details of a developer |
| **Parameters:** | **Request Params:** i. country(required) - String - Two-character country parameter. To get a full list of supported countries check /countries endpoint.  ii. developer id(required) - String - The id of the developer. **Header Params:**i. apiKey(required) - String - license key must be passed in the service header param |
| **Restrictions:** | NA |

### ix. developerApps

|  |  |
| --- | --- |
| **Description:** | Retrieves the list of apps published by a developer |
| **Parameters:** | **Request Params:** i. country(required) - String - Two-character country parameter. To get a full list of supported countries check /countries endpoint.  ii. developer id(required) - String - The id of the developer. **Header Params:**i. apiKey(required) - String - license key must be passed in the service header param |
| **Restrictions:** | NA |

### x. searchApp

|  |  |
| --- | --- |
| **Description:** | Search for an app using the search term q. |
| **Parameters:** | **Request Params:** i. country(required) - String - Two-character country parameter. To get a full list of supported countries check /countries endpoint.  ii. q(required) - String - The term to search. **Header Params:**i. apiKey(required) - String - license key must be passed in the service header param |
| **Restrictions:** | NA |

# Revision History

Adapter version 1.0.0:

## Known Issues

## Limitations

Note: that API is not affiliated in any way by Apple Inc. or its affiliates.