Highly Confidential and Privileged

Invention Disclosure IDF-12971

Submitted by Aditi Aggarwal on Sep 28, 2023 04:35 PM

Title: Generic API connector for Spring boot applications

Inventors:

Other Inventors:

Design Submission: □

From Brainstorm: □

Brief Description:

As part of this invention, we have developed an API connector which can be integrate
service to make API requests in both Rest and GraphQL. This connector provides eas
integration to make these service requests. This is a low code solution to make API re
service without writing or managing several clients.

=> features provided
1) Client can make both Rest and Graphql requests in their application without definin
2) Clients can override connection pool and http settings if needed
3) Supports configurable retries and provides option to write custom retry handlers
4) utilizes the latest spring 5 web client which can be overridden and customised by tl
(Details and implementation of these features is mentioned in detail in the github read

Details: https://github.intuit.com/appintgwkflw-wkflautomate/api-connector

Tech Details: Parts of the Invention:

=> How does a client onboard?
1) The client needs to import generic connector as a dependency in their project
2) They need to define the connector config in the provided format in their application
This includes the url, http method type, request type, body parameters etc.
3) They need to extend the GenericConnector class where they will supply the conne
app
4) Now wherever they need to make the API request, they simply need to call the exe
genericConnector
5) All the examples and sample application is provided here in the readMe:
https://github.intuit.com/appintgwkflw-wkflautomate/api-connector

=>How does the connector work?

1) The client provides the connector class with unstructured input parameters for the r
execute method.

2) The library parses the connector config and input parameters and validate them.

3) Then the library extractes and resolves the body parameters which is essential to c
graphql requests:

a) Developer can provide the body parameters in any format (be it primitive or comple
map)

b) Inside the parameter extractor, we resolve the parameters in a recursive fashion ar
values based on the provided config.

c) After all the variables are resolved, a flatmap structure is created which is further us

4) After resolving body parameters, request headers are generated as per the config |
whether the connector config supplied is of type Rest or Graphql.

5) Post this, library constructs the request object as per the config provided.

6) Now url is parsed based on the env, request parameters and headers set so we ex
latest spring5 web client which can also make async requests if needed. This spring v
customised for our user based on the config provided by them else we initiate a defau

7) Final response is returned back to the user.

8) Retry Config is also taken as part of the config.

Tech Details: Closest Existing Systems:

IBM Connector:
https://www.ibm.com/docs/da/cabi/1.1.2?topic=connectors-configuring-rest-api-conne

Fortinet Connector:
https://docs.fortinet.com/document/fortimanager/7.4.0/administration-guide/321207/cr

Tech Details: Old vs New Parts:

We have seen the connector pattern in several products which are mentioned above.
contributions made in this invention:

-> This is first of its kind connector which takes care of API calls made in both Rest ar
code and maintaining clients

-> It is a generic connector, which means it can be used by anyone and everyone. Ge
available are written in a closed fashion which are only available on certain internal AI

-> This connector is completely extensible and customisable. All the features and con
can be easily overridden and overriden as per the usecase requirement

-> This invention also incorporates spring5 web client to execute requests. Spring 5 w
has less community support. With this connector, clients can leverage the benefits of
having to go into the nitty gritties of it

Distinction from IBM Connector: IBM connector is only available for IBM Cloud Busine
not available for other custom APIs, Graphql requests. They have also not mentioned
available and as per the understanding from docs, they are using traditional restTemp
request. There are no options for customisation and extensibility as well.

Distinction from Fortinet Connector: Fortinet connector can only be used to fortinet AF connect to any generic applications, or custom APIs. There is no support for graphql i customisations are available. Retry configuration is not expected as part of input. The exposed for this system but the use of traditional clients is expected

Intuit Product/Project Relation:	
Nature of Public Disclosure:	
Comments:	
Attachment URL 1:	https://github.intuit.com/appintgwkflw-wkflauto...
Attachment URL 2:	https://github.intuit.com/appintgwkflw-wkflauto...
Attachment URL 3:	

Inventors

First Name	Last Name	Email
Rohit	Kumar	rohit_kumar@intuit.com
Nishant	Sehgal	nishant_sehgal@intuit.com
Purushottam	Sinha	purushottam_sinha@intuit.com
Aditi	Aggarwal	aditi_aggarwal@intuit.com
Raghav	Agarwal	raghav_agarwal@intuit.com

Attachments

IDF-12971-Sep.08.2023 06:04 AM.pdf