APIFORTRESS | MuleSoft

Anypoint Platform Integration

More Information on the API Fortress Connector

Q3-2016

Introduction

API Fortress is a comprehensive API (web services) testing platform that can function both on the cloud and on-premises. Its core functionality is to test JSON / XML / plain text APIs against functional and performance expectations. Using an intuitive visual composer a user with average technical training can write intelligent API tests to confirm datatypes, accuracy, and structure. All testing events and data are stored and made available for further analysis.

THE ENGINE CAN WORK IN 2 WAYS:

- **Active:** API Fortress downloads the specified resource and tests it.
- **Passive:** API Fortress receives a payload from an external agent and evaluates it. In this document we are mainly referring to this type.

The Mulesoft Anypoint Platform is a World leading Enterprise Service Bus that connects heterogeneous services in unified flows, allowing a user with average technical training to define flows connecting those services together. Many of these services communicate using web services (often referred to as APIs) while others use specific protocols.

Benefits

API Fortress and the **Mule Anypoint Platform** are a perfect match. Mule users gain the ability to verify whether all their services and flows are working exactly as expected. Mule deployments are often a central node in an Enterprise architecture. This privileged position contains many of the company's core services and, due to that, represents the best fit for live testing purposes.

THIS FACT LEADS TO TWO KEY BENEFITS:

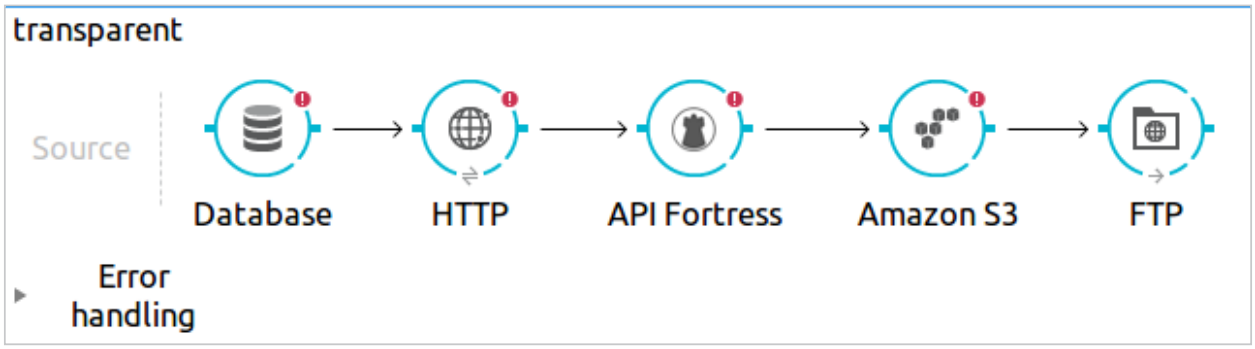
- The ability to test the behavior of every API involved as they work with realistic or real scenarios.
- The ability to test how flows are performing. The API Fortress connector is able to use structured data, such as maps and POJOs, as payloads.

For the **API Fortress** customers, this simplifies the method to forward traffic to the platform for live testing. Being able to do it with a module in one of the most prolific service buses in the World is a big strategic and technological advantage.

Modes

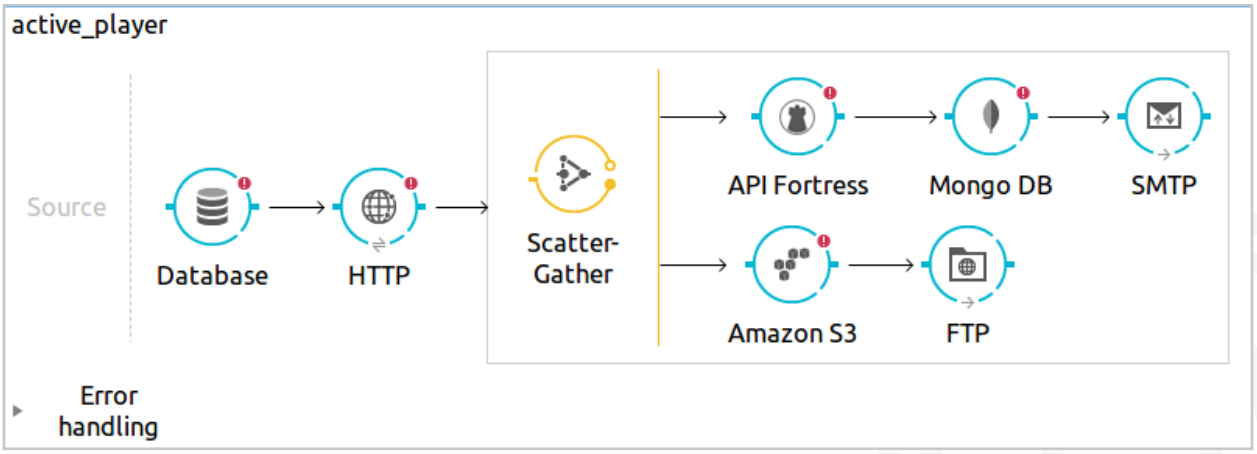
Given the benefits described in the previous chapter, the connector can behave in two different ways:

» **As a silent tester:** The connector forwards the data to an API Fortress instance (cloud or on prem) and lets the flow go its route undisturbed. The tests results will be available in the API Fortress dashboard.



▲ API Fortress is not interfering with the flow and is silently testing the payload.

» **As an active player in the flow:** The connector waits for the testing results, so that the flow can be configured to take action with the results of the test. Examples include (but are not limited to) storing the results in a company owned database, triggering alerts, calling other APIs, etc...

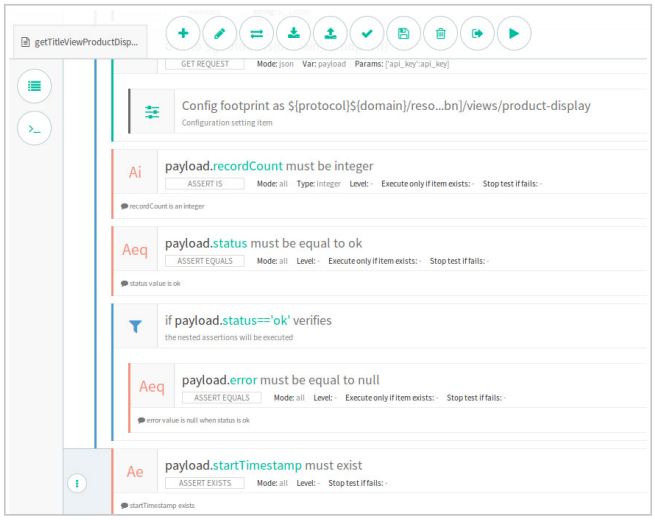


▲ In the top branch, the API Fortress connector waits for the testing results which are then stored in MongoDB, and then eventually sends an email.

Configuration

REQUIREMENTS:

- The user has at least a Trial or Premium account.
- The user has built at least one test in API Fortress that is going to be used.



◂ CONFIGURATION

The composition of a test is done using our one-of-a-kind visual composer.

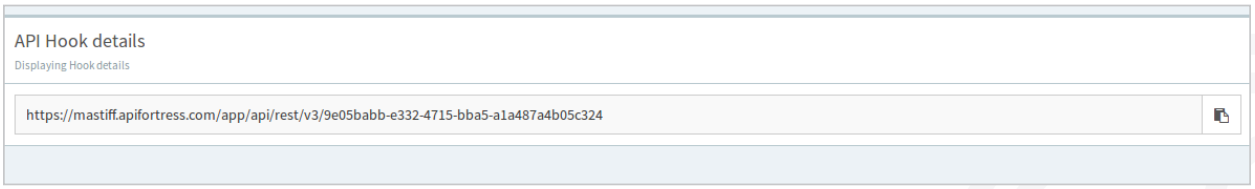
The user is then required to retrieve two pieces of information:

▾ THE TEST ID:

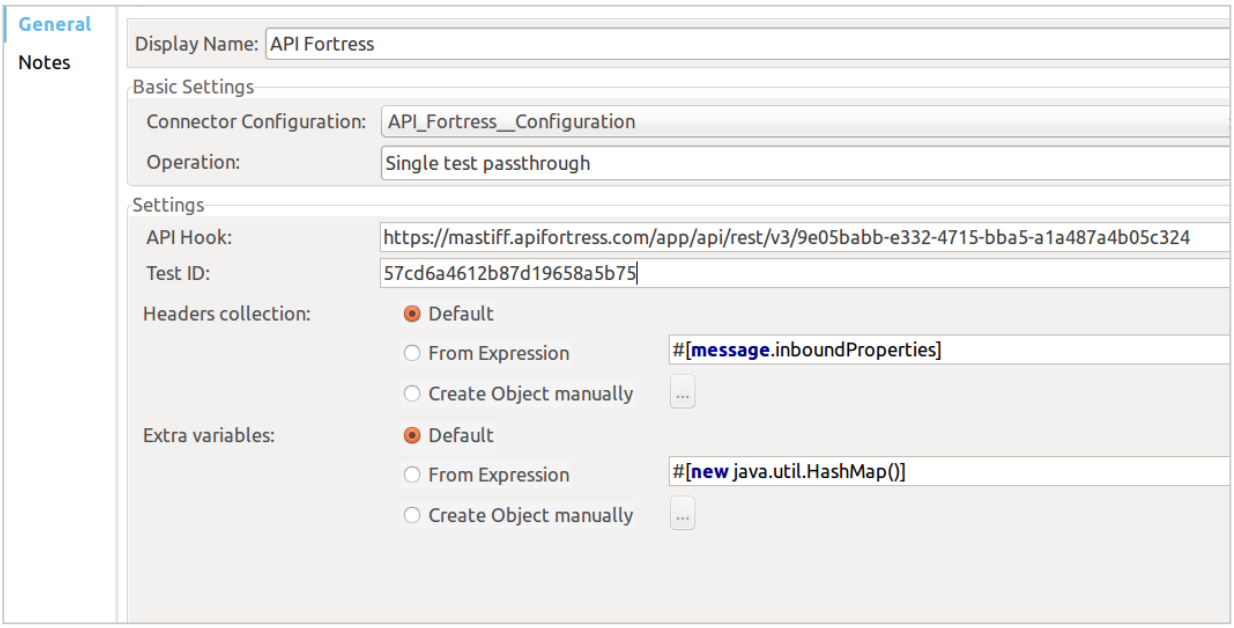
Project ID	862
Test ID	57a9a08abbb0fb4afdf3faf0

AND THE API HOOK URL:

The API Hook URL is a unique URL with randomly generated fragments that represents a project (just like Google Drive shareable URLs). When not dealing with sensitive data, most of its functionalities can be used without an authentication procedure. API Hooks can be revoked anytime.



In the Anypoint Studio the user now can deploy the API Fortress connector.



In this example, we're configuring the connector to work in **passthrough** mode, so that the payload remains untouched. As a tester might be interested in the response envelope, the user is given the capability to define where the **headers** are stored and add **extra parameters** to the scope. Examples include which server the flow is running on, the geographical location, if it's a staging or a production server, etc...

Operations

THE AVAILABLE OPERATIONS ARE:

- **Single Test Synchronous:** the connector runs a specific test and returns the result of the evaluation.
- **Single Test Passthrough:** the connector runs a specific test and leaves the payload untouched.
- **Automatch Synchronous:** lets API Fortress decide which tests are suitable and returns the result of the evaluation.
- **Automatch Passthrough:** lets API Fortress decide which tests are suitable and leaves the payload untouched.

For more information on how Automatch works, please refer to: http://apifortress.com/doc/automatch/

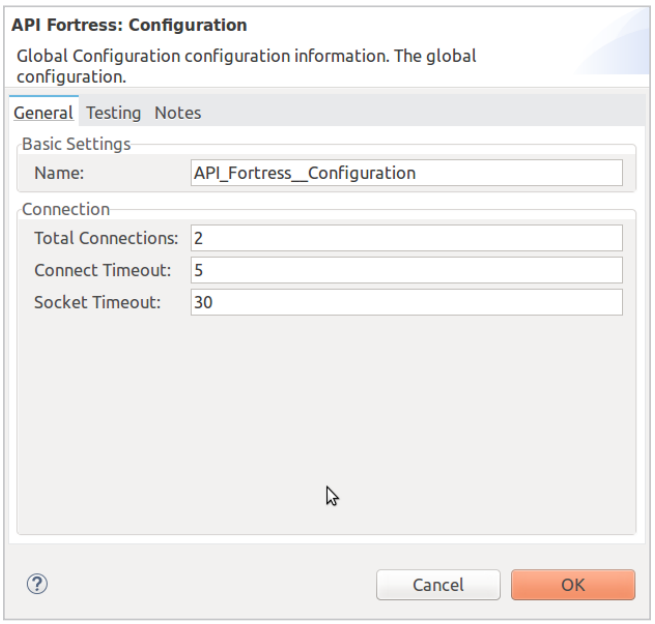
Global

The global configuration items are fairly simple.

The basic connectivity settings. These values should be modified to tune the impact of the API Fortress plugin on the overall instance I/O rates.

The defaults are fine in most occasions, but for heavy testing raising the total connections would be beneficial.

With poor network quality, increasing the timeouts might be required.

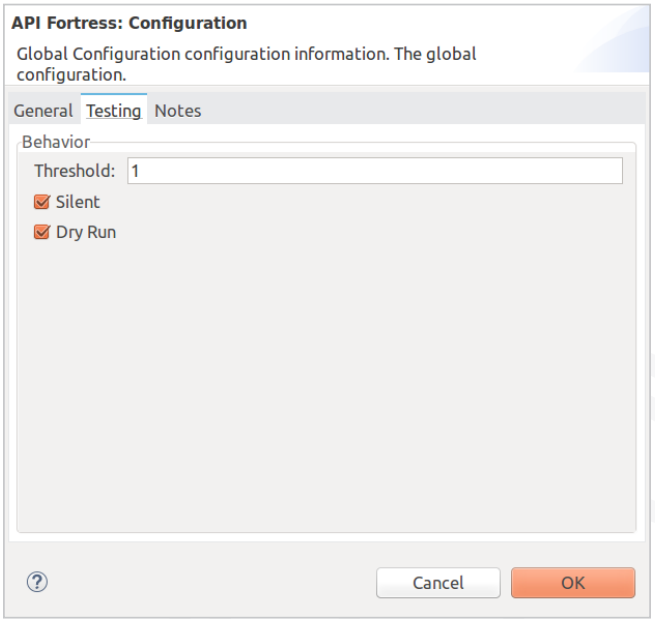


THESE ARE SOME API FORTRESS SPECIFIC SETTINGS.

Threshold: All requests are counted, and only multiples of this number are evaluated. This allows less “test banging” and more sparse results. Using 1 forces all requests to be checked.

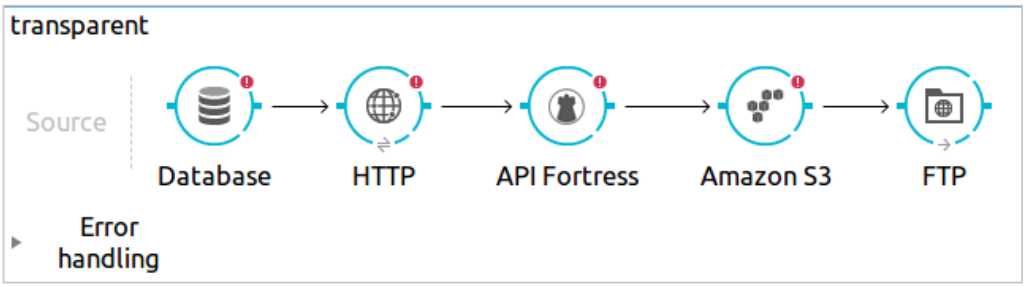
Silent: If checked no API Fortress alerts will be triggered in case of a failure.

Dry run: If checked the API Fortress Cloud will not store the results of the test (to be used in conjunction with synchronous operations).

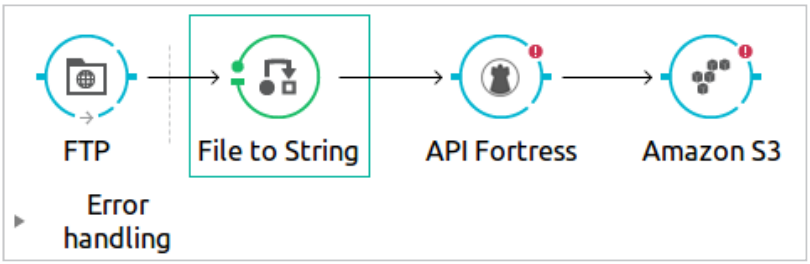


Use Cases

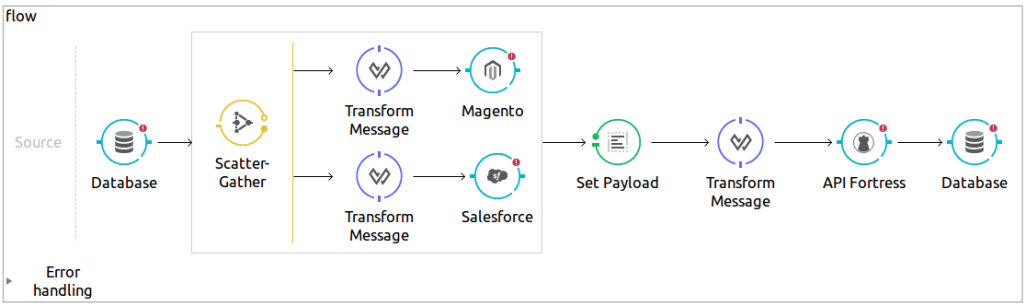
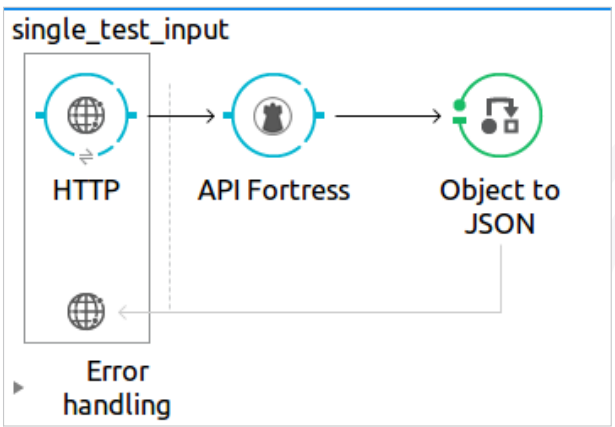
API Fortress tests the HTTP response but acts transparently.



API Fortress tests a plain text file.



API Fortress tests the final result of a flow.



API Fortress tests the final result of a flow.