

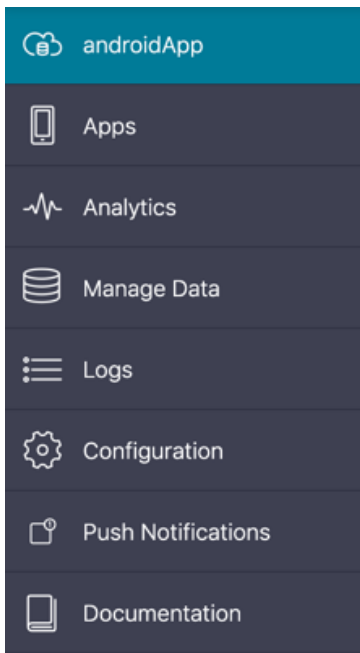
# Managing Mobile Backend Services Datasources

To view service details and manage data properties for a Mobile Backend Services (MBS) datasource, select it from the Dashboard home page **Apps** tab.

- [Mobile Backend Services app left navigation](#)
- [Create a New Datasource](#)
- [Change the Name of the Datasource](#)
- [Delete the Datasource](#)
- [View API Usage Data](#)
- [View Connected Applications](#)
- [View API Analytics](#)
- [Manage Data](#)
- [View and Filter API Logs](#)
- [Configuration](#)
- [Manage Push Notifications](#)
  - [Push Notification Logs](#)
  - [Push Notification Settings](#)
    - [iOS Push Notification Settings](#)
    - [Android Cloud Messaging Settings](#)
  - [View Subscribed Devices](#)
  - [View Scheduled Notifications](#)
- [View Documentation](#)

## Mobile Backend Services app left navigation

The left navigation includes the following tabs:



- [App overview](#)
- [Apps](#)
- [Analytics](#)
- [Manage Data](#)
- [Logs](#)
- [Configuration](#)
- [Push Notifications](#)
- [Documentation](#)

## Create a New Datasource

To create a standalone Mobile Backend Services datasource (an MBS datasource that is not tied to a specific client application), refer to the [Getting Started with Mobile Backend Services Datasources](#).

## Change the Name of the Datasource

To change the name of the datasource:

1. Log into the [AMPLIFY Platform](#).
2. Click Dashboard from the *Service Links* menu.
3. Select the datasource from the apps list.
4. In the datasource details page, click the **Actions (...)** menu in the upper-right of the screen and select **Edit**.
5. Enter a new name for the datasource.

Form fields:

- \* Name** (with help icon): API\_Builder\_Application\_Two
- Description**: API Builder Application

6. (Optional) Enter a datasource description.
7. Click **OK**.

## Delete the Datasource

To delete the datasource:

1. Click the **Actions (...)** menu.
2. Select **Delete**.
3. Confirm you want to delete the datasource. To permanently delete the datasource, type the name of the data store (datasource) you would like to permanently delete and select, **I understand that this a permanent and irreversible action. Continue**. Click the **Cancel** button to return the Overview tab without deleting the data store.

## View API Usage Data

The **Overview** tab displays the Mobile Backend Services Usage data and includes the number of Mobile Backend Services Requests made to MBS, the number of devices connected to receive [push notifications](#), the number of email templates available to the application, and the amount of disk storage used. The **Overview** tab also displays the App GUID, Description, Teams, Created, and Creator information for the selected MBS application.

**App GUID:**

5ab187f205955358842fec5d

**Created:**

Mar 20, 2018 3:15 PM

**Description:**

API Builder Application

**Creator:**

[Blurred]

**Teams:**

Team 1



## Mobile Backend Services Usage

**Month-to-Date** | Last Month

### Mobile Backend Services Requests <sup>?</sup>

Access Control Lists	0	Chats	0	Check-Ins	0	Clients	0
Custom Objects	0	Email Templates	0	Events	0	Files	0
Friends	0	Key Values	0	Messages	0	Photo Collections	0
Photos	0	Places	0	Posts	0	Push Notifications	0
Ratings, Reviews & Likes	0	Social Integration	0	Status	0	Users	0

### Push

 Connected iOS devices **0**    Connected Android devices **0**

### Storage <sup>?</sup>

Database **0 B**   File **0 B**

### Email

Templates **0**



## View Connected Applications

The **Apps** tab displays a list of applications (Titanium or native) that are consuming the selected Mobile Backend Service app. Select an application from the list to manage it.

# API\_Builder\_Application\_Two Development ▾

## Apps

Name	Type	Created
API_Builder_Application_Two	AMPLIFY Runtime	Mar 20, 2018

## View API Analytics

The **Analytics** tab displays the total number of API calls for the selected period by API type (Chats or Users APIs, for instance) or by individual application. For additional information, see [Viewing Analytics](#).

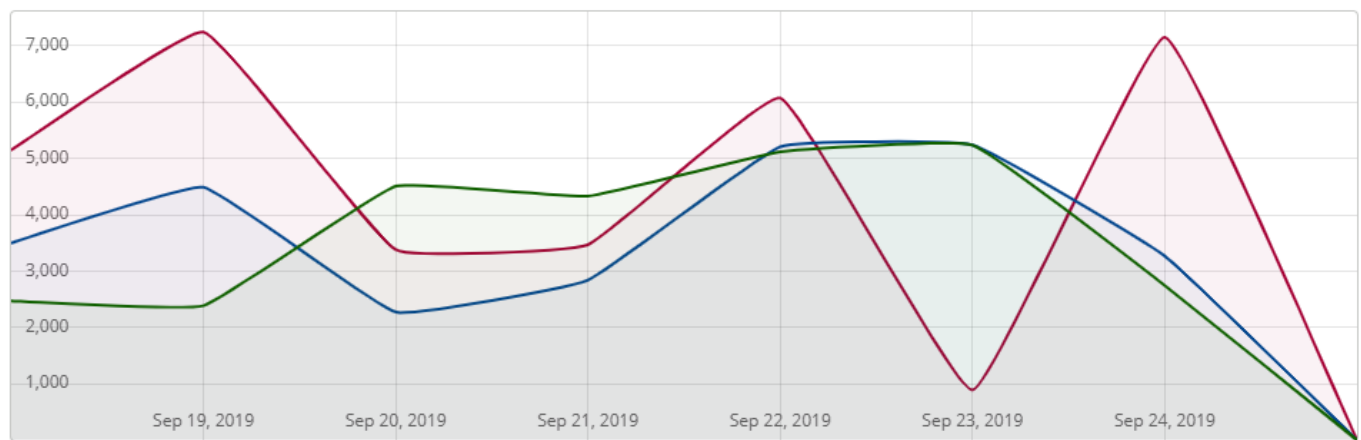
### API\_Builder\_Application\_Two ⋮

Last 7 Days ▾ Production ▾

86,978 Requests

Application | Type

Scale to the current page



Application	Requests	Daily Average	Top Request
Awesome App 2	33,359	4,170	users
Awesome App	26,823	3,353	users
Example App 1	26,796	3,350	users

## Manage Data

The **Manage Data** tab displays the number of each type of MBS object created by the selected application. Click the MBS type in the table to view, edit, create and delete objects (see [Managing Data Objects](#)).

To export your application's MBS data in a JSON format, click the **Actions** (...) menu icon and select **Export Data**. To delete your application's data, click the **Actions** (...) menu icon and select **Delete Data**.

## Manage Data Objects

Chats	0	Files	0	Places	0
Check Ins	0	Geo Fences	0	Posts	0
Photo Collections	0	Key Values	0	Reviews	0
Custom Objects	0	Messages	0	Statuses	0
Events	0	Photos	0	Users	10

## Administration

Access Control	1	Email Templates	0
----------------	---	-----------------	---

## View and Filter API Logs

The **Logs** tab displays a paginated table of all API calls made by your application to the selected MBS service.

## API Request Logs



	Method	Path	Status	Date
+	POST	/v1/users/login.json	200	Feb 19, 2019 2:16 PM
+	POST	/v1/users/login.json	200	Feb 19, 2019 2:16 PM

The table includes the following fields:

- **Method** – HTTP method used.
- **Path** – Path of the HTTP request.
- **Status** – HTTP status of the request.
- **Date** – Date and time of the request in UTC.

Clicking a table row expands it to display the following additional fields:

- **Source IP** – IP address of the client that made the request.
- **Parameters** – Click **Show** to view a list of request parameters, and the full URL request.

You can filter MBS API requests by their date.

## API\_Builder\_Application\_Two Development ▾

### API Request Logs



Date:  ▾ Less than:  ▾ Date:  Time:  ▾

Method	Path	Status
+ POST	/v1/users/login.json	200
+ POST	/v1/users/login.json	200

Calendar: May 2019

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

## Configuration

On the **Configuration** tab you can select from the following sub-tabs:

- **Keys** – View your API Builder application key and OAuth consumer key and OAuth secret.
- **Settings** – Change API Builder settings such as enable or disable new user email verification, friend request type (two-way or one-way), and whether users can be created from the application or only from Dashboard.
- **Email** – Provide SMTP or MessageGears server settings for Dashboard to send an email on behalf of your application. Additionally, you can enable or disable the email test mode.

For details, see [Configuring Cloud Services](#).

## API\_Builder\_Application\_Two Production ▾

### Keys

App Key	Show
OAuth Consumer Key	Show
OAuth Secret	Show

## Manage Push Notifications

The **Push Notifications** tab lets you select from the following sub-tabs:

- **Send** – See [Sending and Scheduling Push Notifications](#) for details.
- **Logs** – Paginated and filterable table of push notification log items. See [Push Notification Logs](#).
- **iOS** – Configure API Builder Push to communicate with the Apple Push Notification Service (APNS). See [iOS and Android Push Notification Settings](#) for details.
- **Android** – Configure Firebase Cloud Messaging service for Android (FCM) on behalf of your application. [Firebase Cloud Messaging \(FCM\)](#) is the new version of GCM. See [iOS and Android Push Notification Settings](#) for details.

- **Devices** – View a list of devices subscribed to receive push notifications. See [View Subscribed Devices](#).
- **Scheduled** – View a list of scheduled pushes.

## Push Notification Logs

Click the **Logs** tab to view a paginated table of push notifications delivered by Dashboard, and the status of the notification. Each row contains the following fields:

- **Payload** – The JSON notification payload.
- **Channel** – The name of the subscription channel to which the notification was delivered.
- **Processed** – The time and date the push notification was processed.
- **Notification ID** – ID of the delivered notification.
- **iOS** – The number of iOS devices that received the push.
- **Android** – The number of Android devices that received the push.
- **Devices** – The total number of devices that received the push.
- **Status** – The status of the notification (Success or Fail).
- **Sent** – Time and date the notification was sent.

You can filter the listing of push notifications by channel name or creation date.

## Push Notification Settings

The **iOS** and **Android** tabs are where you configure Dashboard to use the APNS or either the FCM or GCM push notification services. For details, see [Configuring Push Services](#).

### iOS Push Notification Settings

Click the **iOS** tab to configure the application to use the Apple Push Notification service. Before configuring APN, you need to create a Push Notification Certificate and export it as a PKCS #8 or #12 file. For instructions, see [Configuring push services for iOS devices](#).

*To configure the p8 format*

1. Click **Browse** in the Push Certificate field.
2. Locate the PKCS #8 file with your Push Notification certificate and click **Choose File**.
3. Select p8 for the **Certificate Format**.
4. Complete the following fields.
  - **Key ID** - is a 10 character string (for example, TJ9953DRAJ) and can be found at <https://developer.apple.com/account/resources/authkeys/review>.
  - **Team ID** - is a 10 character string (for example, JKVJBKSZOR) and can be found at <https://developer.apple.com/account/#/membership>.
5. (Optional) Enter the bundle ID in the **Bundle ID** field.
6. Click **Save**.

Apple iOS Push Configuration

[Documentation](#)

Save

* Push Certificate	Choose File	Browse
Certificate Format	<input checked="" type="radio"/> p8	<input type="radio"/> p12
* Key ID	<input type="text"/>	
* Team ID	<input type="text"/>	
Bundle ID	<input type="text"/>	

*To configure the p12 format*

1. Click **Browse** in the Push Certificate field.
2. Locate the PKCS #12 file with your Push Notification certificate and click **Choose File**.
3. Enter the certificate's password in the **Certificate Password** field.
4. Select p12 for the **Certificate Format**.
5. (Optional) Enter the bundle ID in the **Bundle ID** field.
6. Click **Save**.

\* Push Certificate

\* Certificate Password

Certificate Format  p8  p12

Bundle ID

## Android Cloud Messaging Settings

Click the **Android** tab to configure the application to use the Firebase Cloud Messaging (FCM) service. [Firebase Cloud Messaging \(FCM\)](#) is the new version of GCM. FCM HTTPv1 and FCM legacy HTTP is supported. Before configuring FCM legacy HTTP, you need to obtain an FCM or GCM sender ID and Google API server key. For instructions, see [Configuring push services for Android devices](#).

To configure FCM HTTPv1, select a service account JSON file from your Firebase console Service Accounts page.

### Android Cloud Messaging Configuration

Provider  FCM HTTP v1  FCM Legacy HTTP

Service Account JSON ⓘ

To configure FCM Legacy HTTP

1. Enter the Google API server key in the **Server Key** field.
2. Enter the FCM or GCM sender ID in the **Sender ID** field.
3. Click **Save**.

### Android Cloud Messaging Configuration

Provider  FCM HTTP v1  FCM Legacy HTTP

\* Server key

\* Sender ID

## View Subscribed Devices

Click the **Devices** tab to view a list of subscribed users, devices, and channel subscriptions. Each list row contains the following columns:

- **Channel** – The notification channel(s) the device is subscribed to.
- **Type** – The subscription's push notification type (**Android** or **iOS**).
- **User** – The full name of the API Builder user subscribed to the channel. For token-based subscriptions, this field is empty.
- **Device Token** – The device token obtained from the Firebase Cloud Messaging or Apple Push Notification service. Click the expander (+) button to view the complete device token.

You can filter the list of subscribed devices by channel, type, or user ID. To filter the list:

1. Click the **Filter** button.



2. In the first drop-down, select either **User ID**, **Type**, or **Channel** to filter the list according to that criteria.
3. You can only select **Equals** in the second drop-down.
4. In the last field, enter the value you want to filter by:
  - a. For **User ID**, enter the object ID of the User.
  - b. For **Type**, either enter `android` or `ios`.
  - c. For **Channel**, enter the name of the channel.
5. Click **Apply Filters**.

To remove the filter, click **Clear Filter**.

## View Scheduled Notifications

Click the **Scheduled** tab to display a list of one-time or recurring scheduled notifications. Initially, the list displays the following fields:

- **Name** – Schedule name.
- **Recurring** – A Boolean that indicates if the schedule is recurring (Yes) or one-time (No).
- **Payload** – >The JSON payload to deliver with the notification.
- **Start Time** – The date and time (in UTC) that the notifications are scheduled to start being delivered.

Click the expander button (+) next a list item to view details about the schedule, or to edit or delete the schedule. The details view contains the following fields:

- **Channel** – Name of the channel the notification will be sent.
- **To Users** – The API Builder Users to which the message will be sent.
- **Starts On/Ends On** – Start and end date and time the scheduled starts and expires.
- **Recurs** – How often the schedule repeats. Possible values are Daily, Weekly, or Monthly.

## View Documentation

The **Documentation** tab displays a list of links to the Mobile Backend Services guides and other resources.