MERKLE

Marketing automation tech, news & insights | Merkle

UPDATED ON 18.05.2021 BLOG

Journey Builder Custom Activity Deep Dive

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

Introduction

In the previous article, we have reviewed various types of custom components that exist in Marketing Cloud. Today we will describe in detail the development of the first and most popular component – Journey Builder Activity.

At the end of the previous article, we talked about the steps that must be followed before we start developing. At this point, we will assume that you have successfully created an installed package and registered an account on Heroku.

Next, we'll go through the source code of our custom Journey Builder Activity, go through the steps required to deploy and configure the application on Heroku, then connect our application to Marketing Cloud and test it using Journey Builder and logs.

Use Case

For this article, I decided to pick a primitive use case since the main focus will be rather on describing the steps needed to build, deploy and configure the custom application in Salesforce Marketing Cloud.

So, let's assume that we need a custom activity that can be used to log journey entry/exit events for each contact along with some comment into the Data Extension.

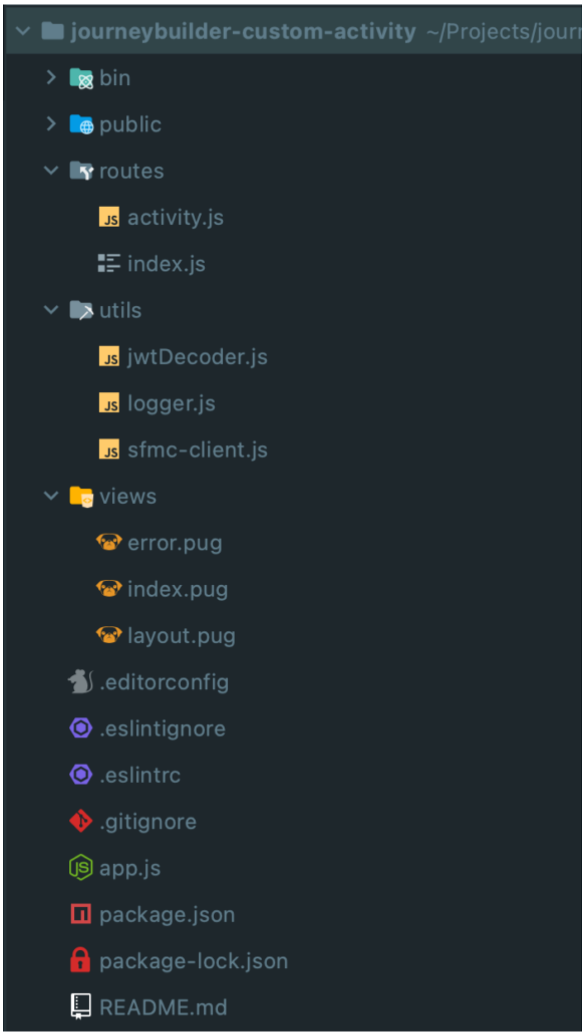
Source Code

We have prepared the repository which you can fork to start working on your own custom activities: https://github.com/sfm-cz/journeybuilder-custom-activity

Now, let's figure out the structure of this component. The component is a NodeJS application using the Express.js framework. Pug, formerly known as Jade, is used as a template engine.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy



```
journeybuilder-custom-activity ~/Projects/jourr
  bin
  public
  routes
    activity.js
    index.js
  utils
    jwtDecoder.js
    logger.js
    sfmc-client.js
  views
    error.pug
    index.pug
    layout.pug
  .editorconfig
  .eslintignore
  .eslintrc
  .gitignore
  app.js
  package.json
  package-lock.json
  README.md
```

App.js is the entry file of the application where all the necessary routes and libraries are connected and the application itself is configured. We tried to comment on the code at greater length in order to make it clearer and easier for you to start.

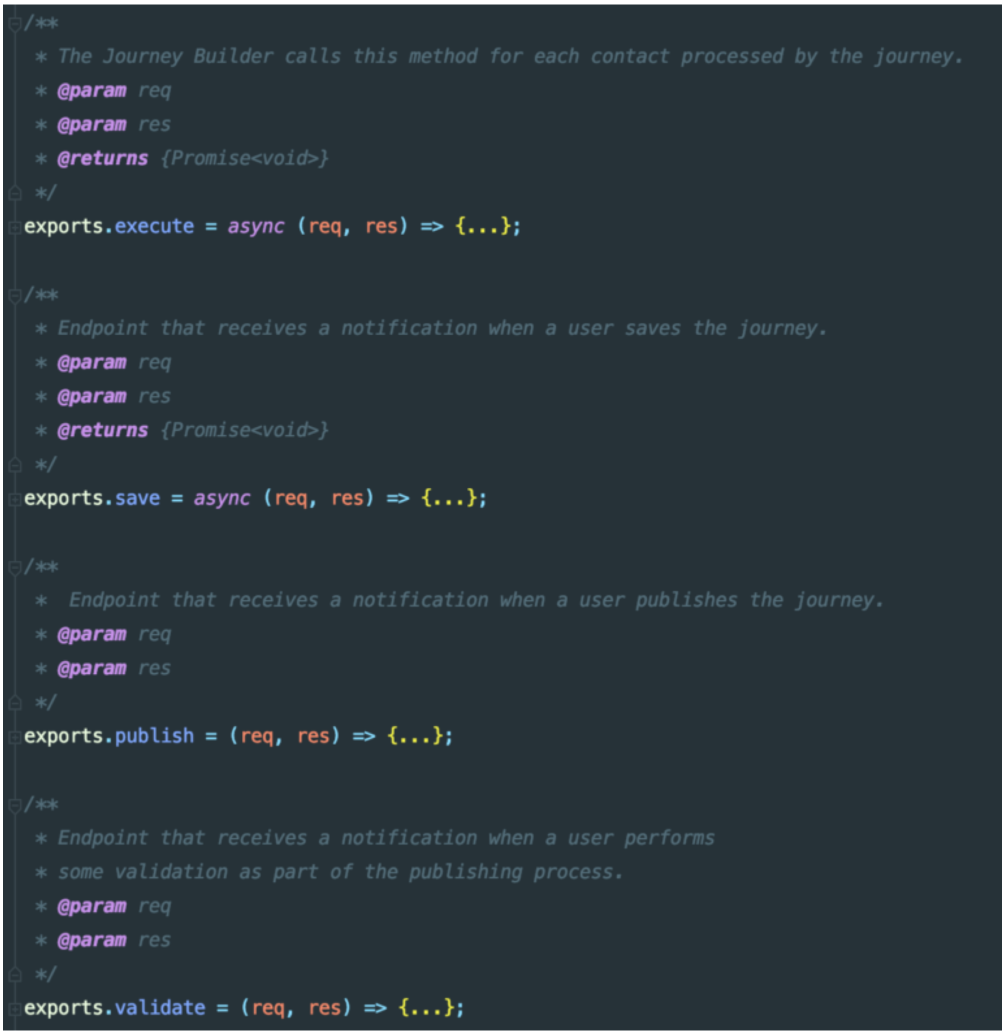
All the necessary routes are stored in the routes folder:

- Index.js contains the UI of the custom activity
- Activity.js contains the minimum required endpoints for a custom activity, namely:
  - Execute is invoked when a person enters a custom activity
  - Save is invoked when Journey is activated
  - Publish is invoked when Journey is validated successfully
  - Validate is invoked before Save and by clicking Validate in the interface of Journey Builder

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

using JWT.

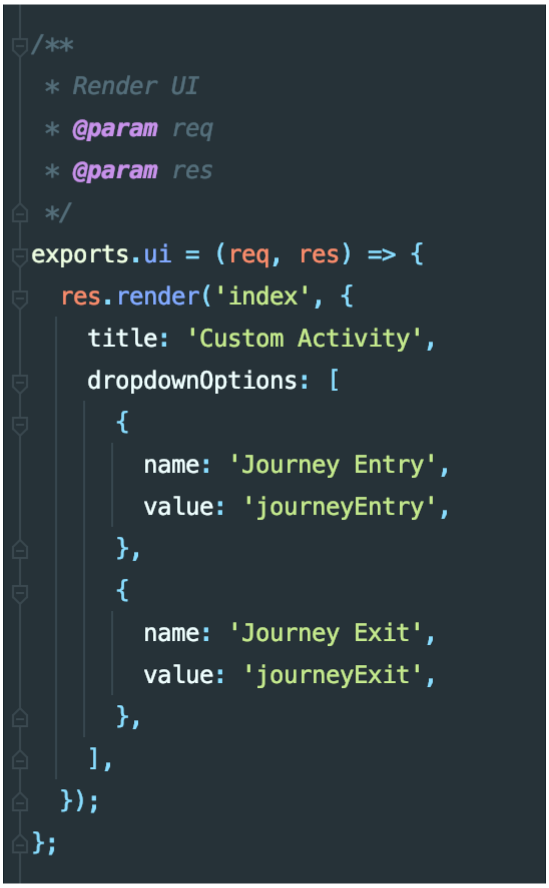


```
/**
 * The Journey Builder calls this method for each contact processed by the journey.
 * @param req
 * @param res
 * @returns {Promise<void>}
 */
exports.execute = async (req, res) => {...};

/**
 * Endpoint that receives a notification when a user saves the journey.
 * @param req
 * @param res
 * @returns {Promise<void>}
 */
exports.save = async (req, res) => {...};

/**
 *  Endpoint that receives a notification when a user publishes the journey.
 * @param req
 * @param res
 */
exports.publish = (req, res) => {...};

/**
 * Endpoint that receives a notification when a user performs
 * some validation as part of the publishing process.
 * @param req
 * @param res
 */
exports.validate = (req, res) => {...};
```

The Views folder contains a template for rendering Custom Activity UI. We prefer to use the template engine instead of plain HTML because it allows us to get dynamic data and transfer it to the template. For instance, we can load some data from the Data Extension and use it as a value for the Dropdown list.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy



```
/**
 * Render UI
 * @param req
 * @param res
 */
exports.ui = (req, res) => {
  res.render('index', {
    title: 'Custom Activity',
    dropdownOptions: [
      {
        name: 'Journey Entry',
        value: 'journeyEntry',
      },
      {
        name: 'Journey Exit',
        value: 'journeyExit',
      },
    ],
  });
};
```

The Assets folder contains static files: pictures, styles, JavaScript.

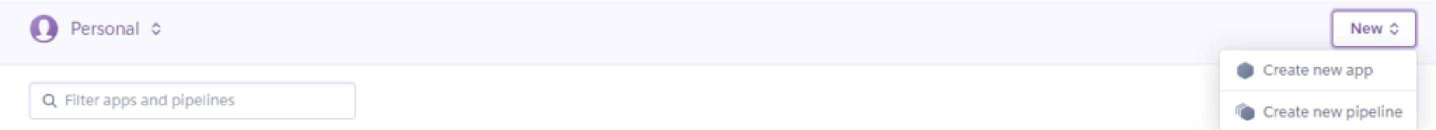
I recommend paying attention to customActivity.js, which implements the ability to save and read settings from Marketing Cloud using Postmonger.

Also at the root of the project, you will see an .eslintrc file that helps us write code within a command in the identical style and using Eslint.

Heroku – Deployment and Configuration

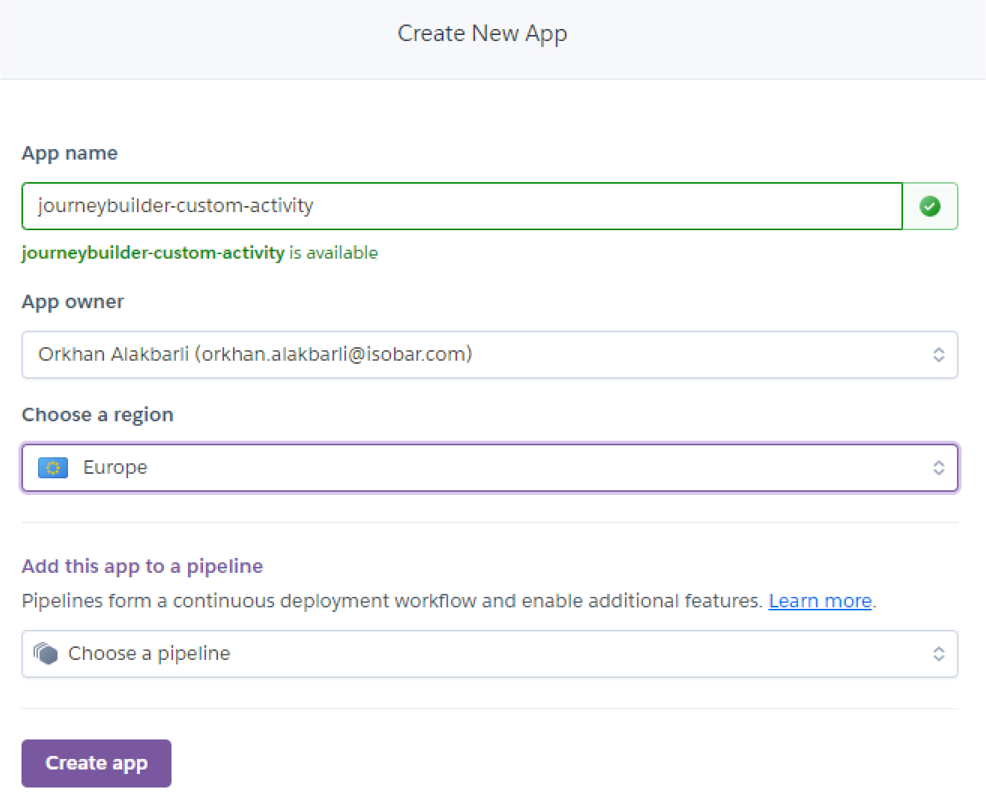
When your code is ready for deployment, you can proceed with the steps below:

1. Go to your applications page on Heroku and in the upper right corner click New – Create new app.



We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy



Create New App

App name

journeybuilder-custom-activity

journeybuilder-custom-activity is available

App owner

Orkhan Alakbarli (orkhan.alakbarli@isobar.com)

Choose a region

Europe

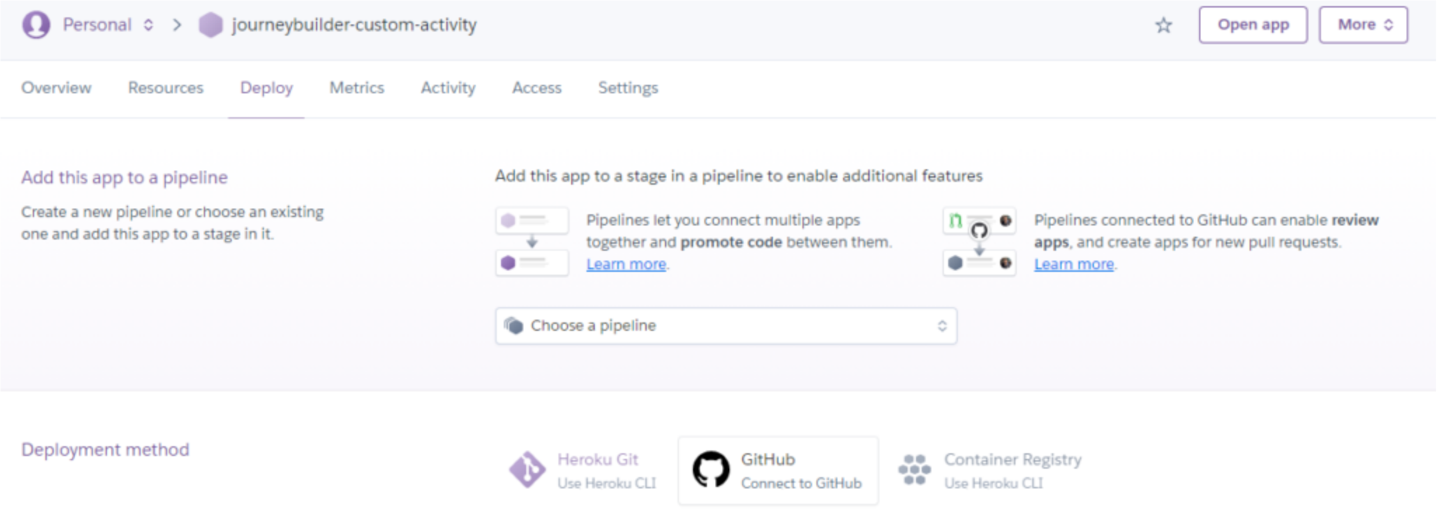
Add this app to a pipeline

Pipelines form a continuous deployment workflow and enable additional features. Learn more.

Choose a pipeline

Create app

3. Once you appear in the Deployment tab of your application, in the Deployment method section, select GitHub.



Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. Learn more.

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

Choose a pipeline

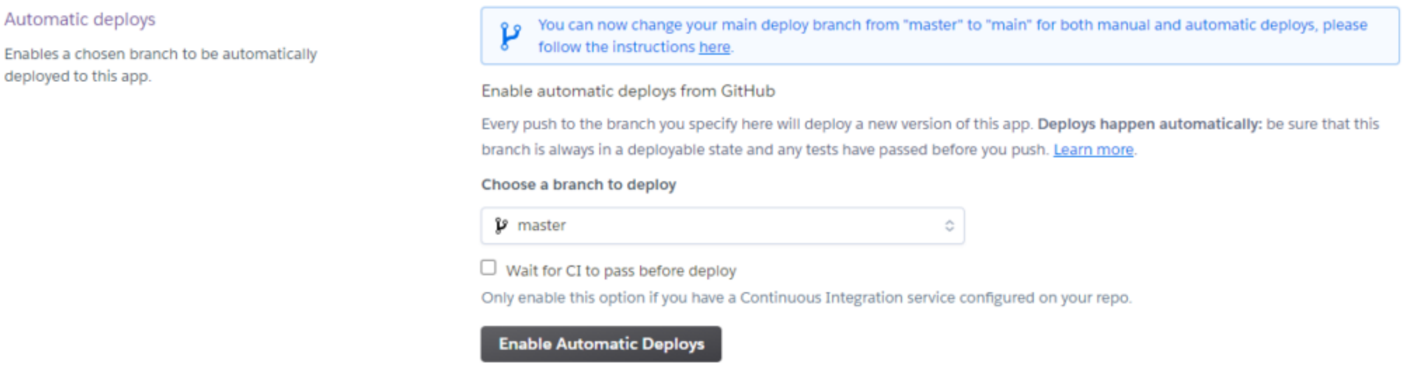
Deployment method

4. In the newly appeared Connect to GitHub section, select your GitHub account, and then enter the names of your repository and click Search. Then click Connect.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

5. Now that your application is ready for code deployment from your GitHub account, you can (optionally) enable automatic deployments. To do this, go to the Automatic deploys section, select your repository branch and click Enable Automatic Deploys.



Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. Learn more.

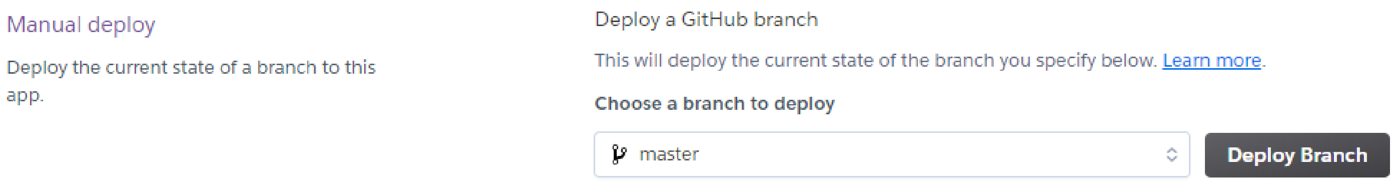
Choose a branch to deploy

master

□ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

6. You can also do manual deployment when needed. To do this, select the repository branch in the Manual deploy section and click Deploy Branch.



Manual deploy

Deploy the current state of a branch to this app.

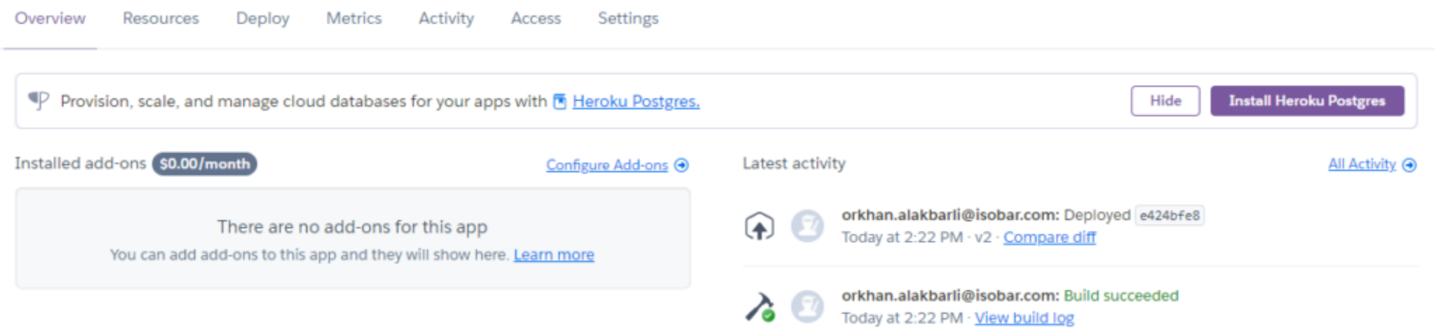
Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

master

7. Deploy your code and make sure that build succeeded. For that, navigate to the Overview tab and check the list of latest activities.



Provision, scale, and manage cloud databases for your apps with Heroku Postgres.

Installed add-ons $0.00/month

There are no add-ons for this app

You can add add-ons to this app and they will show here. Learn more

Latest activity

orkhan.alakbarli@isobar.com: Deployed e424bfe8
Today at 2:22 PM · v2 · Compare diff

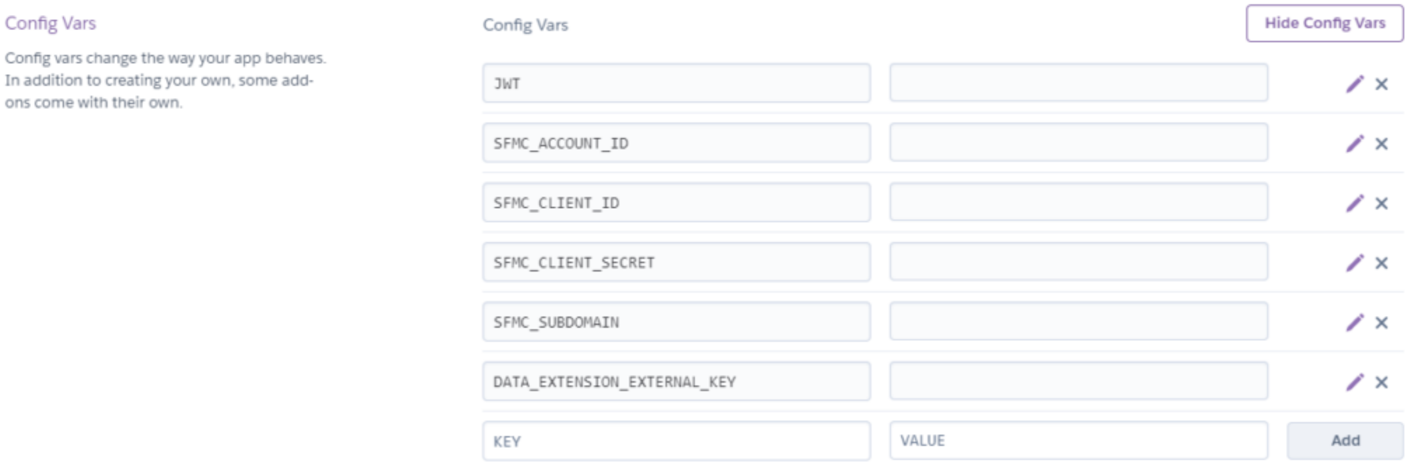
orkhan.alakbarli@isobar.com: Build succeeded
Today at 2:22 PM · View build log

8. Now we need to configure the Variables that are used in our application. Heroku's Config Vars is a way to avoid storing sensitive information such as IDs

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

and in the Config Vars section click Reveal Config Vars. Now you need to add the following variables:



Config Vars

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

```
JWT
SFMC_ACCOUNT_ID
SFMC_CLIENT_ID
SFMC_CLIENT_SECRET
SFMC_SUBDOMAIN
DATA_EXTENSION_EXTERNAL_KEY
```

Note: If you want to run your application locally before deploying it to Heroku you can store all your config vars in .env file. Don't forget to add .env to your .gitignore file to avoid pushing it to Git.

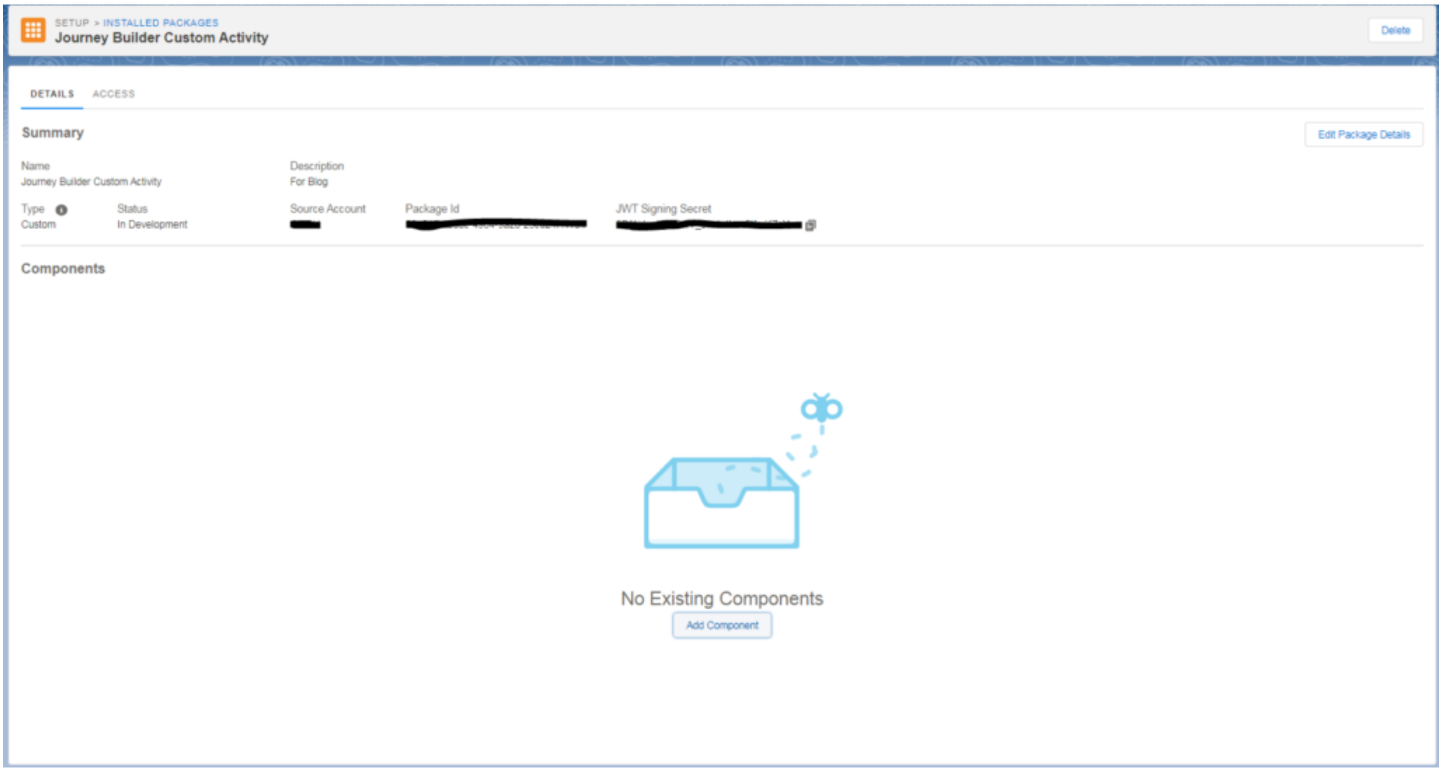
That's all for now. Although we haven't finished configuring the variables yet, now we need to set up the connection in Marketing Cloud. Don't worry, we'll come back to Heroku in a moment. Therefore, leave the current browser window open.

Marketing Cloud – Installed Package and DE Configuration

1. Log into the Marketing Cloud and navigate to Setup -> Apps -> Installed Packages, choose the Installed Package you have previously created and click Add Component.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy



SETUP > INSTALLED PACKAGES

Journey Builder Custom Activity

DETAILS ACCESS

Summary

Name: Journey Builder Custom Activity

Description: For Blog

Type: Custom

Status: In Development

Source Account: [redacted]

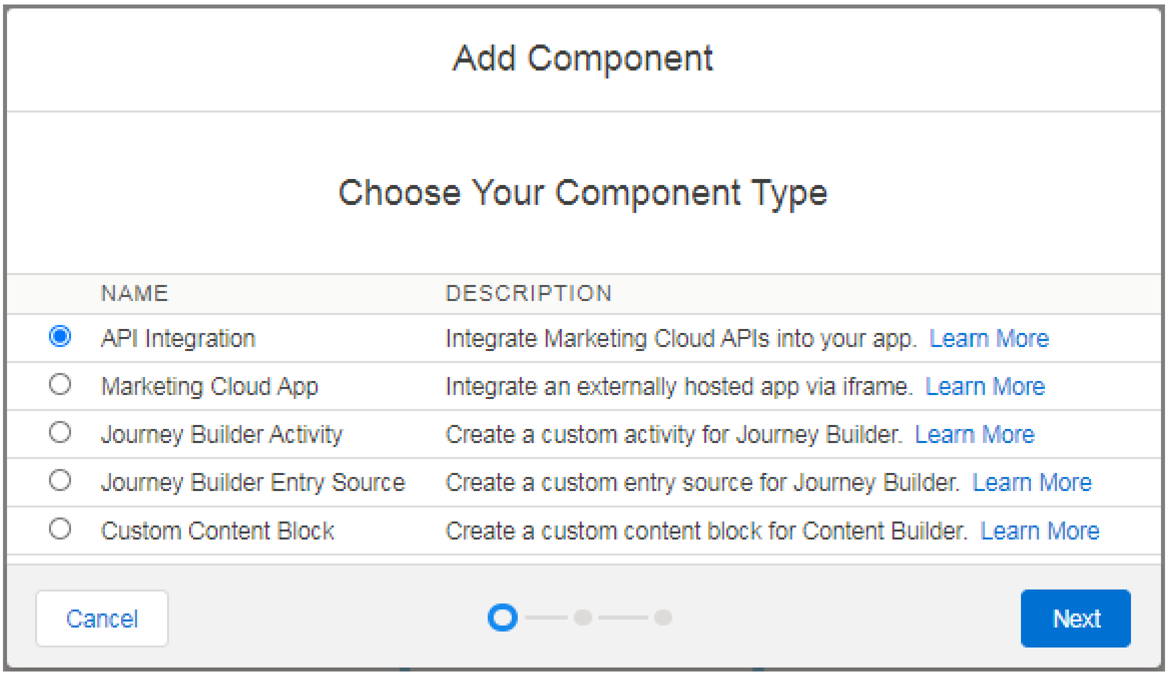
Package Id: [redacted]

JWT Signing Secret: [redacted]

Components

No Existing Components

2. Then choose API Integration and click Next.



Add Component

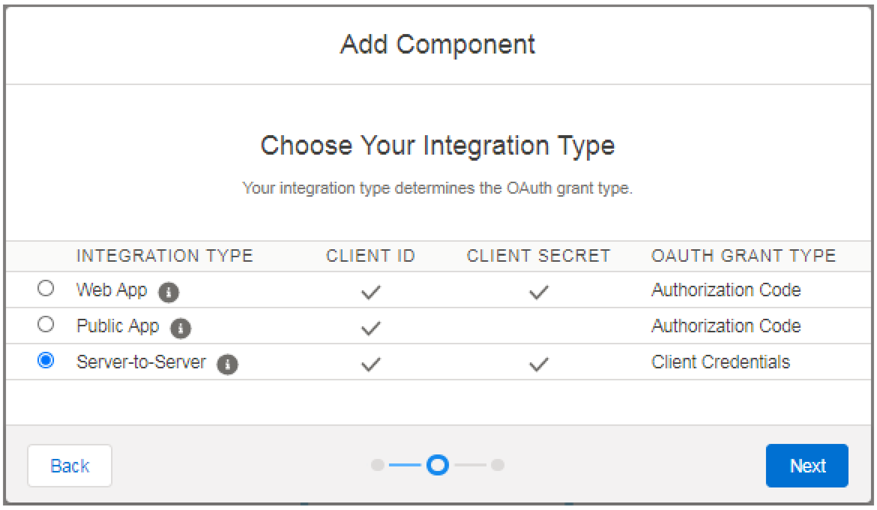
Choose Your Component Type

	NAME	DESCRIPTION
●	API Integration	Integrate Marketing Cloud APIs into your app. Learn More
○	Marketing Cloud App	Integrate an externally hosted app via iframe. Learn More
○	Journey Builder Activity	Create a custom activity for Journey Builder. Learn More
○	Journey Builder Entry Source	Create a custom entry source for Journey Builder. Learn More
○	Custom Content Block	Create a custom content block for Content Builder. Learn More

3. Choose Server-to-Server and click Next.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy



Add Component

Choose Your Integration Type

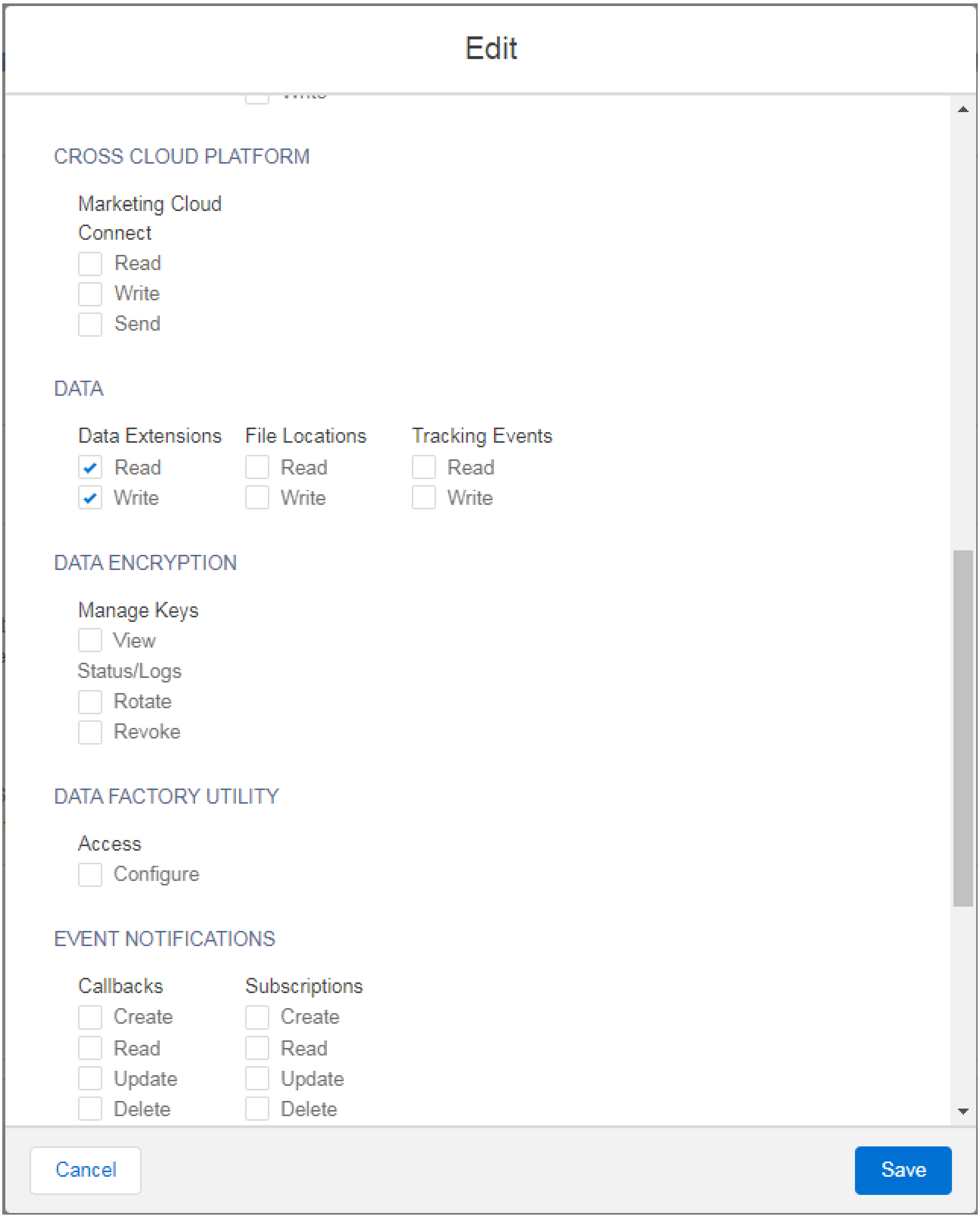
Your integration type determines the OAuth grant type.

	INTEGRATION TYPE	CLIENT ID	CLIENT SECRET	OAUTH GRANT TYPE
○	Web App	✓	✓	Authorization Code
○	Public App	✓		Authorization Code
●	Server-to-Server	✓	✓	Client Credentials

4. Give the Data Extensions (Read, Write) permissions. Click Save.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy



Edit

CROSS CLOUD PLATFORM

Marketing Cloud Connect
□ Read
□ Write
□ Send

DATA

Data Extensions
☑ Read
☑ Write

File Locations
□ Read
□ Write

Tracking Events
□ Read
□ Write

DATA ENCRYPTION

Manage Keys
□ View
Status/Logs
□ Rotate
□ Revoke

DATA FACTORY UTILITY

Access
□ Configure

EVENT NOTIFICATIONS

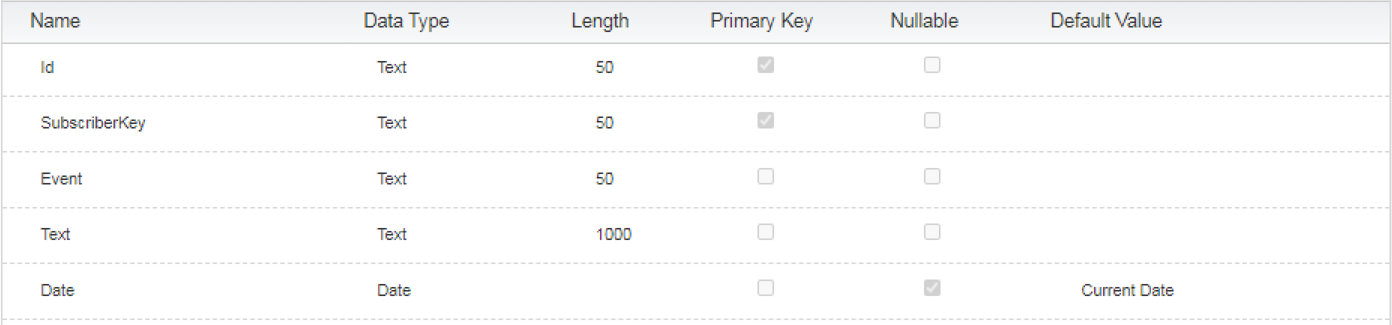
Callbacks
□ Create
□ Read
□ Update
□ Delete

Subscriptions
□ Create
□ Read
□ Update
□ Delete

5. Your installed package is ready but we are not done with it yet. Leave this browser tab open for now and navigate to Data Extension creation in Email Studio or Contact Builder. Create the new Data Extension with the following fields:

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

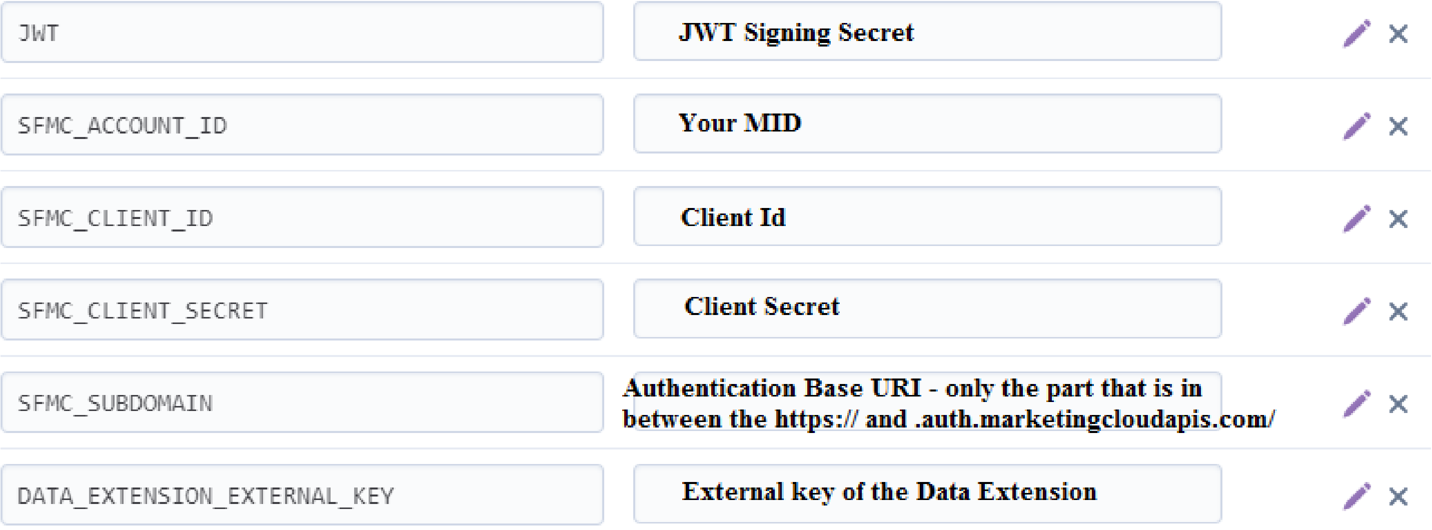
Ok Privacy policy



Name	Data Type	Length	Primary Key	Nullable	Default Value
Id	Text	50	☑	□	
SubscriberKey	Text	50	☑	□	
Event	Text	50	□	□	
Text	Text	1000	□	□	
Date	Date		□	☑	Current Date

Note that the Date field will be populated by Marketing Cloud Itself.

6. Now you have everything to complete your Config vars in Heroku. Go back to your app's Settings in Heroku and fill in the values:



JWT	JWT Signing Secret
SFMC_ACCOUNT_ID	Your MID
SFMC_CLIENT_ID	Client Id
SFMC_CLIENT_SECRET	Client Secret
SFMC_SUBDOMAIN	Authentication Base URI - only the part that is in between the https:// and .auth.marketingcloudapis.com/
DATA_EXTENSION_EXTERNAL_KEY	External key of the Data Extension

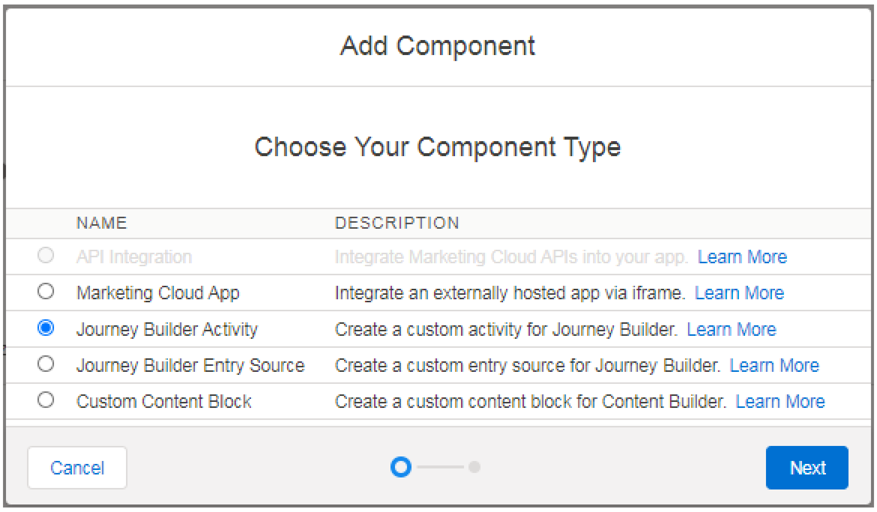
Note: Due to the security enhancements in one of the latest commits you would also need to add the 'STACK' variable and assign it a value of your Marketing Cloud Stack (e.g. s10, s50, etc.).

7. Let's make sure that everything works. To do this, click Open app. You should see the UI of your application.

8. All that's left is to complete the setup in Marketing Cloud and we'll be ready to test. Go back to your Installed Package window and click Add Component. Choose Journey Builder Activity and click Next.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

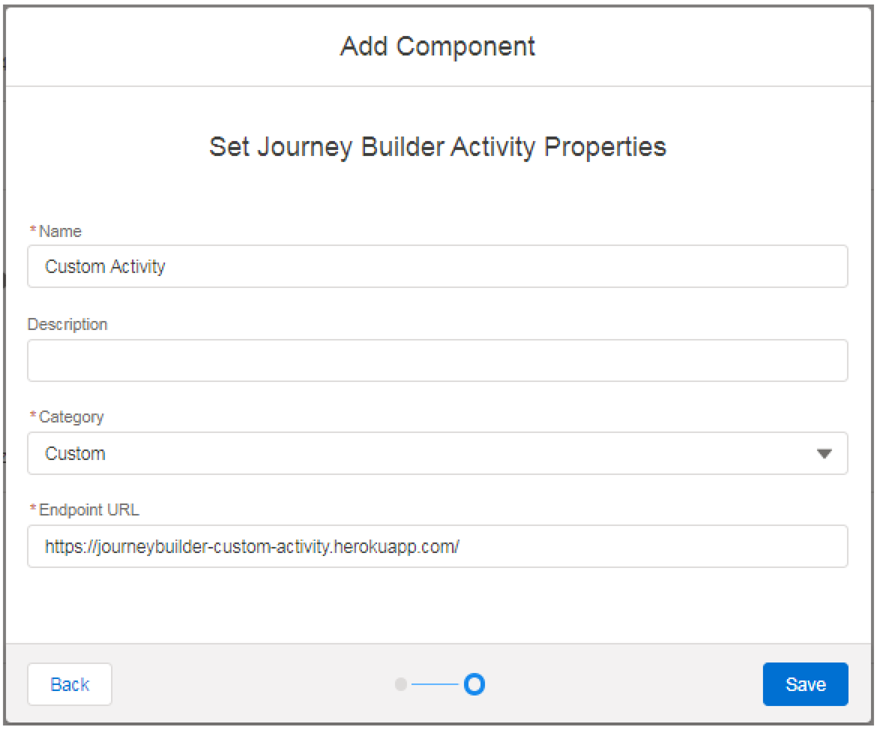


Add Component

Choose Your Component Type

	NAME	DESCRIPTION
○	API Integration	Integrate Marketing Cloud APIs into your app. Learn More
○	Marketing Cloud App	Integrate an externally hosted app via iframe. Learn More
●	Journey Builder Activity	Create a custom activity for Journey Builder. Learn More
○	Journey Builder Entry Source	Create a custom entry source for Journey Builder. Learn More
○	Custom Content Block	Create a custom content block for Content Builder. Learn More

9. Name your custom activity, choose the category where you want it to be displayed, paste your Application URL, and click Save



Add Component

Set Journey Builder Activity Properties

*Name
Custom Activity

Description

*Category
Custom

*Endpoint URL
https://journeybuilder-custom-activity.herokuapp.com/

Hooray! We are done with the configuration and our app is ready for testing. Let's jump into the Journey Builder and proceed with our custom activity configuration and testing.

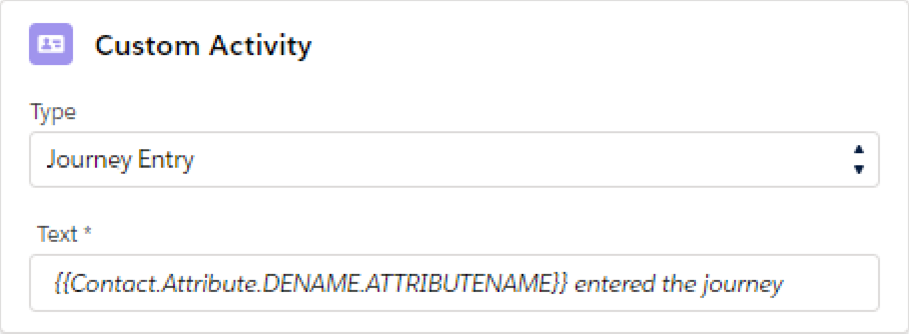
Journey Builder – Custom Activity Configuration and Testing

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

activity into the Journey canvas. You will find your custom activity in the category you selected when creating the component. You can always go back to the component settings and choose a different category.

Now open your custom activity and fill it in as shown in the picture:

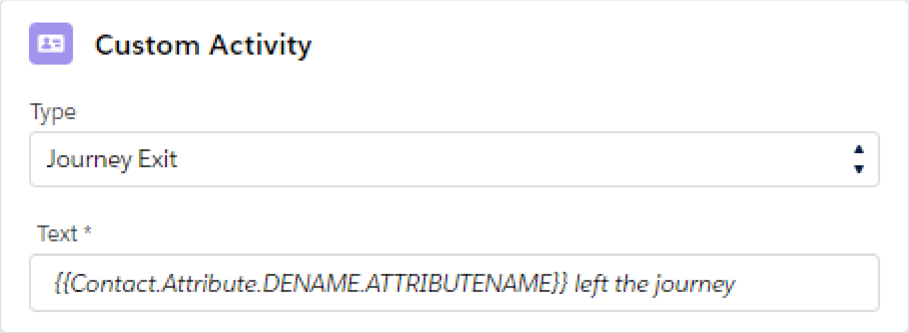


Custom Activity

Type
Journey Entry

Text *
{{Contact.Attribute.DENAME.ATTRIBUTENAME}} entered the journey

Add another custom activity and fill it like this:



Custom Activity

Type
Journey Exit

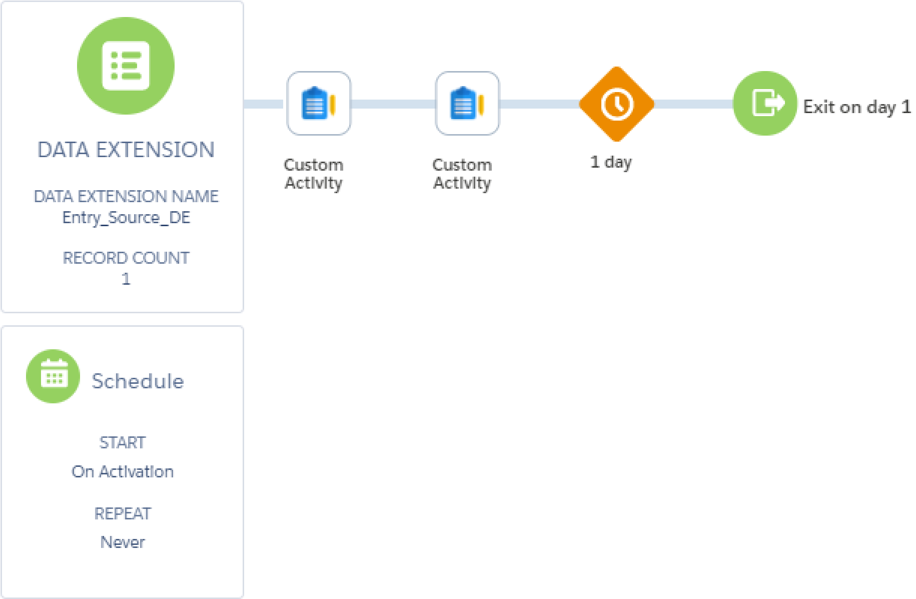
Text *
{{Contact.Attribute.DENAME.ATTRIBUTENAME}} left the journey

In both examples above, remember to replace DENAME with the name of your Data Extension, and ATTRIBUTENAME with the name of the attribute you want to use. If you are not familiar with this syntax, we recommend reading about Data Binding. Most importantly, make sure the Data Extension you choose is in the Attribute Group inside the Data Designer. You can also use the attributes of your Entry Source using the syntax: *{{Event.EventDefinitionKey.AttributeName}}*. Don't forget to replace the EventDefinitionKey and AttributeName with actual values.

So, after you're done, your Journey canvas should look similar to this one:

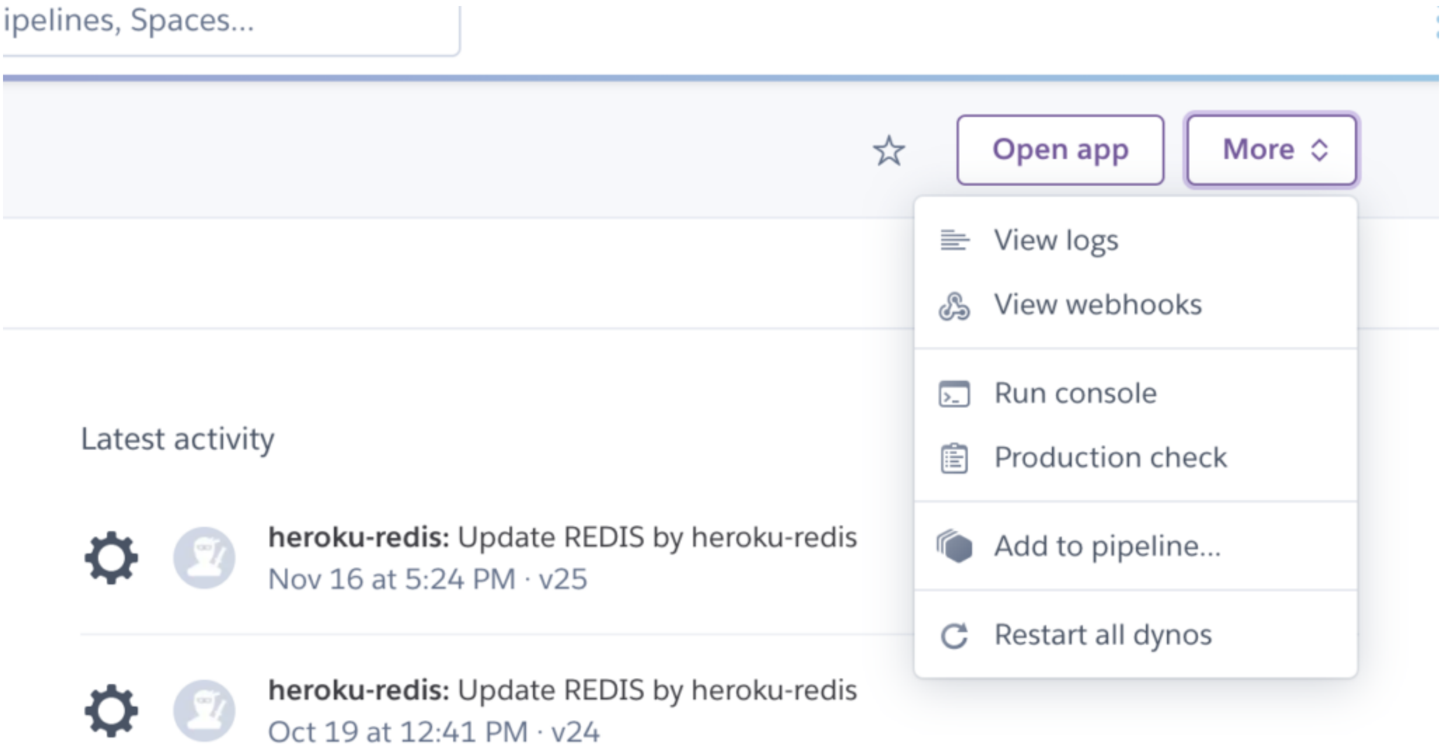
We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy



If something went wrong, we recommend looking at the logs on Heroku first. There are two options:

Using the Web Interface, open your application and click More -> View logs:



heroku-redis: Update REDIS by heroku-redis
Nov 16 at 5:24 PM · v25

heroku-redis: Update REDIS by heroku-redis
Oct 19 at 12:41 PM · v24

This method is not the most convenient because you will only see the last few lines from the logs. It's way cooler to use the Command Line Interface!

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

Then you should log in using the heroku login command

Next you can view the logs using the command:

```
Heroku logs --tail --app application_name
```

We also recommend installing one of the free addons for storing logs, this can be useful for debugging: https://elements.heroku.com/addons#logging

You can check the result of data recording in your Data Extension.

Conclusion

So, let’s sum up. In this article, we have gone through:

- How to develop custom Journey Builder Activities
- How to deploy and configure them on Heroku
- How to connect them to Marketing Cloud
- How to test them and check the logs

Stay tuned for more custom components in the following articles:

- Journey Builder Decision Split
- Journey Builder Entry Source
- Content Block
- Marketing Cloud Application

Spread the love

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

Orkhan Alakbarli

Salesforce Marketing Cloud Operations Director with solid experience in Marketing Automation and Custom Components Development.

← PREVIOUS ARTICLE

Multi-Factor Authentication

NEXT ARTICLE →

Identity Management with Interaction Studio (Evergage)

4 Comments

Diogo
29.09.2021 at 10:58

Hello,
First of all cheers for the article. Very helpful and well-structured. How could one do to throttle contacts entering a custom activity? I have a custom activity through which contacts enter and for each of them is performed a request to an external api, that has itself a limit of 1000 requests/s. Thereby, I would like to limit the number of contacts going through the custom activity to 1000 at once.

Many thanks,
Diogo Pereira.

REPLY

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

as JB is a 1-to-1 communication tool. However, on the backend of your custom activity you can temporarily store all incoming contacts into some database (e.g. MongoDB, PostgreSQL, etc.) and then process them in batches within regular intervals (e.g. 1000 contacts per minute). Then once processed, remove these contacts from the database to ensure they are not processed again during the next iteration. This should solve your issue as you will be doing fewer requests to the external API.
Hope that helps
Orkhan

REPLY

Clayton Haberman
17.03.2022 at 19:55

Hi Orkhan,

This is a great post. Absolutely everything runs fine except the data it not being entered into the DE. When running a tail I see this:

```
[LOGGER] 22-03-17 16:03:71 info : {
2022-03-17T16:40:29.714640+00:00 app[web.1]: inArguments: [
2022-03-17T16:40:29.714641+00:00 app[web.1]: {
2022-03-17T16:40:29.714641+00:00 app[web.1]: contactKey: ‘chaberman+proof@credible.com’,
2022-03-17T16:40:29.714642+00:00 app[web.1]: DropdownOptions: ‘journeyExit’,
2022-03-17T16:40:29.714642+00:00 app[web.1]: Text: ‘exited the Journey’
2022-03-17T16:40:29.714642+00:00 app[web.1]: }
2022-03-17T16:40:29.714643+00:00 app[web.1]: ],
2022-03-17T16:40:29.714643+00:00 app[web.1]: activityObjectID: ‘8fe82107-ef2a-4e22-bc6e-3a477c330dfc’,
2022-03-17T16:40:29.714643+00:00 app[web.1]: journeyId: ‘7b0ba3ee-591c-48c4-8c2a-fd712d92ff50’,
2022-03-17T16:40:29.714644+00:00 app[web.1]: activityId: ‘8fe82107-ef2a-4e22-bc6e-3a477c330dfc’,
2022-03-17T16:40:29.714644+00:00 app[web.1]: definitionInstanceId: ‘ee214c01-a00d-
```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

```
'chaberman+proof@credible.com',
2022-03-17T16:40:29.714645+00:00 app[web.1]: mode: 0
2022-03-17T16:40:29.714645+00:00 app[web.1]: }
2022-03-17T16:40:29.896449+00:00 app[web.1]: [LOGGER] 22-03-17 16:03:90 error :
error
```

The problem is that there's no description of what the error is so I don't know how to debug. I have checked and doubled checked my installation and setup against your article and can't find anything wrong. Do you have any thoughts?

REPLY

Orkhan Alakbarli

18.03.2022 at 11:47

Hi Clayton and thank you for your feedback!

It errors during the attempt to make an API call to SFMC. Have you tried doing the call from your computer? If not, I would suggest trying it (you can use Postman for that). Use the same credentials you set up in Heroku (SFMC_ACCOUNT_ID, SFMC_CLIENT_ID, SFMC_CLIENT_SECRET, SFMC_SUBDOMAIN, DATA_EXTENSION_EXTERNAL_KEY). If you'll get an error – double-check the followings:

1. Installed Package with Server-to-Server integration component exists and has Data Extensions Read/Write permissions.
2. Data extension with proper configuration exists. Watch out for Primary Keys, Data Types and other parameters.
3. Proper values used for Env Vars. Use the Client ID, Client Secret and Subdomain from the Installed Package with the Integration Component. Don't forget other variables such as SFMC_ACCOUNT_ID and DATA_EXTENSION_EXTERNAL_KEY.

To execute the above mentioned you can refer to steps 1-6 in **Marketing Cloud – Installed Package and DE Configuration** section.

REPLY

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy

Comment

Name*

Email*

Website

□ Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

Marketing automation tech, news & insights Vilva | Developed By Blossom Themes. Powered by WordPress.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok Privacy policy