**About The Project**

**Built With**

1. Anypoint Platform

2. RAML

3. Apache ActiveMQ

4. JMS

5. AUTH0 by Okta

**Getting Started**

I’ve created a RAML specification in which I have turned on the mocking service to check the API whether it is fetching/updating/deleting all the data by using console on API portal & Postman.

This API serves as a design best practice to work with Staff data via a set of RESTful services that are designed in RAML, making it easily consumable by any developer. This API allows the following:

•List staff  
•Create a new staff member  
•Update a staff member  
•Offboard a staff member

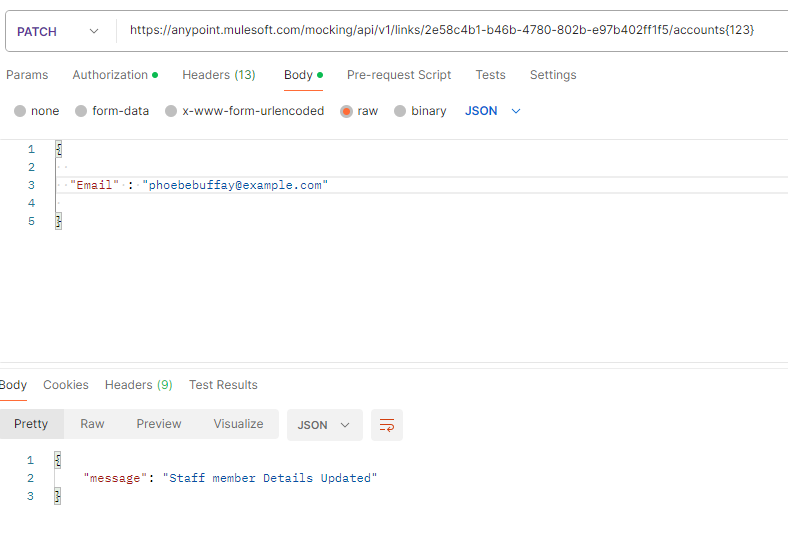
API End Points with the required inputs: first name, last name, email address, role, active/inactive flag:

/accounts:

Methods - GET and POST

/accounts/{id}:

Methods - GET, Patch, Delete



**FYI** - I have added delete method for offboarding a staff member but usually data is not deleted instead it is updated using Patch (a.k.a soft delete).

**Use Cases:**

1. **A client can periodically consume the API to enable it (the client) to maintain a copy of the provider API's staff records**

Client will query the Mule API to get the list of all staff members by using GET initially to maintain a copy of whole data on their end.

**There are 2 scenarios to achieve this**:

1. We can use DB connector’s on **table row** & specify watermark column in it.

**Solution** - In the general section of the connector, we’ll specify the column that we have to watermark. It’s usually the one which is increasing when the new records are added. According to the watermark level, it’ll pick up the latest records. For the first time, it’ll take all the records & will insert it into the downstream system & will watermark it. So, next time when our polling will run it’ll use that watermarking & from that record it’ll take the remaining records.

1. We can use **Object store** & configure the query on basis on Timestamp in a periodic way (every 5-10 mins) using scheduler, so that it returns the any new update on the record post that Timestamp.

Explanation – We can configure “**objectstore:retrieve**” which will retrieve the last inserted timestamp from the Object Store. Post this we have to query API to fetch new records based on the timestamp & “**objectstore:store”** will update the object store with the current timestamp.

1. **The API should provide an endpoint that allows the HR system to notify the API of a changed record(s), this can be dispersed to any interested listeners**

This can be achieved using JMS connector. For this we need to have an Apache MQ account where we will create a queue with a name “Staff record update”. We’ll add the queue name in JMS connector & select the destination type as topic & publish a message “data changed”. All the active subscribers will receive this message & can use the API to consume/fetch the updated records.

1. **The creation and removal of a staff member requires a special privilege that only certain applications can have (such as a recruitment system notifying the HR system of a new joiner)**

This can vary based on your specific use case and security requirements. I have used JWT validation policy & applied it to the POST & DELETE method in my RAML specification. For that I’ve created an account in **Auth0** & configured it accordingly by creating an application in it. So if someone wants to access the API, they first have to generate a **bearer token** and then have to pass the **token** in the Header for authorisation.

Similarly you can use OAuth 2.0 Access Token Enforcement using Mule OAuth Provider.

