Setting Up Jira Webhooks: 6 Easy Steps to Create, Manage and Automate Events

[Ofem Eteng](https://hevodata.com/learn/author/ofem-eteng/) on [Data Driven](https://hevodata.com/learn/category/data-integration/data-driven/), [Data Driven Strategies](https://hevodata.com/learn/category/data-integration/data-driven-strategies/), [Data Integration](https://hevodata.com/learn/category/data-integration/), [ETL Tutorials](https://hevodata.com/learn/category/etl-tutorials/), [Jira](https://hevodata.com/learn/category/jira/), [Project Management Tool](https://hevodata.com/learn/category/project-management-tool/), [Webhook](https://hevodata.com/learn/category/webhook/) • October 19th, 2021 • [**WRITE FOR HEVO**](https://community.hevodata.com/write-for-hevo)

<https://hevodata.com/learn/jira-webhooks/>

Keeping up with changes and updates can be a challenging and daunting task, hence, we tend to look for methods in which this could be done in an automated manner without having to query our entire data every time to see where an alteration occurs but get the new updates with ease through effective methods.

There are many reasons why we need to get updates on data, especially to make informed and specific decisions about clients, so certain criteria are set using events and if they are met, an action is taken to notify us about the change.

In this article, you will understand more about **Jira**, a software that is used for managing, tracking, and reporting by companies, and you will also learn how to set up **Jira Webhooks** to enable you to keep track of everything going on with your customer base.

**Table of Contents**

* [Introduction to Jira](https://hevodata.com/learn/jira-webhooks/#jira)
* [Introduction to Webhooks](https://hevodata.com/learn/jira-webhooks/#webhooks)
* [Steps to Set Up Jira Webhooks](https://hevodata.com/learn/jira-webhooks/#steps)
  + [Creating a Webhook in Jira Administration Console](https://hevodata.com/learn/jira-webhooks/#s1)
  + [Registering a Jira Webhooks using the REST API: Connect and Oauth 2.0 Apps](https://hevodata.com/learn/jira-webhooks/#s2)
  + [Managing Jira Webhooks](https://hevodata.com/learn/jira-webhooks/#s3)
  + [Automating Jira Webhooks](https://hevodata.com/learn/jira-webhooks/#s4)
  + [Executing Jira Webhooks](https://hevodata.com/learn/jira-webhooks/#s5)
  + [Adding Jira Webhooks as a Post Function to a Workflow](https://hevodata.com/learn/jira-webhooks/#s6)
* [Conclusion](https://hevodata.com/learn/jira-webhooks/#conclusion)

**Introduction to Jira**

Jira Webhooks - Jira Logo[Image Source](https://www.atlassian.com/software/jira)

**Jira** is a proprietary tool developed by Australian company **Atlassian** used for **Bug Tracking**, **Issue Tracking**, and agile **Project Management**. The word Jira is derived from a truncation of the Japanese word “Gojira” which means “Godzilla” as developers in Atlassian used to refer to Jira as Bugzilla because it was initially used for just bug tracking.

Jira is a suite of agile work management solutions that powers collaboration across teams to help you plan, assign, track, report, and manage work and brings teams together for everything from agile software development to customer support.

**Jira**has become popular over the years and it is used by over **180,000**customers in more than **190** countries, it has listed products such as Jira Software, Jira Service Management, Jira Work Management, and Jira Align.

To explore Jira in detail, visit the [Jira Homepage](https://www.atlassian.com/software/jira).

**Introduction to Webhooks**

[Image Source](https://www.testdevlab.com/blog/wp-content/uploads/2017/11/webhook-1680x423.png)

A **Webhook** can be defined as a method of altering and modifying the behavior of web pages or web applications by using custom callbacks. It can be explained basically as a user-defined HTTP callback that is triggered by specific events.

A **Webhook** can also be described as a type of API that is driven by events rather than requests, so instead of an application requesting another application to send a response, a Webhook service allows a program to send data to another as soon as a particular event takes place as designated by the user. They are also referred to as “**Reverse APIs**” in the sense that the application initiates communication by sending the data when an event triggers it rather than being the one receiving a request.

As web services have increased astronomically over the last few years, Webhooks have encouraged real-time notifications of data updates without the need to create or develop APIs to handle such scenarios.

In Jira, you can use **Jira Webhooks** to notify your web app or web application when certain events occur in Jira like alerting your remote application when an issue has been updated or when a sprint has begun and this will help your remote applications not to periodically poll Jira to ascertain if changes have occurred.

**Simplify Jira ETL with Hevo’s No-code Data Pipeline**

[Hevo Data](https://hevodata.com/?utm_source=Learn_jira-webhooks&utm_medium=l1_home&utm_campaign=blog_click), a No-code Data Pipeline, helps load data from any data source such as Databases, SaaS applications, Cloud Storage, SDK,s, and Streaming Services and simplifies the ETL process. It supports [100+ Data Sources](https://hevodata.com/integrations/?utm_source=Learn_jira-webhooks&utm_medium=l2_integrations&utm_campaign=blog_click) including **40+ Free Sources**such as **Jira**. It is a 3-step process by just selecting the data source, providing valid credentials, and choosing the destination. Hevo loads the data onto the desired Data Warehouse/destination and enriches the data and transforms it into an analysis-ready form without having to write a single line of code.

Its completely automated pipeline offers data to be delivered in real-time without any loss from source to destination. Its fault-tolerant and scalable architecture ensure that the data is handled in a secure, consistent manner with zero data loss and supports different forms of data. The solutions provided are consistent and work with different BI tools as well.

[**GET STARTED WITH HEVO FOR FREE**](https://hevodata.com/signup/?utm_source=Learn_jira-webhooks&utm_medium=mango_l1_signup&utm_campaign=blog_click)

**Check out why Hevo is the Best:**

* **Secure**: Hevo has a fault-tolerant architecture that ensures that the data is handled securely and consistently with zero data loss.
* **Schema Management**: Hevo takes away the tedious task of schema management & automatically detects the schema of incoming data and maps it to the destination schema.
* **Minimal Learning**: Hevo, with its simple and interactive UI, is extremely simple for new customers to work on and perform operations.
* **Hevo Is Built To Scale**: As the number of sources and the volume of your data grows, Hevo scales horizontally, handling millions of records per minute with very little latency.
* **Incremental Data Load**: Hevo allows the transfer of data that has been modified in real-time. This ensures efficient utilization of bandwidth on both ends.
* **Live Support**: The Hevo team is available round the clock to extend exceptional support to its customers through chat, email, and support calls.
* **Live Monitoring**: Hevo allows you to monitor the data flow and check where your data is at a particular point in time.

Simplify your Data Analysis with Hevo today!

[**SIGN UP HERE FOR A 14-DAY FREE TRIAL!**](https://hevodata.com/signup/?utm_source=Learn_jira-webhooks&utm_medium=mango_l2_signup&utm_campaign=blog_click)

**Steps to Set Up Jira Webhooks**

Typical **Jira Webhooks** are made up of the following information which is supplied when creating a Webhook.

* **Name:**This is the name of the Webhook created in Jira.
* **URL:**This is the URL where the callback will be sent when created.
* **Scope:**This determines the scope of the Webhook, either all issues or a limited set of issues specified by JQL like**project = TEST and fixVersion = future.**
* **Events:**This is the event that will determine the posting of the URL when triggered and it can be either all events or a specific set of events.

**Methods to Register/Create a Webhook**

**Jira Webhooks**can be registered/created in Jira by using either of the following methods:

* Creating a Webhook in Jira Administration Console.
* Declaring a Webhook in the description of your Atlassian Connect app.
* Registering a Webhook using the REST API: Connect and Oauth 2.0 apps
* Registering a Webhook using the REST API: Other integrations.

In this write-up, you will be introduced to Creating a Webhook in Jira Administration and Registering a Webhook using the REST API: Connect and Oauth 2.0 apps. So, follow the steps below to set up Jira Webhooks.

* [Creating a Webhook in Jira Administration Console](https://hevodata.com/learn/jira-webhooks/#s1)
* [Registering a Jira Webhooks using the REST API: Connect and Oauth 2.0 Apps](https://hevodata.com/learn/jira-webhooks/#s2)
* [Managing Jira Webhooks](https://hevodata.com/learn/jira-webhooks/#s3)
* [Automating Jira Webhooks](https://hevodata.com/learn/jira-webhooks/#s4)
* [Executing Jira Webhooks](https://hevodata.com/learn/jira-webhooks/#s5)
* [Adding Jira Webhooks as a Post Function to a Workflow](https://hevodata.com/learn/jira-webhooks/#s6)

**1) Creating a Webhook in Jira Administration Console**

To register a Webhook using this method, carry out the following steps:

* Go to the Jira administration console, and navigate to **System,**then **Webhooks**found in the **Advanced** section.
* Click on **Create a webhook**
* In the form that pops up, enter the details for your new Webhook.
* Finally, click on **Create**to register the Webhook**.**

[Image Source](https://developer.atlassian.com/server/jira/platform/images/jira-admin-webhooks.png)

**2) Registering a Jira Webhooks using the REST API: Connect and Oauth 2.0 Apps**

To register a Webhook via REST, execute POST in JSON format to the following URL **<JIRA\_URL>/rest/webhooks/1.0/webhook**

{

"name": "my first webhook via rest",

"url": "http://www.example.com/webhooks",

"events": [

"jira:issue\_created",

"jira:issue\_updated"

],

"filters": {

"issue-related-events-section": "Project = JRA AND resolution = Fixed"

},

"excludeBody" : false

}

The response will return the Webhook in JSON with additional information which will include the user that created the Webhook, the created timestamp, among others.

**3) Managing Jira Webhooks**

To be able to carry out changes to your Webhooks in Jira, you will need to manage them by editing, deleting, and disabling the Webhooks. Two methods to manage Webhooks, using the administration console and REST API are described below.

To manage Webhooks in Jira using the administration console, do the following:

* Log in to Jira with your **Administered Jira** global Permission.
* Choose Settings, navigate to **System.**
* Under **Advanced,**select **Webhooks.**
* Click the **Summary**of the Webhook in the left Webhooks column to display the details of the Webhook to enable you to edit, delete, and disable a Webhook from the details panel. Deleting a Webhook removes it permanently while disabling a Webhook prevents it from firing.

To manage Webhooks in Jira using the Jira REST API to Unregister (Delete) a Webhook, execute DELETE to the following URL:

<JIRA\_URL>/rest/webhooks/1.0/webhook/{id of the webhook}

curl --user username:password -X DELETE -H "Content-Type: application/json" <JIRA\_URL>/rest/webhooks/1.0/webhook/70

The example above will delete a Webhook of ID 70.

To **edit**Webhooks in Jira using the Jira REST API, execute the**PUT**method to the following URL:

<JIRA\_URL>/rest/webhooks/1.0/webhook/{id of the webhook}

{

"name": "my first webhook via rest",

"url": "http://www.example.com/webhooks",

"events": [

"jira:issue\_created",

"jira:issue\_updated"

],

"filters": {

"issue-related-events-section": "Project = JRA AND resolution = Fixed"

},

"excludeBody" : false

}

To **query** Webhooks in Jira using the Jira REST API, execute **GET** to the following URL:

<JIRA\_URL>/rest/webhooks/1.0/webhook

curl --user username:password -X GET -H "Content-Type: application/json" <JIRA\_URL>/rest/webhooks/1.0/webhook

To**get**a specific Webhook by its ID, execute **GET** to the following URL:

<JIRA\_URL>/rest/webhooks/1.0/webhook/<webhook ID>

curl --user username:password -X GET -H "Content-Type: application/json" <JIRA\_URL>/rest/webhooks/1.0/webhook/72

The example above will get a Webhook with an ID of 72.

**4) Automating Jira Webhooks**

Applications can react to conditions you set by triggering an action when an event occurs or a condition set is reached. When you configure a rule, you can specify **WHEN**s and **IF**s, then use the Webhook to define your action. To configure an automation rule to fire a Webhook to notify your application, do the following:

* Navigate to **Project settings,**
* Go to **Automation**then create a **Custom rule**or edit an existing rule.
* Configure the **WHEN**and **IF**settings as desired, then add a **THEN**action and choose **Webhook**.

[Image Source](https://developer.atlassian.com/cloud/jira/service-desk/images/jsd-then-webhook.png)

* Configure the Webhook setting to indicate the URL, include payload, name your rule, and then save it.

**5) Executing Jira Webhooks**

When a Webhook is triggered, it sends a request with a JSON callback and that usually contains the **Webhook Event ID, timestamp,**and information about the entity associated with the event. An example of a callback for an issue-related event is shown below:

{

"timestamp"

"event"

"user": {

--> See User shape in table below

},

"issue": {

--> See Issue shape in table below

},

"changelog" : {

--> See Changelog shape in table below

},

"comment" : {

--> See Comment shape in table below

}

}

A Webhook execution can be done in either of the following:

* **Request Method**: When a webhook **THEN**is triggered, Jira will fire an **HTTP POST** request to the URL which you configured.
* **Success Criteria and Timeout:**A Webhook **HTTP POST** request can be considered a success when the server returns a response with a status code ranging from 200 to 300.
* **Asynchronous Webhook Execution:**The Webhook **THEN** action is executed asynchronously. When many executions are triggered in a short period, they are queued and executed one at a time without obstructing other automated rules you set in Jira.

**6) Adding Jira Webhooks as a Post Function to a Workflow**

Webhooks can be attached as a workflow post function to make it easy to trigger a Webhook when an issue makes a workflow transition. To add a Webhook as a post function to a workflow, do the following:

* Go to workflow configuration and select the desired transition in the workflow designer.
* Click **Post functions,**then select **Add post function.**
* Select **Trigger a Webhook** from the list of post functions and click **Add**.
* Select the desired Webhook from the list of webhooks and click **Add**to add the Webhook as a post function.

**Conclusion**

This article has explained the workings of **Jira Webhooks** and described how to create, manage, and automate Webhooks amongst other operations on the **Jira software**. Having gone through the article, you should now understand how Webhooks work and how you can leverage them to track events on the Jira platform.

To further increase your productivity and business know-how, combining **Jira with Hevo Data** can be very helpful as **Hevo Data** has baked in **Jira** support alongside access to other providers in its data warehousing platform.

[Hevo Data](https://hevodata.com/?utm_source=Learn_jira-webhooks&utm_medium=l3_home&utm_campaign=blog_click) is a **No-Code Data Pipeline** that offers a faster way to move data from **100+ Data Sources** including **40+ Free Sources** such as **Jira**, into your Data Warehouse to be visualized in a BI tool. Hevo is fully automated and hence does not require you to code.