REST API Tutorial

REST API Tutorial

Jitterbit Harmony can be used to consume any RESTful web service, also known as a REST API. As REST APIs are HTTP-based, you connect to them using an HTTP Source or HTTP Target in Harmony Design Studio. REST APIs are not to be confused with web services that use a SOAP API, for which a Harmony web service method is used instead.

This tutorial walks through a complete example of how to consume REST APIs in Jitterbit Harmony using Atlassian Jira as an endpoint for demonstration purposes. The same process can be applied to any endpoint that uses a REST API. Some other examples of REST APIs include Magento, ServiceNow, Shopify, DocuSign, SharePoint, Epicor, Zendesk, Zoho, and Sugar CRM, to name a few.

The intended audience for this tutorial is someone who is unfamiliar with REST APIs and the testing of them, and has a basic knowledge of Jitterbit Harmony. If instead you have more experience with one or all of these topics, we recommend skimming through the outline of steps below for any points of interest. The last topic, Using a REST API in Operations, may be of most interest to those experienced with REST APIs but less experienced with Jitterbit Harmony.

Tutorial Outline

These topics are covered by this tutorial, presented in the order in which they should be completed:

1. **Researching a REST API**
   There are a few specific items you should look for in an API's documentation that you will need later to validate the API and provide in the configuration in Design Studio. These include setting up the type of authentication you want Jitterbit Harmony to use to authenticate with the API, and obtaining request URLs, request headers, request structures, and expected response structures.
2. **Validating a REST API**
   Before connecting to a REST API with Jitterbit Harmony, testing and validation using an independent tool is strongly recommended. This page walks through testing authentication, and validating and saving structures for each request and response.
3. **Connecting to a REST API**
   In Design Studio, an HTTP source or target must be configured for the appropriate HTTP method of your request (GET, PUT, POST, DELETE, or custom method) so you can use it an operation. While this page focuses on common configuration options, the pages HTTP Source and HTTP Target provide more detailed information about all options that are available to configure.
4. **Using a REST API in Operations**
   Although each REST API adheres to the same architectural constraints, they are not all designed the same way for each HTTP method. As each specific request and response depends on the specific API, we present four design patterns for designing operations:
   - **A Response Structure Only**
     This pattern applies to methods where you need to provide only a response structure and no request structure. This is usually the case with GET methods, since you are typically only requesting data to be sent back from the API, rather than providing data to the API. You are still sending a request to the API, but your request isn't in the form of structured data. The request is made simply using the request URL and any request headers or other configurable options set during HTTP source configuration.
   - **Both Request and Response Structures**
     This pattern applies to methods where you are providing both request and response structures. This is often the case with POST methods, where you are requesting data to be created and then receive back information about what was created. The response back from the API can be used in any way, but it is often the case to use the new object IDs in a later operation.
   - **A Request Structure Only**
     This pattern applies to methods where you provide a request structure, but there is no structured data returned by the REST API. For APIs that have a request structure only, this doesn't mean the API doesn't respond; it means that the API response may be as simple as a status code.
   - **Neither a Request Nor a Response**
     This pattern applies to methods that do not accept any structured input data nor return any structured output data. In such cases, the request is usually specified entirely with the URL and request headers; the response is normally a status code.

On This Page

Related Articles

- Connecting to a REST API
- HTTP Source
- HTTP Target
- Researching a REST API
- Using a REST API in Operations
- Validating a REST API

Related Topics

- Design Studio
- HTTP
- Operations
- Sources and Targets
- Web Services

Additional Resources

- **API Testing Tools:** Postman or SoapUI
- **Free APIs for Testing:** https://github.com/toddmotto/public-apis
- **Other APIs by Category:** https://www.getpostman.com/api-network/