

Mobile Backend Services Getting Started

- [Create a Mobile Backend Services datasource](#)
 - [Standalone Mobile Backend Services datasource](#)
 - [Mobile Backend Services with Titanium](#)
 - [Mobile Backend Services with Android or iOS](#)
- [Make calls to Mobile Backend Services](#)
- [Next steps](#)

Mobile Backend Services (MBS) provides a set of REST APIs for creating, managing, and accessing different types of data in your cloud datasource, such as [Users](#), [Places](#), and [Photos](#) over HTTP or HTTPS. You can integrate MBS into your application using the [Titanium](#), [iOS](#), [Android](#) or [Node.js](#) SDKs, or by calling the [REST APIs](#) directly.

To manage your application and its data—for example, to create or edit [Users](#) or manage [Photos](#)—you use [Dashboard](#).

This guide explains how to create a standalone MBS datasource and make API calls to the datasource using the REST APIs. For integrating MBS with a specific platform, see the following SDK guides:

- [Titanium SDK](#)
- [Android SDK](#)
- [iOS SDK](#)
- [Node.js SDK](#)

Create a Mobile Backend Services datasource

You can create either a standalone MBS datasource or create an MBS datasource associated with a Titanium, Android or iOS application.

Standalone Mobile Backend Services datasource

A standalone MBS datasource does not have a specific client application associated with it. Use a standalone MBS source if you want multiple applications to access the same datasource, or if your client application is not Titanium, Android or iOS.

1. Log into the [AMPLIFY Platform](#).
2. Select **Dashboard** on the Dashboard tile.
3. Click the Add menu (+) in the top navigation bar.

Dashboard navigation: Home, DASHBOARD, +

App Actions: Register App for Services, Create Mobile Backend Services Datasource, Add Org Member

Search: Search by app name, ID, or create

Name	Type	Created
API_Builder_Application_Two	Mobile Backend Services	Mar 20, 2018 3:15 PM
API_Builder_Application_Two	Runtime Services	Mar 20, 2018 3:15 PM
API_Builder_Test	Android	Feb 26, 2018 1:14 PM
Apple	iOS	Mar 23, 2018 11:15 AM
Performance_Test	Titanium SDK	Mar 8, 2018 2:41 PM
SDK Services	Android	Sep 28, 2018 10:36 AM
Test SDK Services	Mobile Backend Services	Sep 28, 2018 10:34 AM
203	Mobile Backend Services	Oct 24, 2018 3:12 PM
203	Runtime Services	Oct 24, 2018 3:12 PM
API	Windows	Dec 10, 2018 9:26 AM

Page navigation: 1 2 3 4 5 6 7 8 9 | 10 per page

4. Select **Create Mobile Backend Services Datasource**.

Create Mobile Backend Services Datasource

* Name

Assign Teams

Name	Members	Apps	Select
Team 1 DEFAULT	2	43	×
Test Team	3	5	+

5. Enter a name for the datasource. This will create a Mobile Backend Services datasource for each available environment.

6. Assign teams to the application by selecting the + icons from the Assign Teams list.
7. If you have multiple environments, select the environments to enable for the datasource.
8. Click **Create**.

To make calls to MBS, you will need your MBS applications key. After Dashboard creates the datasource, click **Configuration** in the left navigation and select the **Keys** tab, then click the **Show** link next to **App Key**. Use the MBS application key to make requests to MBS. Note that you have a key for each deployment environment.

The screenshot shows the configuration page for 'API_Builder_Application_Two' in the 'Production' environment. The left sidebar has a 'Keys' tab selected. The main content area lists three keys:

Key Name	Action	Value
App Key	Hide	1TfFefl6bXFMpI0J7faNMRVJ9WIO9k1
OAuth Consumer Key	Show	
OAuth Secret	Show	

Mobile Backend Services with Titanium

Use Studio or the CLI to enable platform services and create an MBS datasource associated with a Titanium application. After creating the application, load the `ti.cloud` module to make requests to MBS.

For directions, see the [Titanium SDK Guide](#).

Mobile Backend Services with Android or iOS

Use Dashboard to register an Android application built with Java or iOS application built with Objective-C or Swift. The registration process creates a new MBS datasource associated with the application. Then, use the APS SDKs to integrate MBS services with the application.

For directions, see:

- [Android SDK](#)
- [iOS SDK](#)

Make calls to Mobile Backend Services

You can make calls to MBS using the following SDKs and modules, or by making HTTP requests directly to MBS.

- [Titanium Cloud Module](#)
- [AMPLIFY Appcelerator Services API Reference for Android](#)
- [AMPLIFY Appcelerator Services API Reference for iOS](#)

- [Node.js SDK](#)
- [Mobile Backend Services API docs](#)

To make MBS calls from other applications, you need to use the platform's native HTTP client to make HTTP requests directly. You will need to pass the MBS application key in the URL as the `key` parameter with each request.

For example, the following Ruby code uses the `Net::HTTP` library to make an MBS request:

```
require 'net/http'
require 'json'

base_url = 'https://api.cloud.appcelerator.com/v1/'
key_param = 'key=<APP_KEY>'
url = URI(base_url + 'users/create.json?' + key_param)
req = Net::HTTP::Post.new(url)
req.set_form_data(:username => 'user1', :password => 'pass1', :password_confirmation => 'pass1')
res = Net::HTTP.start(url.host, url.port, :use_ssl => true) do |http|
  http.request(req)
end
response = JSON.parse(res.body)
puts "You are now logged in as " + response["response"]["users"][0]["username"]
```

For requests that require a user to be logged in, you will need to retrieve the `session_id` from the meta header of the response from either the `users/login.json` or `users/create.json` method, then pass the `session_id` in the URL as the `_session_id` parameter with the request.

```
session_id_param = '_session_id=' + response["meta"]["session_id"]
url = URI(base_url + 'posts/create.json?' + key_param + '&' + session_id_param)
req = Net::HTTP::Post.new(url)
req.set_form_data(:content => 'Calling ArrowDB from Ruby')
res = Net::HTTP.start(url.host, url.port, :use_ssl => true) do |http|
  http.request(req)
end
puts res.body
```

The SDKs and modules provided by Appcelerator abstract the HTTP request and will automatically handle passing the application key and session ID between the client application and MBS datasource. For example, the following is an equivalent call using the Titanium Cloud module:

```
var Cloud = require('ti.cloud');
Cloud.Users.create({
  username: 'user1',
  password: 'pass1',
  password_confirmation: 'pass1'
}, function (e) {
  if (e.success) {
    alert('You are now logged in as ' + e.users[0].username);
  } else {
    alert('Error:\n' + ((e.error && e.message) || JSON.stringify(e)));
  }
});
Cloud.Posts.create({
  content: 'Calling ArrowDB from Titanium'
}, function (e) {
  if (e.success) {
    alert('Post succeeded!');
  } else {
    alert('Error:\n' + ((e.error && e.message) || JSON.stringify(e)));
  }
});
```

Next steps

Review the [REST Guide](#) if you are making HTTP requests directly to MBS, and the [Mobile Backend Services API reference](#), which contains Titanium, REST, Android, iOS and Node.js examples. To set up a specific type of client application, see the following SDK guides:

- [Titanium SDK](#)
- [Android SDK](#)
- [iOS SDK](#)
- [Node.js SDK](#)