



Kony Fabric

Debugging your apps with Kony Fabric

Release V8

Document Relevance and Accuracy

This document is considered relevant to the Release stated on this title page and the document version stated on the Revision History page. Remember to always view and download the latest document version relevant to the software release you are using.

Copyright © 2017 Kony, Inc.

All rights reserved.

January, 2018

This document contains information proprietary to Kony, Inc., is bound by the Kony license agreements, and may not be used except in the context of understanding the use and methods of Kony, Inc., software without prior, express, written permission. Kony, Empowering Everywhere, Kony Fabric, Kony Nitro, and Kony Visualizer are trademarks of Kony, Inc. MobileFabric is a registered trademark of Kony, Inc. Microsoft, the Microsoft logo, Internet Explorer, Windows, and Windows Vista are registered trademarks of Microsoft Corporation. Apple, the Apple logo, iTunes, iPhone, iPad, OS X, Objective-C, Safari, Apple Pay, Apple Watch, and Xcode are trademarks or registered trademarks of Apple, Inc. Google, the Google logo, Android, and the Android logo are registered trademarks of Google, Inc. Chrome is a trademark of Google, Inc. BlackBerry, PlayBook, Research in Motion, and RIM are registered trademarks of BlackBerry. SAP® and SAP® Business Suite® are registered trademarks of SAP SE in Germany and in several other countries. All other terms, trademarks, or service marks mentioned in this document have been capitalized and are to be considered the property of their respective owners.

Revision History

Date	Document Version	Description of Modifications/Release
11/02/2017	1.0	Document release for Debugging your apps with Kony Fabric Training - V8

Table of Contents

1. Objective	5
2. Introduction to Debugging	6
3. Service Testing using Kony Fabric Console	9
3.1 Testing a Service from Kony Fabric Console	9
4. Log Services	11
4.1 Kony Log Server	11
5. Testing Services from Server Consoles	18
5.1 Retrieving the list of books based on the Published Date	18
5.2 App Server Console	19
5.3 Sync Console	20
5.4 Engagement Console	21
5.5 Management Console	21
5.6 Testing a Service from Admin Console	22

1. Objective

Hello, everyone! Welcome back. By now, we assume that you have already installed and created your first Kony Fabric application and linked your backend services.

In this course, we will walk you through the process of debugging your backend services with Kony Fabric, and familiarizing yourself with the various options available to debug your services.

In this training, you will learn:

- Various debugging mechanisms in Kony Fabric.
- Viewing service logs using Kony Fabric Log Server.
- Customizing and filtering your logs.
- Visualizing your service logs, including easy error capturing.

2. Introduction to Debugging

While working with any software solution as you build and expand your services, you may run into several problems that you need to analyse and fix. We have tried to make this process simple for you, by providing a wide range of debugging mechanisms.

Kony Fabric provides three options to debug your application:

From Fabric console

- Test your service when configuring it in Kony Fabric. This will help you check the configuration of the service.
- You can also validate the service with different input values and test its behaviour

Using Fabric Log Services console

- See detailed logs and perform many filtering and search operations on your logs. You can also visualize your log statistics here.
- A powerful tool to understand and analyse the performance of your services.

Using Server consoles

- Test the working of your published service from the server console for each Fabric service. You have more detailed options to test and view service response from this section.
- This will simulate the call that your frontend application will use to call the Fabric service

Through this course, we will understand each mechanism with more details.

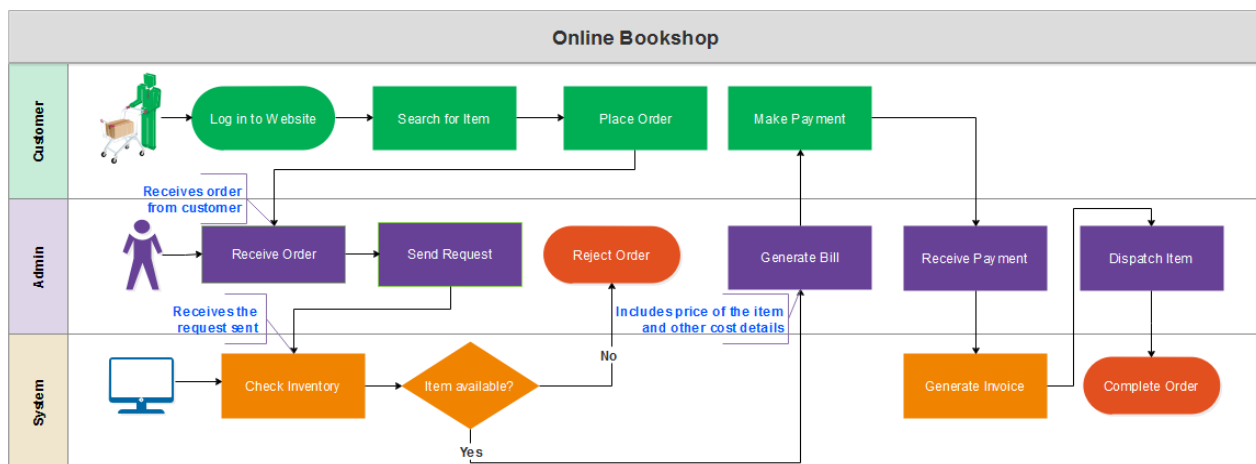
To understand this course better, we will try to explain each debugging option in detail by creating a simple **Book Store Management** application.

The main features of the application include:

- Allow customers to orders a book, and track it.
- Verify the number of books in the stock.

- Track activities performed by the Store staff.
- The planned application flow is as below :

A customer logs on to the website of the bookstore and submits an order for the required book. The system uses a web service to address this inquiry and to verify if the quantity currently in stock is sufficient to process the requested order. If the requested book is available in the inventory, the system generates a quotation that shows the price and other cost-related details of the book. Upon receiving payment, the item is dispatched with the billing address and other necessary details. Each action performed by the staff in the process of dispatching the order to the customer uses a web service. Take a look at the flow diagram below to understand this application better:



Note: Using Kony Fabric, the Administrator for this application will be able to monitor all services in this application, and resolve problems quickly.

We have created this sample Kony Fabric application for you to play around with. You can download the application and import it into your Kony Fabric instance to get a hands-on feel of this course.

To handle book title searches, we have used an API provided by The New York Times as an [OpenAPI service in the Integration Services page](#). You can refer the API spec [here](#).

The **Inventory Management** for the **Book Store** is handled by a **Storage Object** service in Kony Fabric. You can learn more about Storage services [here](#).

The rest of this course will walk you through debugging various facets of developing this application.

For more information, refer [How to Debug in Kony AppPlatform Tech Talk](#)

3. Service Testing using Kony Fabric Console

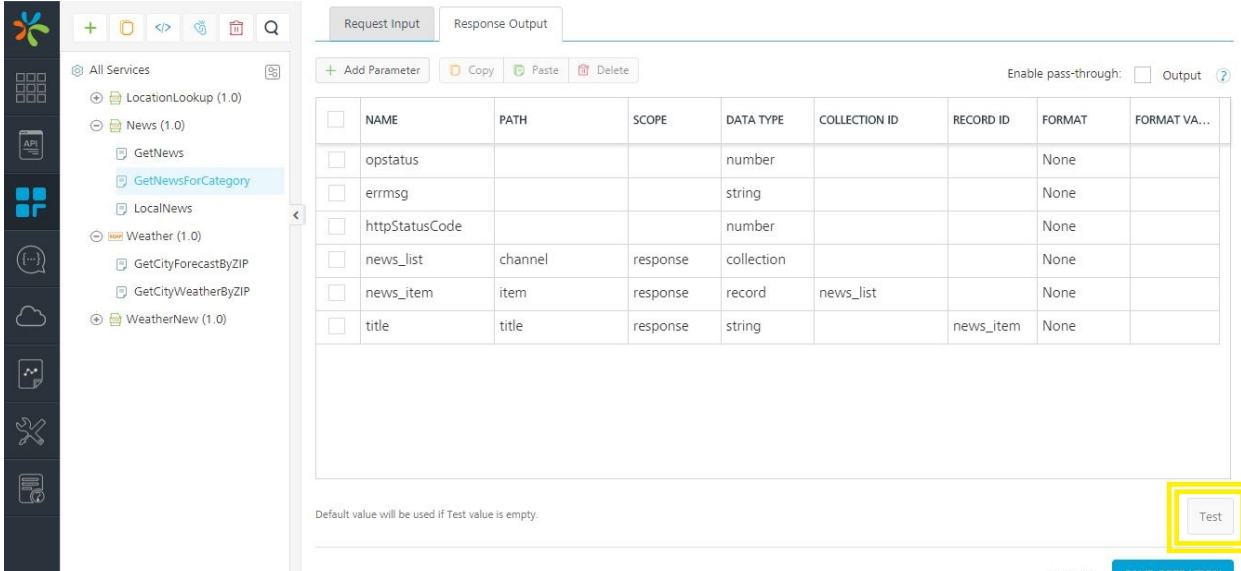
When configuring your Kony Fabric services, the first thing you must do, is to ensure the connectivity to the backend services is established properly by testing them from Kony Fabric console. For this module, you can use the application provided for the **Sample BookStore** application (Kony BookStore Sample) to try out service testing.

3.1 Testing a Service from Kony Fabric Console

First, import the provided application via the **Apps** section. Once you import the application, navigate to the **NYTimes Integration** service. You see that we have a list of pre-configured operations here :

- `GET_lists-date-list-json` : Retrieves the list of bestsellers on a given date.
- `GET_lists-format` : Search for book lists by a given name.
- `GET_lists-overview-format` : Returns the latest bestsellers list.

The Testing mechanism in Kony Fabric console differs slightly based on the type of the service you configure. For Integration services, you can see **Fetch Response** and **Test** buttons to retrieve service data, as shown below :



The screenshot displays the Kony Fabric console interface. On the left, a sidebar lists services under categories like 'All Services', 'LocationLookup (1.0)', 'News (1.0)', 'Weather (1.0)', and 'WeatherNew (1.0)'. The main area shows the 'Request Input' tab for a service. It features a table with columns: NAME, PATH, SCOPE, DATA TYPE, COLLECTION ID, RECORD ID, FORMAT, and FORMAT VA... The table contains several rows of parameters, including 'opstatus', 'errmsg', 'httpStatusCode', 'news_list', 'news_item', and 'title'. A 'Test' button is highlighted with a yellow box at the bottom right of the interface. Below the table, there is a text input field with the placeholder text 'Default value will be used if Test value is empty.' and buttons for 'CANCEL' and 'SAVE OPERATION'.

<input type="checkbox"/>	NAME	PATH	SCOPE	DATA TYPE	COLLECTION ID	RECORD ID	FORMAT	FORMAT VA...
<input type="checkbox"/>	opstatus			number			None	
<input type="checkbox"/>	errmsg			string			None	
<input type="checkbox"/>	httpStatusCode			number			None	
<input type="checkbox"/>	news_list	channel	response	collection			None	
<input type="checkbox"/>	news_item	item	response	record	news_list		None	
<input type="checkbox"/>	title	title	response	string		news_item	None	

RAML, **OpenAPIs**, and **Data Adapters** behave slightly differently to make the experience better for you. For these services, you can see a **Test** tab on the **Operation Details** page. Lets look at one of the services in our **BookStore** application as `GET_lists-overviewformat` to understand a test scenario.

The service shows you the latest bestsellers list, based on the publishing date. Here's what the API doc says :

Parameters

published_date string

Location: `query` `?published_date=xyz`

The best-seller list publication date. YYYY-MM-DD

You do not have to specify the exact date the list was published. The service will search forward (into the future) for the closest publication date to the date you specify. For example, a request for `lists/overview/2013-05-22` will retrieve the list that was published on 05-26.

If you do not include a `published_date`, the current week's best-sellers lists will be returned.

api-key string

Location: `query` `?api-key=xyz`

format string *required*

Location: `path` `/lists/overview.json`

Allowed values are:

- json
- jsonp

Let us see how Kony Fabric responds to a valid and invalid data in this API call. Once your services are configured properly and tested in Kony Fabric Console, you can deploy your service, and invoke it. Once a service is published, you can test the live API using Kony Log Server, or the Admin Console.

In the next section of this tutorial, we will dive deeper into debugging your live APIs.

4. Log Services

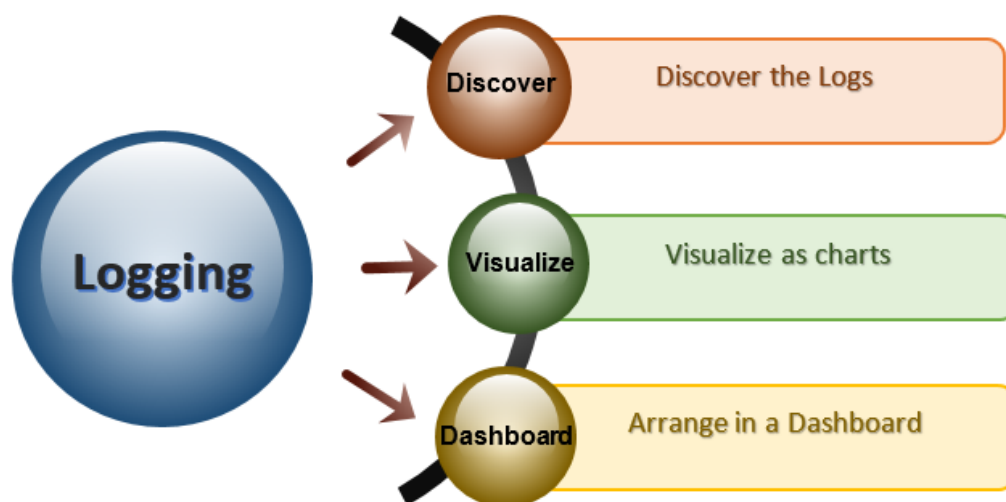
Once your developed APIs are live, they can be integrated into your apps. Once your client application is built and in use, you will need a deeper diagnostic tool to analyse any runtime issues with your services. Sometime you may encounter sudden outages in sections of your application. There can be any reason such as a service is no longer working as expected, or you run into issues after updating your device to a new OS version. This becomes a difficult situation to debug, and for live apps, it's critical to resolve such problems as quickly as possible.

4.1 Kony Log Server

Kony Log server was designed to help you solve this very problem. This tool provides granular log level control and the ability to view logs in real-time. Using Kony Log server, you will be able to view all application and service issues in one place, making it easy to analyze the root cause of your problem. The tool also lets you quickly filter out logs to quickly triangulate the piece causing the problem. The log services console is accessible from the **Environments** tab in Kony Fabric console. There is a host of features available in the portal to help you quickly debug your issue.

More on that below!

The **Log Services** portal displays three main features - **Discover**, **Visualize**, and **Dashboard**.

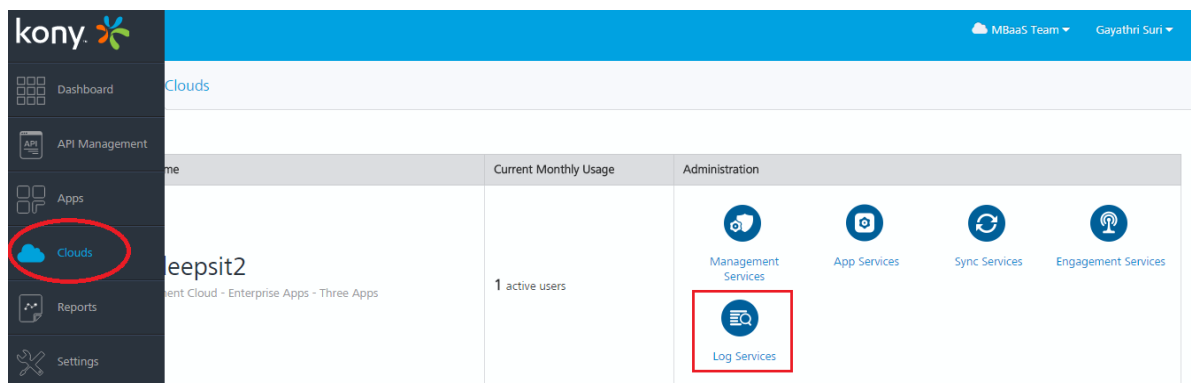


Important: To get access to Kony Log Server, contact your Sales representative.

- **Discover:** You can **Search logs** by providing a keyword or customizing the available filters to view your desired logs.
- **Visualize:** You can **Visualize your filtered** logs into predefined chart forms. We provide easy, predefined charts for you to choose from.
- **Dashboard:** Arrange your visualizations into a dashboard for daily updates.

To access **Log Services** follow the below steps:

1. Log on to Kony Fabric Console. For more information on accessing Kony Fabric Console, refer to [Accessing Kony Fabric Cloud](#).
2. After successful login, click **Environments/Clouds** from the left pane on the page. In this section, you can view all the consoles displayed for the registered cloud account.
3. From the list of consoles displayed, click **Logging Services**.



As discussed in the previous sections of this course, the log services module includes the following three sections:

- Discover
- Visualize
- Dashboard

Let's look into each of these items in a bit more detail.

4.1.1 Discover

As the name suggests, the section lets you discover logs generated by **Integration** and **Storage** services. All the service requests made by your account will be shown in this section. You can use some suggested filters, such as `status:200`, or simply enter the search criteria in the provided search fields to find your required logs.

You can also modify your query to retrieve log data for different time periods using the time filter provided on the upper right corner of the screen. By default, the system displays the logs captured in the last 15 minutes of the current day. You can also enter a keyword in the search field to find log data of services containing your keyword. The right pane displays filters to refine the displayed search results.

4.1.1.1 Tracking an Error in your application

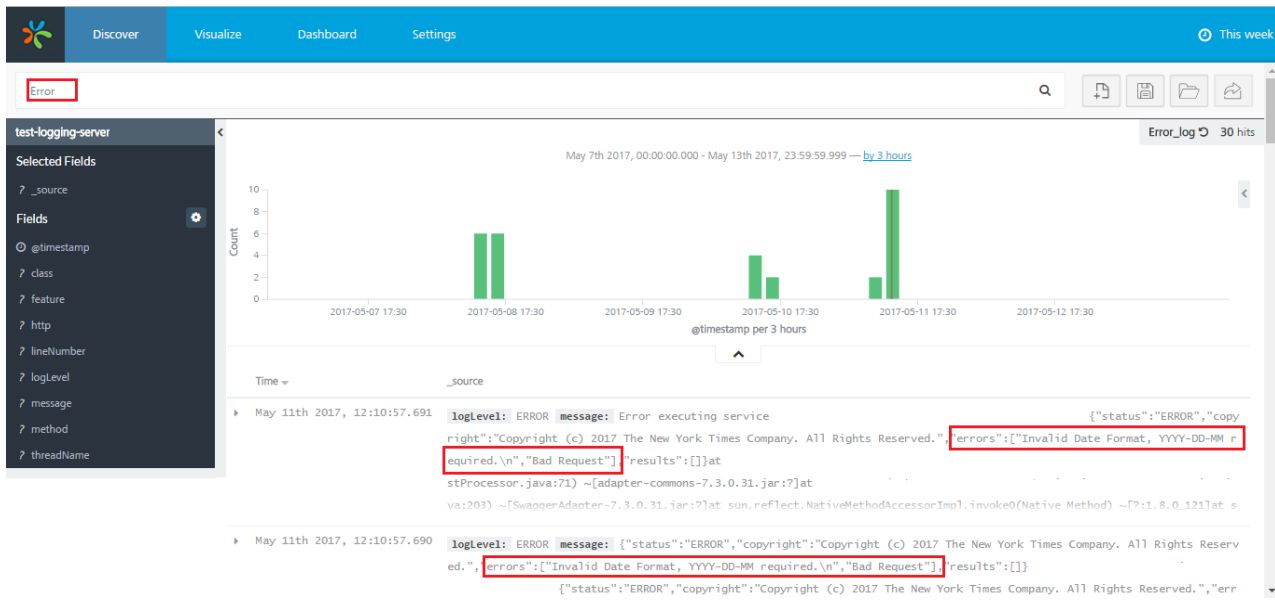
Having your live application malfunctioning is every app developer's nightmare. In this section, we will show you how to pick out your error logs from the entire application's log data. Let us continue with the scenario of the online bookstore described in the previous section.

In the previous section, while looking at the **Test from Console** option, we received an error message from the server, when we sent parameters in an incorrect input format.

Now, let us view the error logs that were generated in the **Log server** while testing the service for the same scenario - Retrieving the list of books available in the inventory based on the published date.

Here is a refresher - Our primary test parameter was the published date; if the date format provided is incorrect, the system displays an error. Using the **Log Services** module, we should be able to track and analyze this error quickly.

By providing a search parameter in the **Search** bar, you can track the error logs, which were generated while testing the service for the bookstore. In this example, let us use the key parameter as **ERROR**. Notice that it enables the system to display all results that fulfill the criteria as displayed below.



You can also save your search filter by clicking the **Save Search** option. Saving the search filter is an efficient way to access your results quickly in the future.

The screenshot shows the 'Save Search' dialog box. It has a search bar with the text 'NewSavedSearch_1' and a 'SAVE' button.

4.1.2 Visualize

You can visually illustrate relevant information by using several types of visual statistic charts. Visualization provides you options to design different charts for the different log scenarios.

You can either perform a new search or use your saved search to create visualizations.

To create an Visualization chart, perform the following steps in the Log Server console:

1. Click on the **Visualization** tab. The list of available charts appears.
2. Choose a chart type from the **Create Visualization** page.
3. Select a search source from a **Saved Search**, or create a new **Search** criteria.

As in our user scenario, we will select our saved search to tracks error logs that were generated while testing the bookstore service.

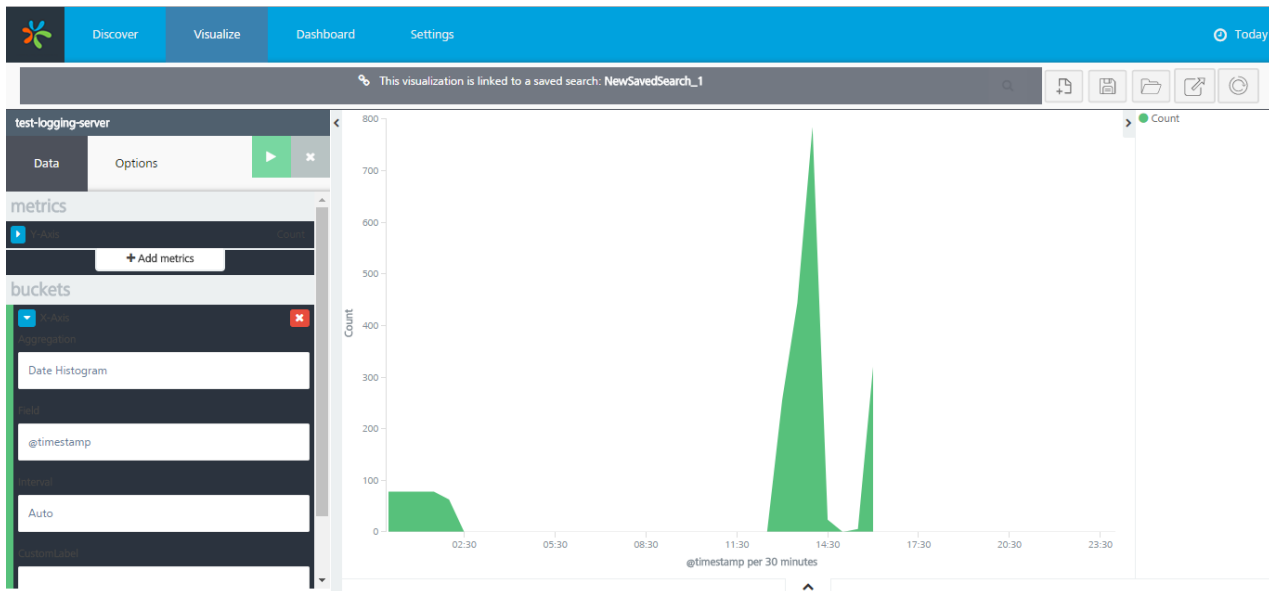
To track error logs, follow these steps:

1. The system displays the visualization based on the search parameters you have provided.
2. Make the required changes by using the filters provided in the left pane.
3. Click **Save Visualization** to save the visualization.

Let us walk thru a few of the visualization options available

4.1.2.1 Area Chart

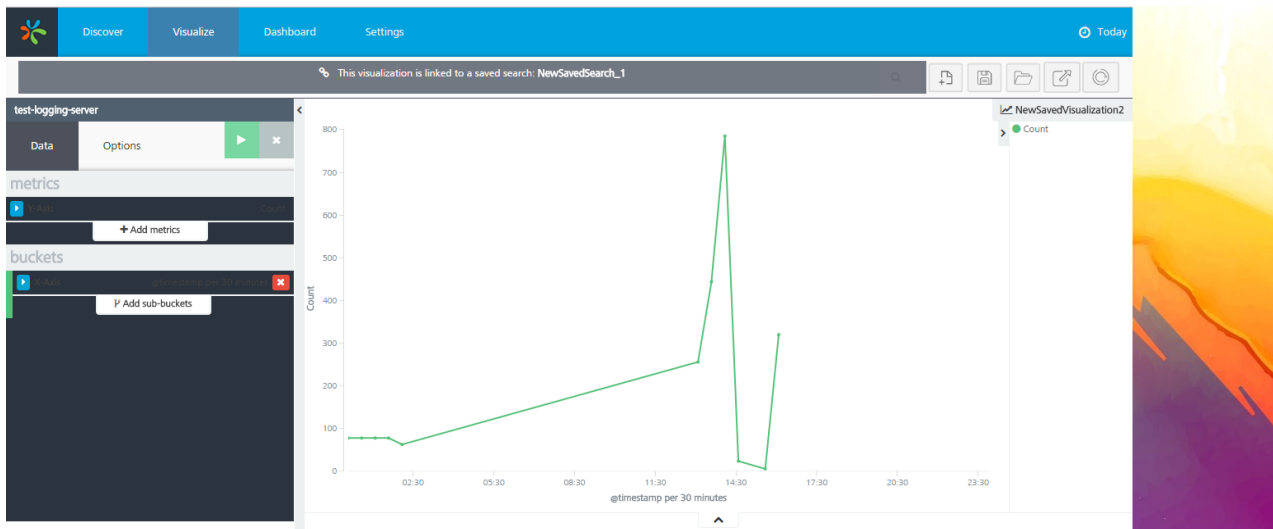
You can create an area chart to display stacked timelines. This graph shows the times when an `ERROR` message was encountered.



Similarly, we can prepare more charts with the same data.

4.1.2.2 Line Chart

You can use line charts for high-density time series.



You should choose the Chart that best represents your data.

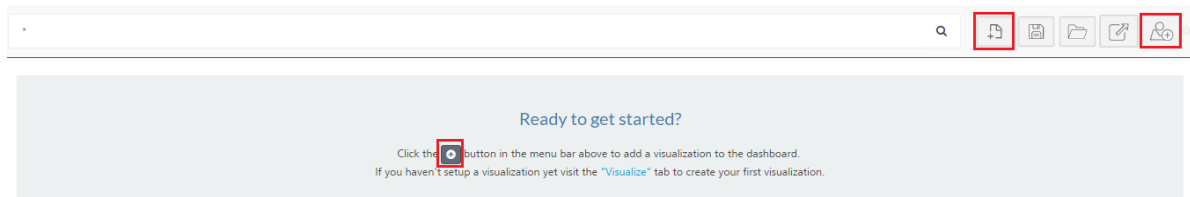
4.1.2.3 Dashboard

For a one-stop view of all your important log data, you can represent the charts created in the **Visualization** tab in a dashboard.

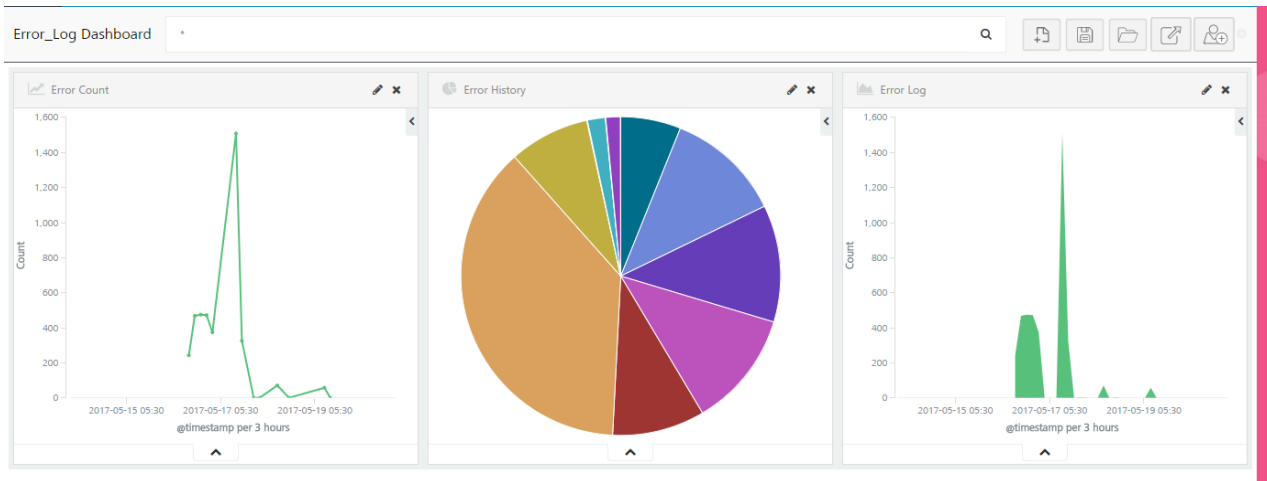
Based on the error log data, we have created three types of charts. Now, let us select the three saved visualizations and create a dashboard for our scenario.

To access and create a dashboard in **Log Services**, follow these steps:

1. Click the **Dashboard** tab.
2. Click the + icon on the screen, or click the **Add Visualization** button at the upper right corner to add the saved visualizations to the dashboard. You can use the **Add Visualization** button to add any number of visualizations to the dashboard. We have created three different visualizations to represent the same log data, and added them in our **Dashboard**.



Here is what my dashboard looks like now :



You can even edit or delete visualizations from the dashboard. Click **Save** to save the created dashboard.

5. Testing Services from Server Consoles

The **Logging Service** module displays the logs generated in the **Integration** and **Storage** services. If your application uses other capabilities of Kony Fabric, such as **Sync** and **Engagement**, you can use the individual feature's console interface to retrieve the desired logs.

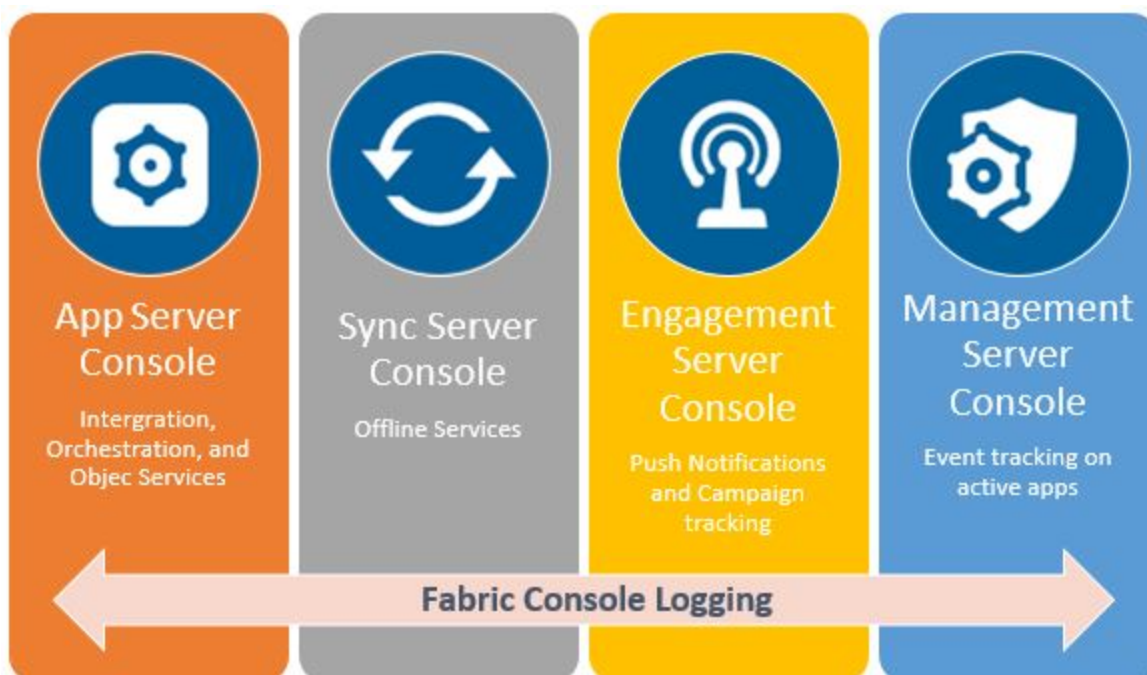
Note: You can test the specific service in the individual feature's console interface only after publishing the service in Kony Fabric environment.

As in previous sections of this tutorial, we discussed a scenario of an online bookstore that received orders from customers. Let us consider an example related to the scenario.

5.1 Retrieving the list of books based on the Published Date

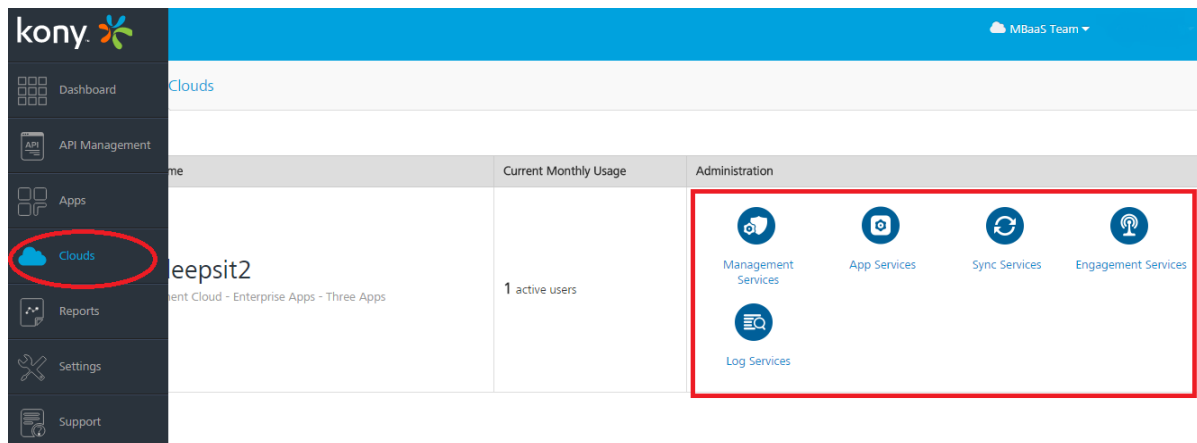
To retrieve the books available in the inventory based on the specified published date, we need to create a service (related to the scenario) and test it. For more information on creating a service and corresponding operations to the service, refer to [Integration Service Designer](#).

You can generate logs for the following consoles.



To access server consoles from **Kony Fabric Console**, follow the steps below :

1. Log on to your **Kony Fabric Console** account and click **Environments/Clouds**.
2. Under the **Environments/Clouds** section, you can view the different consoles available for your cloud account.



5.2 App Server Console

The App server console helps you to test in **Integration** and **Object** services that are published in Kony Fabric.

The **Logging** section in the **App Services Console** allows you to customize the type logs generated on the server. You can customize the generated logs by modifying few logging settings in the User Interface (UI). Kony Fabric will then record only logs pertaining to your settings.

To customize server logs, follow these steps:

1. Under the **Environment/Clouds** tab, click the **App Services** icon.
2. Click **Settings** from the left pane of the **App Services** console.

The **Settings** page appears containing **Logging**, **Runtime Configuration**, and **Environment** details. The **Logging** page displays the settings of various server logs.

The screenshot shows the Kony Settings page with the 'Logging' tab selected. The 'Request and Response Trace Logs' section has two checkboxes: 'Trace All Client Request and Response' and 'Trace All Backend Request and Response', both of which are unchecked. Below this is a red warning message: 'Enabling full request and response trace logs will impact server performance. Do not enable trace for long periods of time or during high user traffic.' The 'Log Level by Class' section has a 'Root Logger' dropdown set to 'ERROR' and a text input for 'Type fully qualified class or package name'. The 'Log Level by Client Filter' section has a 'Select Parameter' dropdown, a text input for 'Enter filter value to add more conditions', and a 'Select Log Level' dropdown. A checkbox for 'Enable Log Level Override from Client' is checked.

3. Customize the settings to fit your requirement.

5.3 Sync Console

The logs section in Kony **Sync Console** shows the data synchronization logs between client devices and the enterprise data source.

To access the logs in **Sync Console**, follow these steps:

1. From the **Environment/Clouds** tab, click **Sync Services**.
2. On the **Sync Console Services** page, click **Logs** from the left pane.

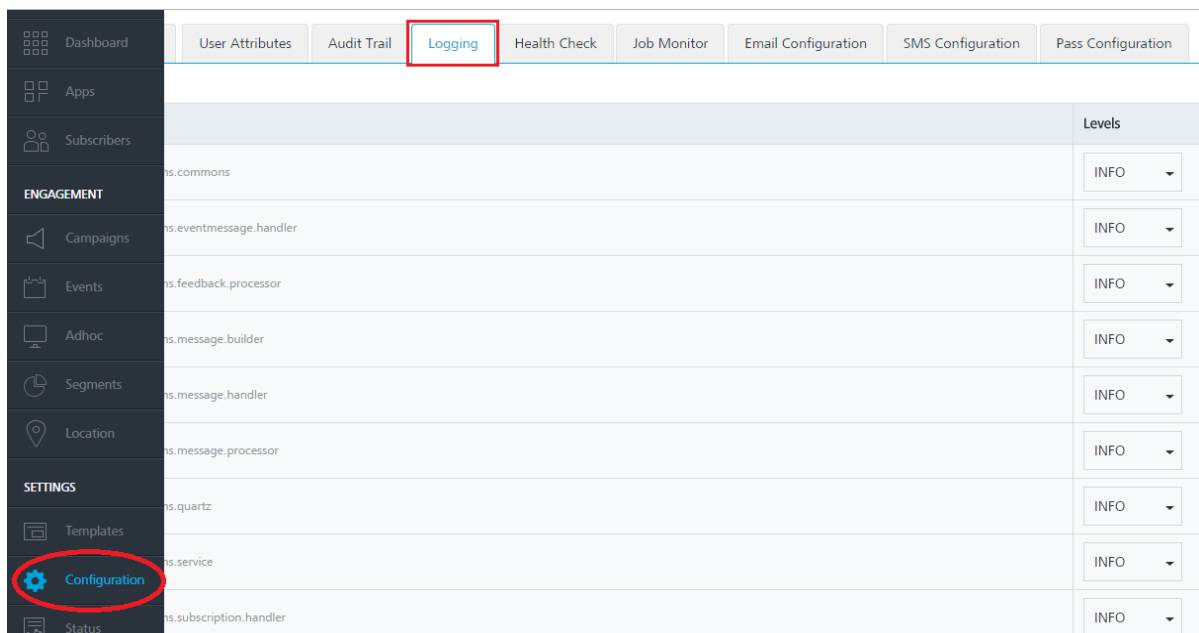
The screenshot shows the Kony Sync Services Trace Logs page. The left navigation pane has 'Logs' highlighted with a red box. The main area shows a table with columns: User ID, Device ID, Application ID, Request Type, Start Time, End Time, Elapsed Time (sec), DataSource Elapsed Time (sec), Http Request, Http Response, Request Size (Bytes), and Response Size (Bytes). The table is currently empty, and a message 'No records to show' is displayed below the table.

5.4 Engagement Console

Kony **Engagement Console** helps the administrator list out all Engagement services and related actions initiated by their registered users.

To view the logs in Kony **Engagement Console**, follow these steps:

1. From the **Environment/Clouds** tab, click **Engagement Services**.
2. In the **Engagement Console**, go to **Settings/Configuration**.
3. On the **Configuration** screen, click the **Logging** tab to view and control **Engagement Server** logs.

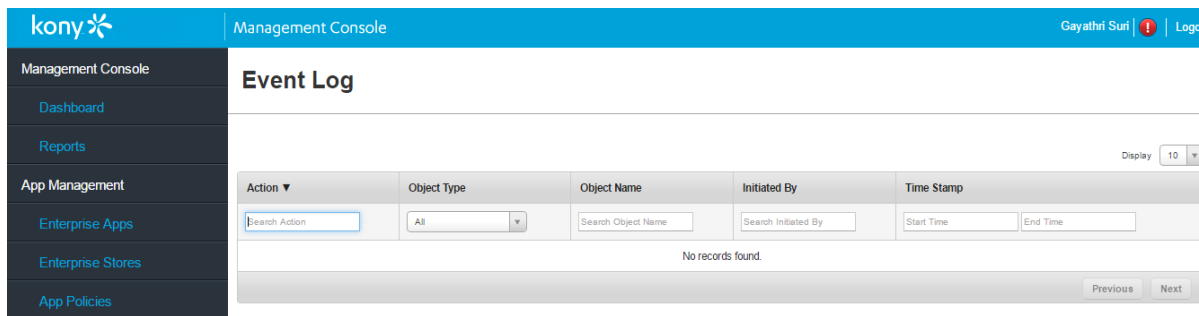


5.5 Management Console

Kony **Management Console** displays the list of actions of a device, device set, or policy that you initiated.

To access the logs section in **Management Console**, follow these steps:

1. From the **Environment/Clouds** tab, click **Management Services** from the list of consoles displayed.
2. On the **Management Console** screen, under the **Settings** section in the left pane, click **Event Log** to view the **Management Console** logs.



Let's take the case of pulling out logs for the **Books API** in detail from the **App Services Console**.

5.6 Testing a Service from Admin Console

As we saw in previous parts of this course, it is possible for us to test a service from the Kony Fabric Console. We can also test the API after publishing it from the **App Services Console**. This lets you debug your live API, to find the **Root Cause** for the failure quickly.

Note: You can test the service in **App Services Console** only after publishing the service in Kony Fabric environment.

To try out services from the **App Services Console**, follow these steps:

1. From the **App Services** console, click the **Integration** tab from the left pane.
You will be able to view the list of Integration services created and published in Kony Fabric.
2. Search for your service name. In our case, we used the **Books API** service.
3. Select an operation from the **Operations** List.

- The **Request Input** window displays the server URL along with **Header** and **Body** parameters. You can modify parameter values, and hit the **Get Response** button to try out the AI for different combinations.

Here's what this interface looks like :

The screenshot shows the 'Integration Services' interface for a service named 'NYTimes > GET_lists-overview-format'. The 'Request Input' tab is active, displaying a POST request to the URL 'https://sampletestcloud.qa-konycloud.com:443/services/NYTimes/GET_lists-overview-format'. The 'Body' tab is selected, showing a table of parameters:

Param Name	Param Value
format	json
published_date	2017-01-29
api-key	a3fccf5ac2d4fa397b8e9eff116ce24

Below this, the 'Response Headers' and 'Response Body' sections are visible. The 'Response Body' contains a JSON object:

```
{
  "copyright": "Copyright (c) 2017 The New York Times Company. All Rights Reserved.",
  "opstatus": 0, "results":
  {
    "next_published_date": "2017-02-05",
    "bestsellers_date": "2017-01-14",
    "published_date_description": "",
    "lists": [
      {
        "books": [
```

Hope you found this tutorial useful, please reach out to us for any queries on [basecamp](#).