GraphQL Integration with Mulesoft



IT for people. People for IT.

What is GraphQL?

GraphQL for APIs (Application Programming Interfaces) and runtime is an open-source query language allowing clients to choose the fields they are interested in, thus cut down on the amount of data they receive when doing a query, only concentrating on what they need. GraphQL queries always return predictable results. Using GraphQL may also reduce the number of API's you need to create to meet your client's needs.

Why GraphQL?

- Clients can handle logic as per their requirement. By using the GraphQL query they will get an expected response.
- By using GraphQL, Apps will be fast and stable because they control the data they get, not the server.
- Forget about versioning APIs. Evolve your API without versions meaning that adding more fields to an existing endpoint will not break your API.
- The frontend and backend teams can work independently. The frontend team can easily generate a schema from the backend without knowing the code. The schema generated can directly be used to create queries.

User Case 1:

Suppose you have a REST API that returns 8 fields in response, and you have a mobile application and a web application. As per the requirement, the web application expects only 6 fields, whereas the mobile application expects only 4 fields in response. Here, according to the REST API architecture we need to create two endpoints, but with GraphQL we can achieve the above scenario with a single endpoint without any code change. Below is the illustration of User Case 1.



User Case 2:

Suppose you must call multiple REST APIs and each API will respond with a different JSON (JavaScript Object Notation) payload. GraphQL will route multiple requests from REST API and aggregate all the responses and send back the required response to the client. Below is the illustration of User Case 2.



The above user cases can be implemented using two solutions: Anypoint Datagraph framework or GrahpQL connector. We will explore each of them below.

Anypoint Platform Datagraph tool:

Anypoint Datagraph enables you to connect with these graphs into one unified schema that runs as a single SaaS (Software as a Service) GraphQL endpoint. It contains all the links and fields defined in your APIs. As a result, you can query across the underlying APIs without needing to understand all the relationships or specific capabilities that exist within them.

This article assumes that you are familiar with Anypoint Platform and Anypoint Studio.

Firstly, you must create an API specification in the design centre and publish RAML into exchange.

→ Step 1:

Go to the Anypoint Datagraph tool and select Add API.

🚍 🏫 DataGraph		唐 Billennium 📍 AP
Unified Data Graph Sandbox		
	Start building your unified schema!	
	Before we can create your unified schema, wrill first need to get an API added + Add API	
DataGraph Quick Start Guide d	DataGraph Tutorial Paplare aur DataGraph hatorial for a detailed step-by-step walkthrough.	DataGraph Fundamental Concepts Detailed information covering all aspects of Anypoint DataGraph

→ Step 2:

Search your application in the search bar and after 'select application', click on Next: Configure URL.

🖶 🕎 DataGraph		点 Billennium	?	AP
Sandbox / Add API	Select API — ③ Configure UR: — ③ Provide authentication — ④ Preview Schema — ⑤ Edit Schema			
Q customer			3	×
Customer-bank-exp ☐ Published to Exchange: Jun, 30 2021		्र ⁹ View in E	xchange	
	Cancel	Next: Conf	igure U	a.

→ Step 3:

Confirm the API and Asset version and click on Confirm selection.

😑 🜸 DataGraph			,ត Billennium	?	AP
Sandbox / Add API	(1) Select API — (2) Configure URL — (3) Provide authentication — 1	4) Preview Schema — (5) Edit Schema			
customer-bank-exp					^
To configure a URL first select and confirm the API version	in you want to add				
API version ①	Asset version ®	About this API			
Confirm selection		ওঁ View API specification ঔ View API documentation			
		Cancel Back			ion

→ Step 4:

After confirming the selection, provide the base URL of your application and click on Next: Provide Authentication. (In my case I am using a mock URL from the Design Centre)

						리 Billennium	?
dbox / Add API) Select API — 🧿 Configure U	RL — ③ Provide authentication — ④	Preview Schema	— (5) Edit Schema		
API version @ V1 laser: Confirm selection @ Everything	Assec	version @ 0 Tarest	v	l Abou 2 View 2 View	nt this API API_specification API_documentation		
Add/Edit API URL Mis will be used to make requests to the AP Get an existing URL from Anypo This is for when the API is baing manage	nt Platform dim API manager er if Sechange.	Add a new URL Use this option if you want to	o manually add a URL for this APL				
and a state we dear manually about it							
API instance in Anypoint Platform v1:17018096	v		_				
API Instance in Anypoint Pietform v1:17018006 URL for the selected API instance above https://anypoint.mulesoft.com/mo	V :king/api/v1/Links/d748319e-	-ea76-47de-9cde-97a1df95319b/	Edit Life.				

→ Step 5:

If you have any authentication enabled for your API, please select the authentication method accordingly and click on Next: Preview Schema (In my case I am using No Auth).

Too Add API	— 🜔 Provide authentication — 🕢 Preview Schema — 🌀 Edit Schema	si dinerimum ?
stomer-bank-exp		
Provide authentication let: the authentication policy and provide credentials associated with this APP's GET endpoints		
select authentication policy Seponded authentication policy types are limited to those isled below. <u>Learn more</u> No Auth	About this API Image: API specification Image: View API specification Image: View API specification	

→ Step 6:

In the preview schema section you can see all the listed methods in your RAML and data types. Click on Next: Edit Schema.

🚍 🧇 DataGraph		🖧 Billenniun	m ? AP
Sandbox / Add API	🕕 Select AM — 😨 Configure URL — 🏐 Provide authentication — 🜒 Preview Schema — 🛞 Edit Schema		
customer-bank-exp v1 1.0.0		e API documentation	API apecification
	//// 🐵 API schema preview mode ////		
API schema Search API Schema	e+ Query QueryType Type description *		
23 All types in one Aprischema V	Query methods ③		
Query	Q Search methods	L.	🔓 Expand All
	Method rame 💿		
	> customer (): Any		
	> customerByCustomerId (customerIditring); lyps		
	Cancel	Back Nex	it: Edit Schema

→ Step 7:

In the edit schema section you can hide and enable methods as per your requirement. Click on Next: Add to unified schema.

🚍 🥎 DataGraph		취 Billennium 🤶 AP
Sandbox / Add API	(1) Select API — (2) Configure URL — (3) Provide authentication — (4) Preview	w Schema 🔲 🌀 Edit Schema
customer-bank-exp vi 1.0.0		e ^o <u>ADI documentation</u> e ^o <u>ADI specification</u>
() API schema Q Search API Schema	••• Query Ouerafites Type description •	
S All types in this API schema v	🌮 Edit methods settings	
Query Query	Q Search methods	🛓 Expand All
	Method name (i)	Desired state ①
Level 1 types	> customer (): Any	Visible Rename method
	> customerByCustomerId (oustomerIdiOtring)); Type	Visible Rename method
		Cancel Back Next: Add to unified schema

→ Step 8:

It may take up to 15 minutes to update the unified schema.

😑 🌸 DataGraph				ती Billennium ? AP
United Data Graph Sandbox	(2) Up to dote Unified scheme			(1) Run a Query: + Add API
e- Hide side panel	Q Unified Schema	••• Query Countrie Type description •		
52 Overview			_	
() Unified Schema				
List of APIs added	Query	We are updating the unified schema to reflect this change. This may take up to 15 minutes if there are other ongoing changes to the unified schema		
(f) Response logs		③ Note! Changes are instantly available in the unified schema to view or edit. However, to query the new changes, the update must finish.		
		Once the update is done, unified schema status will change to 🖉 Up to date		
		Proceed to unified schema		
		You may leave this dialog without disrupting the update		

→ Step 9:

Once it is completed, you can see the up to date status as shown below and click on Run a Query.

☴ 🤝 DataGraph			🖞 Billennium 💡 AP
Unified Data Graph Sendbox	2 Up to date Unified schema		(;) Run a Query + Add API
← Hide side panel	() Unified Schema	••• Query Cover/type Type description •	
En Overview	Q Search API Schema		
🕅 Unified Schema			
Ⅲ List of APIs added	Query	Query methods ()	
(> Response logs	Туре	Q Search methods	🛓 Expand All
		Method name (3	
		> customer (): Any	
		<pre>> customerByCustomerId (customerId(String)): Type</pre>	

→ Step 10:

You can select an existing application or create a new application and click on Request Access.



→ Step 11:

After application request access, your graph query console will open, and you can start to write and validate your query here.



According to your requirement, you can add more fields.



GraphQL Implementation with Mulesoft:

Prerequisites:

- Anypoint Studio 7
- JDK 1.8
- Postman

Firstly, you must download the graphql router that is available on GitHub as part of open source.

Go to: https://github.com/mulesoft-labs/graphql-router

Copy the clone URL or download a .zip file into your local system.

	O tags	Go to file 👱 🖞	Lode *	, ibout
🤹 juancavallotti Merge pull req	uest #1 from ymenager/fix-build	E Clone HTTPS GitHub CLI	?	Router to expose services as graphql
icon	Updated icon to make it smaller an	https://github.com/mulesoft-labs/graphql-r	2	
STC .	Updated the connector UX to work	Use Git or checkout with SVN using the web URL.		Releases
gitignore	Updated icon to make it smaller an	[1] Open with Citlinh Desister		No releases published
README.md	Initial commit	G Open with Github Desktop		
D pom.xml	Forced 2.x versions of maven-deplo	Download ZIP		Packages
README.md				No packages published
README.md Mule-module Add description	-graphql Extension			No packages published Contributors 2 juancavallotti Juan Alberto Lopez Ca transfer ymenager

After cloning or downloading the project, go to the pom.xml file and change the mule-modules-parent version from 1.0.0 to 1.1.3.

Add below plugin repository into your pom.xml file.

Go to the location where the project is cloned or downloaded and then go to command prompt and run command:

<pluginRepositories>

After Building Success, let's get started...!

→ Step 1:

Create a sample Mule project in the Anypoint studio and add the below dependency in the pom.xml file for the GraphQL module.

<dependency> <groupId>com.mulesoft.services</groupId> <artifactId>mule-module-graphql</artifactId> <version>1.0.0-SNAPSHOT</version> <classifier>mule-plugin</classifier> </dependency>

→ Step 2:

Create a sample REST API for account, balance, and transaction details.

customer-balance-sys API

v customer-ba	alance-sys-api
Listener	Transform Message Account balance data
Message Flow	Global Elements Configuration XML
General	There are no errors.
MIME Type	Display Name: Listener
Redelivery	Basic Settings
Responses	Connector configuration: HTTP_Listener_config
Advanced	General
Metadata	Path: /api/accounts/{accountld}/balance
Notes	



customer-account-sys API

▼ customer-a	ccount-sys-api
	$\rightarrow \bigotimes$
€ Listener	Transform Message
	Customer account details
Message Flow	Global Elements Configuration XML
Cistener ×	Console Problems
	S There are no errors.
General	
MIME Type	Display Name: Listener
Redelivery	Basic Settings
Responses	Connector configuration: HTTP_Listener_config
Advanced	General
Metadata	Path: /api/accounts/{accountId}
Notes	



customer-transaction-sys API

v customer-tra	ansaction-sysFlow
Message Flow	Global Elements Configuration XML
l istener v	
General	V There are no errors.
MIME Type	Display Name: Listener
Redelivery	Basic Settings
Responses	Connector configuration: HTTP_Listener_config
Advanced	General
Metadata	Path: /api/accounts/{accountld}/transactions
Notes	



→ Step 3:

Now we will create a customer-bank-exp API.

→ Step 4:

Go to /src/main/resource, create a new folder and name it schema.

→ Step 5:

Right click on the schema Folder under /src/main/resources New \rightarrow File.



→ Step 6:

Create customer-bank-exp-api flow as shown below.

customer-back customer-back Listener Error handli	ank-exp-api Router ng
Message Flow	Global Elements Configuration XML
Cistener ×	Console 🔝 Problems
General	⊘ There are no errors.
MIME Type	Display Name: Listener
Redelivery	Basic Settings
Responses	Connector configuration: HTTP_Listener_config
Advanced	General
Metadata	Path: /api/graphql/customer
Notes	
Heln	

customer-bank- Listener Frror handling Message Flow Glot	exp-api Router bal Elements Configuration XML
🕸 Router 🗙 🖳 C	Console 🖹 Problems
General	⊘ There are no errors.
Advanced	Display Name: Router
Error Mapping	Basic Settings
Metadata	Module configuration: Graphql_Config
Notes	General
Help	Payloads Expression
	fx #[payload

→ Step 7:

Configure GraphQL in global configuration as shown below.

Solobal Element Properties	×
Graphql Config Default configuration	
General Advanced Notes Help Name: Graphql_Config General Config name: Argraphql Schema location: Argraphql Schema location: Argraphqls	
ОК	Cancel

→ Step 8:

Create customer account flow as show below and configure GraphQL field resolver with the field name: account.

• customer-a	ccount-flow	→ () –		
Graphql fie resolver	eld Execute	Request	Transform Message	Logger
 Error handli Message Flow Graphql fiel 	ing Global Elements C d resolver × QC	Configuration XML	ems	
General	There are no er	rors.		
Redelivery	Display Name:	Graphql field resolv	ver	
Metadata	Module configu	ration: Graphql_	Config	
Notes	General			
	Field name:	account	-	
Help	Response body:	fx 1 pay	load	
Help	Response body:	fx 1 pay	load	

→ Step 9:

Configure the execute component. If you are not able to find the Groovy engine inside the connector then change the scripting version 1.1.3 in the pom.xml file.

▼ customer-accoi	nt-flow	
-		
Graphql field resolver	Execute Request Transform Calling Message account system api	Logger
Message Flow Glo	bal Elements Configuration XML	
@ Evenite v		
General	There are no errors.	
Advanced	Display Name: Execute	
Error Mapping	General	
Notes	Code:	
Help	payload = ['params': attributes.getArgume	nts()]
	Engine: <i>fx</i> Groovy	
	Parameters: fx	

→ Step 10: Configure the request component.

 customer-accou 	unt-flow	
\$ -\$		
Graphql field resolver	Execute Request Transform Logger Calling Message account system api	
lessage Flow Glo	bal Elements Configuration XML	
Calling account	system api 🗶 📮 Console 🛛 🔝 Problems	
	There are no errors.	
General	Display Name: Calling account system api	
MIME Type	Basic Settings	
Request	Configuration: HTTP_Request_config_for_account	 4 Z
Response	http://localhost:8083/api/accounts/{accountId}	
Advanced	Request	
Error Mapping	Method: <i>fx</i> GET (Default)	~
Metadata	Path: /api/accounts/{accountId}	
Notes	URL: fx	
Help	Body Headers Query Parameters URI Parameters	
	<u>fr</u> • × ×	
	Name	Value

 custo 	mer-account	t-flow			
Z	-\$	→ @ -	→ 💮 —		
Grap	hql field	Execute	Request	Transform Message	Logger
			account system api	menage	
Vessage	Flow Globa	I Elements Confi	guration XML		
🛯 Trans	form Messa	ge 🗙 📮 Consol	e 🕄 Problems		
2	Output Pay	load ▼ =+ 🖍	Ĩ.		
Pay Var 🗸 Att	1⊖%dw 2 2 outpu 3 4 paylo	.0 t application ad	n/java		

For Use Case 1:

Go to the postman and hit graphql endpoint. Select body \rightarrow GraphQL.

GET ~ http://localhost:8081/api/graphql/customer	Send ~
Params Authorization Headers (8) Body Pre-request Script Tests Set	tings Cookie: No Schema 😪 C
QUERY	GRAPHQL VARIABLES ①
1 query 2 { 3 account(id:"123") 4 firstName 5 lastName 6 } 7 }	1
Rody Cookies Headers (3) Test Results Pretty Raw Preview Visualize JSON V	Status: 200 OK Time: 57 ms Size: 219 B Save Response
1 * * * * * * * * * * * * * * * * * * *	

Now Change the fields according to the requirement.

GET	http://localhost:8081/api/graphql/customer	Send v
Params	Authorization Headers (8) Body Pre-request Script Tests Settings	Cookies
none	e 🖲 form-data 🕘 x-www-form-urlencoded 🔍 raw 🔍 binary 🛞 GraphQL No Schema \vee C	
QUERY	GRAPHQL VARIABLES ①	
1 2 3 4 5 6 6 7 8 Body C Pretty 1 2 3 4 5 6 6 7 8	query 1 i account(id:"123")] firstName firstName lastName emeil mobileNo i i mobileNo i i i mobileNo i i <	re Response 🗸
10		1

For Use Case 2:

Add the fields inside the schema under src/main/resources/schema folder.

customer.graphqls 🛛	∀customer-bank-exp	∀ customer-balance-sys	∀ customer
1 schema { 2 query: Query 3 } 4 5 type Query{			
6 account(7 id: Stri 8): Account	ng		
10 11 type Account{ 12 id: String			
13 firstName:St 14 lastName:Str 15 mobileNo:Str 16 email:String	ring ing ing		
17 address:Stri 18 pincode:Stri 19 accType:Stri	ng ng		
20 balances:[Ba 21 transactions 22}	lance] :[Transaction]		
<pre>23 24type Balance{ 25 balanceType: 26 amount: Floa 27} 28</pre>	String t		
29 type Transaction 30 transId: Str 31 transType: S 32 transAmount: 33 transDate: S 34 }	{ ing tring Float tring		

• Add the balance flow as shown below.

customer-based	alance-flow		
Graphql fie resolver	ld Execute	Request Calling balance system api	Transform Message
		antinumbion VMI	
viessage Flow	Global Elements Co	onfiguration XML	
Craphql field	I resolver × 🖳 Co	nsole 🖹 Problems	
General	There are no error of the second s	rors.	
Redelivery	Display Name: G	raphql field resolver	
Advanced	Basic Settings		
Metadata	Module configu	ration: Graphql_Cont	ig
Notes	General		
Help	Field name:	balances	
	Response body:	fx 1 payloa	d

• Add the following script.

customer-balance	e-flow			
Graphql field resolver	Execute			
lessage Flow Glob	al Elements Configuration XML			
Execute ×	Console 🔝 Problems			
General	There are no errors.			
Advanced	Display Name: Execute			
Error Mapping	General			
Notes	Code:			
Help	<pre>payload = ['params': attributes.getArguments(), 'source': attributes.getSource()]</pre>			
	Facine R. Coorre			
	Parameters: fx			

Configure the request component as below.

 customer-balance 	ce-flow				
Graphql field resolver	Execute	Request Calling balance system api	→ Transform Message		
Message Flow Glob	oal Elements Configur	ation XML			
Calling balance s	system api 🗙 📮 Con	isole 🔝 Problems	5		
	There are no error	rs.	2		
General	Path: //api/accounts/laccountid}/balance				
MIME Type					
Request		1 <u>×</u>			
Response	Body Headers	Query Parameter	s URI Parameters		
Advanced	fx 💠 💥 🔆				
Error Mapping	Name			Value	
Metadata	"accountio	d"		payload.source.id	
Notes					
Help	Send correlation in	d: fx Empty			
	Correlation id:	fx			

• Add the transaction flow as show below.

 customer-tr 	ansactions-flow		
		\rightarrow	\rightarrow
'≓' Graphql fie resolver	eld Execute	Request Calling transaction system api	Transform Message
Vessage Flow	Global Elements Co	nfiguration XML	
🕸 Graphql field	d resolver 🗴 📮 Cor	nsole Problems	
General	There are no error	ors.	
Redelivery	Display Name: Gra	aphql field resolver	
Advanced	Basic Settings		
Metadata	Module configura	tion: Graphql_Config	9
Notes	General		
Help	Field name:	transactions	
	Response body:	fx 1 payload	

• Add the following script as shown below.

▼ customer-transa	sactions-flow			
\$ -\$	$\rightarrow \textcircled{} \rightarrow \textcircled \rightarrow \textcircled{\Box} \rightarrow \Box \rightarrow$			
Graphql field resolver	Execute Request Transform Calling Message transaction system api			
Message Flow Glo	obal Elements Configuration XML			
œ Execute ×	Console 💽 Problems			
General	There are no errors.			
Advanced	Display Name: Execute			
Error Mapping	General			
Notes	Code:			
Help	<pre>payload = ['params': attributes.getArguments(), 'source': attributes.getS</pre>	ource()]		
	Entry R. Count			
	Engine: JA Groovy			
	Parameters: Jx			

• Configure the request component as shown below.

 customer-transa 	ctions-flow						
Ø -							
∉ Graphql field resolver	Execute Request Transform Calling Message transaction system api						
Message Flow Glo	oal Elements Configuration XML						
Calling transacti	on system api 🗴 🖳 Console 🔝 Problems						
General	O There are no errors. Usplay Name: Calling transaction system api						
MIME Type	Basic Settings						
Request	Configuration: HTTP_Request_config_for_transaction v						
Response	http://localhost:8084/api/accounts/{accountId}/transactions						
Advanced	Request						
Error Mapping	Method: JA GET (Default)	GET (Default) /api/accounts/(accountId)/transactions					
Metadata	Path: JX /api/accounts/{accountId}/transa						
Notes	URL: Jx						
Help	Body Headers Query Parameters URI Parameters						
	<i>f</i> x						
	Name	Value					
	"accountId"	payload.source.id					

- Go to the postman and hit graphql endpoint.

GET v http://localhost:8081/api/graphql/customer			Send ~
Params Auth Headers (8) Body • Pre-req. Tests Settings	010	Body ~	Dennistra Viennistra 1001 co
COLERY 1 query 2 i 3 account(id:"123'){ 4 firstName 5 lastName 6 email 7 mobileNo 8 transactionsi 9 ttansType 10 ttansDate 12 j 13 balances { 14 amount 15 balanceType 16 j 177 j 18 j		Pietry 1 1 2 1 3 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	<pre>"data": { "data": { "data": { "istName": "Arun", "lastName": "Arun", "mobileNo": "898798787", "transactions": [</pre>

This API source code is available \rightarrow here.

References:

GraphQL: https://graphql.org/ Mulesoft: https://anypoint.mulesoft.com/datagraph

Summary:

GraphQL queries are fast and stable because they control the data and get exactly what you need, nothing more, nothing less. GraphQL queries always return predictable results.