[**Topcoder Member Service - Add Additional Traits v1.0**](https://apps.topcoder.com/forums/?module=Category&categoryID=43403)

**Verification Guide**

# 

# Setup

Following the LocalSetup.docx to setup the enironments.

This challenge add the follow command to local/aws-cli/init-dynamodb.sh

# Create MemberProfileTrait table

aws dynamodb create-table --table-name MemberProfileTrait --attribute-definitions AttributeName=userId,AttributeType=N AttributeName=traitId,AttributeType=S --key-schema AttributeName=userId,KeyType=HASH AttributeName=traitId,KeyType=RANGE --global-secondary-indexes '[{"IndexName":"traitId-index","KeySchema":[{"AttributeName":"traitId","KeyType":"HASH"}], "Projection":{"ProjectionType":"ALL"}, "ProvisionedThroughput": {"ReadCapacityUnits": 2, "WriteCapacityUnits": 2}}]' --region us-east-1 --provisioned-throughput ReadCapacityUnits=4,WriteCapacityUnits=2 --endpoint-url http://$IP:7777

It's used to create the table MemberProfileTrait to store the member profile trait data.

Open command line terminal, in ap-member-microservice/service

$mvn clean compile package -DskipTests

$java -jar target/member-microservice-\*.jar server src/main/resources/member-service.yaml

Then enter to the ap-member-microservice-dev/local/aws-cli, run the following command to insert the data for the user heffan(please check LocalSetup.docx for the environment setup):

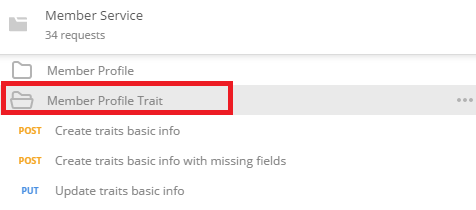
aws dynamodb batch-write-item --request-items file://heffan.json --endpoint-url http://$IP:7777

# Verify

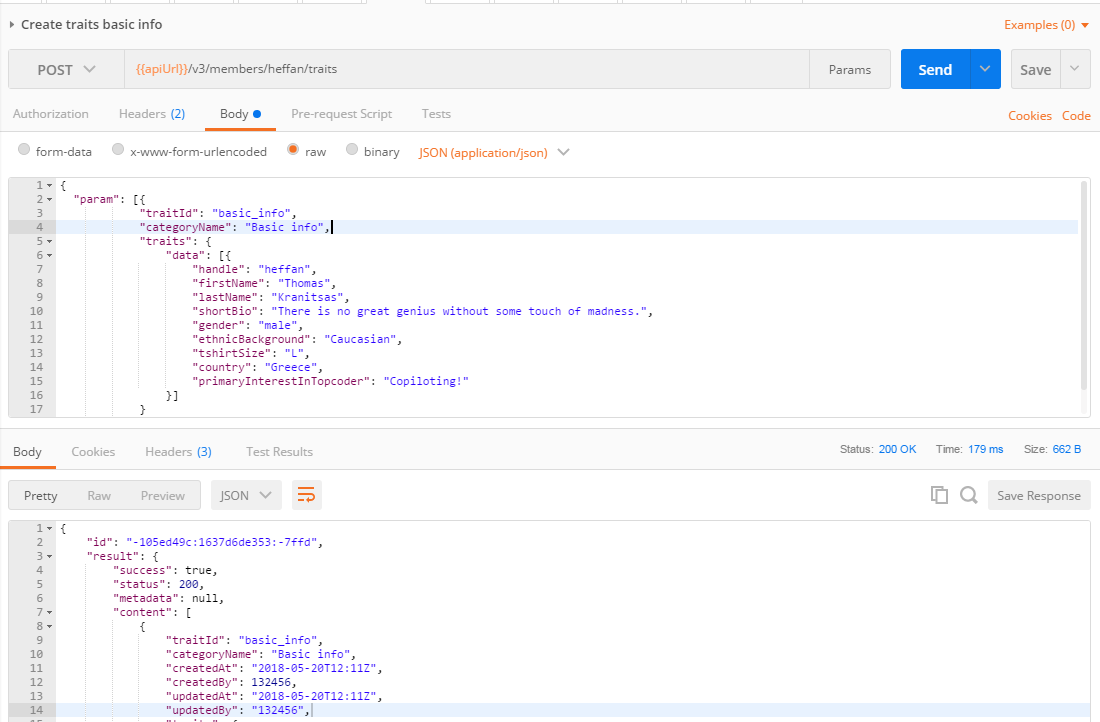
Import the docs/postman/Member Service.postman\_collection.json to the postman.

Member Service.postman\_environment.json should also be imported as environment.

Click the Memeber Profile Trait sub folder



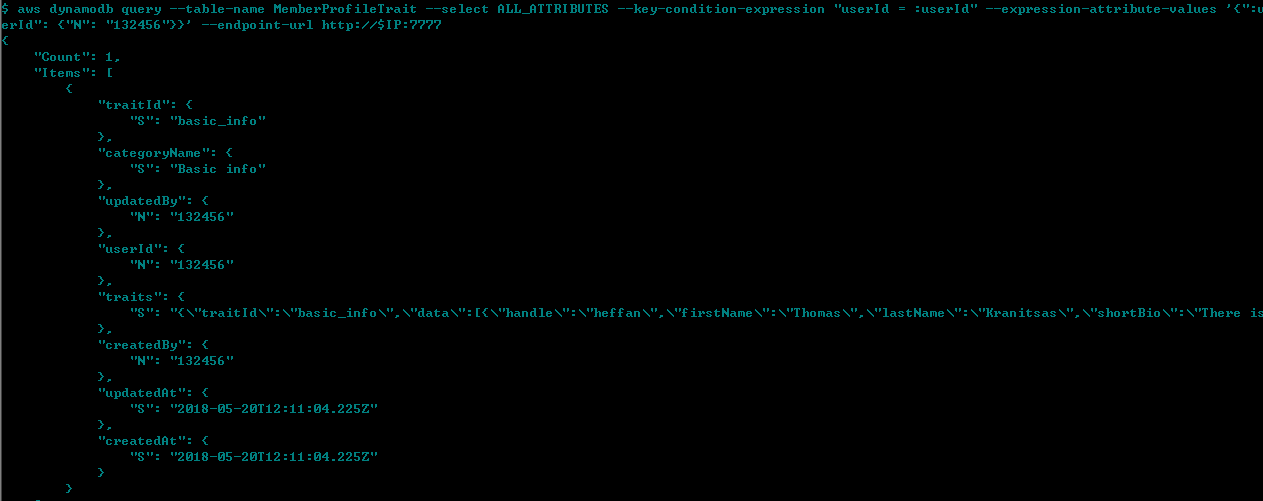
Create traits basic info



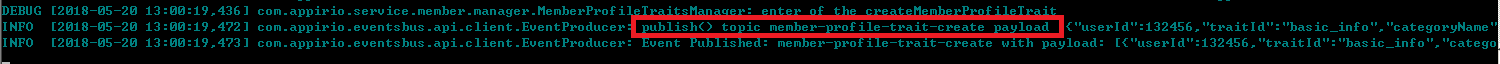
Then check the database.

enter to the ap-member-microservice-dev/local/aws-cli, run the following command:

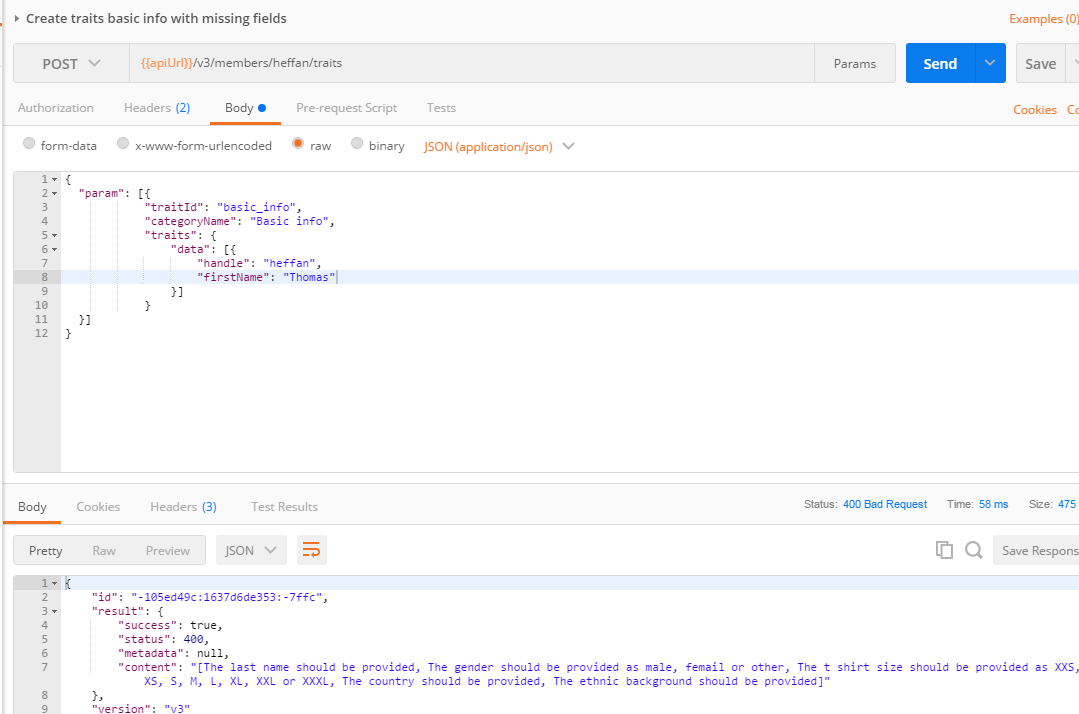
aws dynamodb query --table-name MemberProfileTrait --select ALL\_ATTRIBUTES --key-condition-expression "userId = :userId" --expression-attribute-values '{":userId": {"N": "132456"}}' --endpoint-url http://$IP:7777



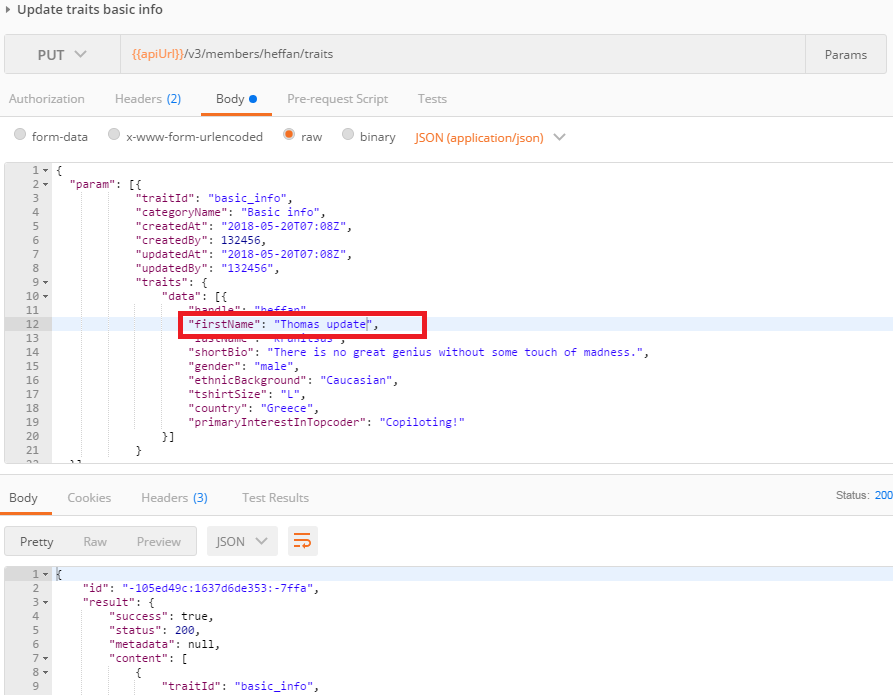
The create operation will also publish the event via kafka, check the logging info in the console:



Create traits basic info with missing fields



Update traits basic info



enter to the ap-member-microservice-dev/local/aws-cli, run the following command:

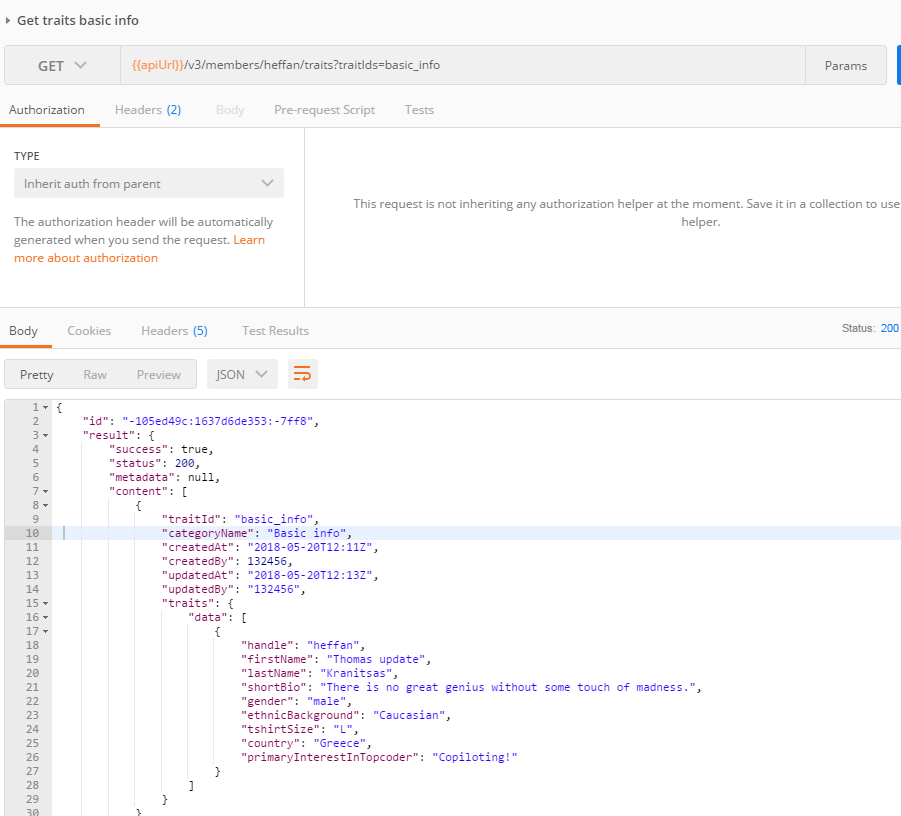
aws dynamodb query --table-name MemberProfileTrait --select ALL\_ATTRIBUTES --key-condition-expression "userId = :userId" --expression-attribute-values '{":userId": {"N": "132456"}}' --endpoint-url http://$IP:7777



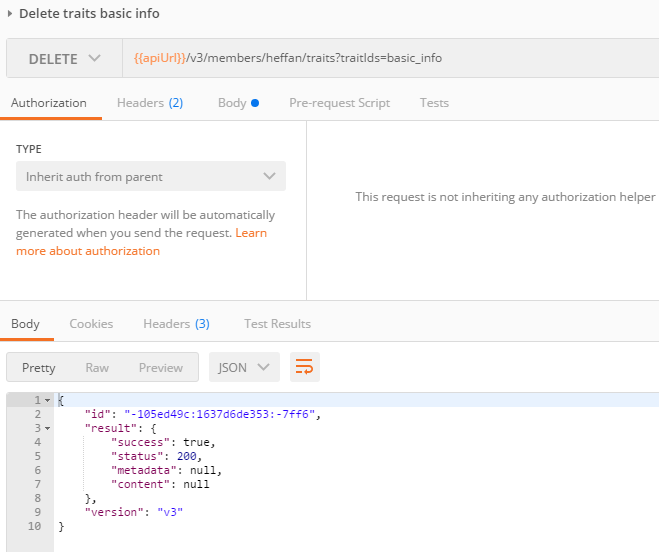
The update operation will also fire the event via kafka:



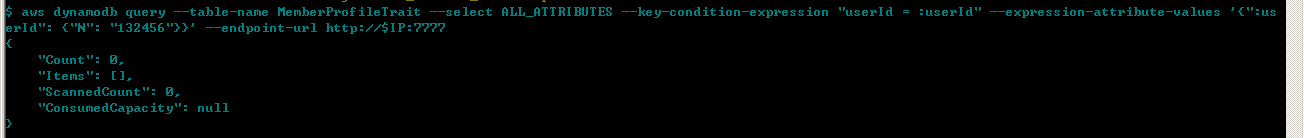
Get traits basic info



Delete traits basic info



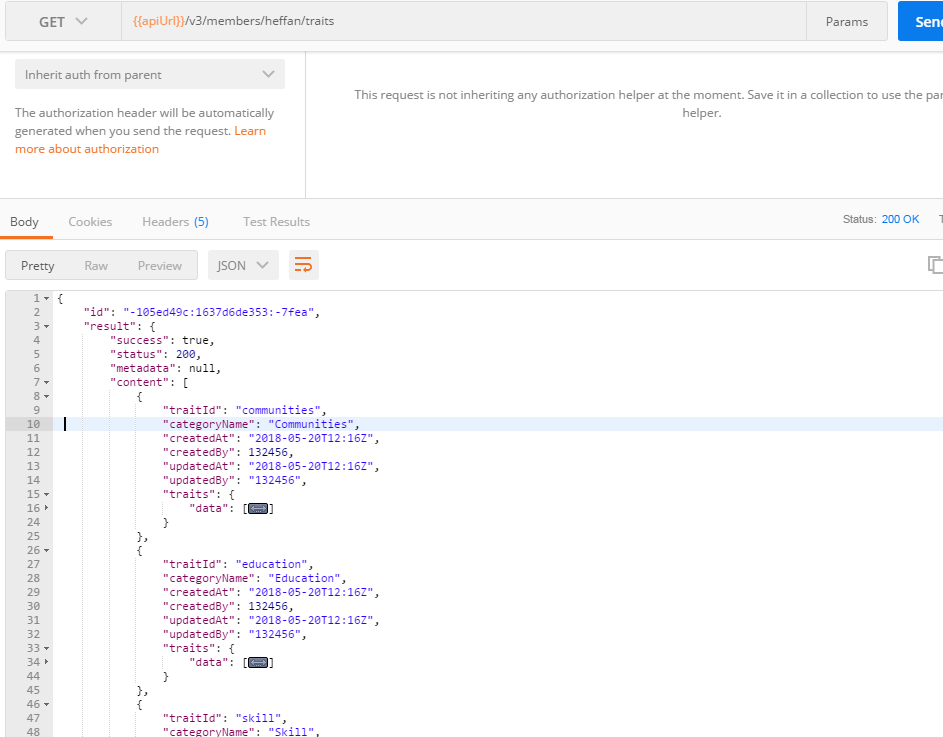
check the database with the same command as create/update operation



For other trait data(education, skill, work and communities) check them simliarly.

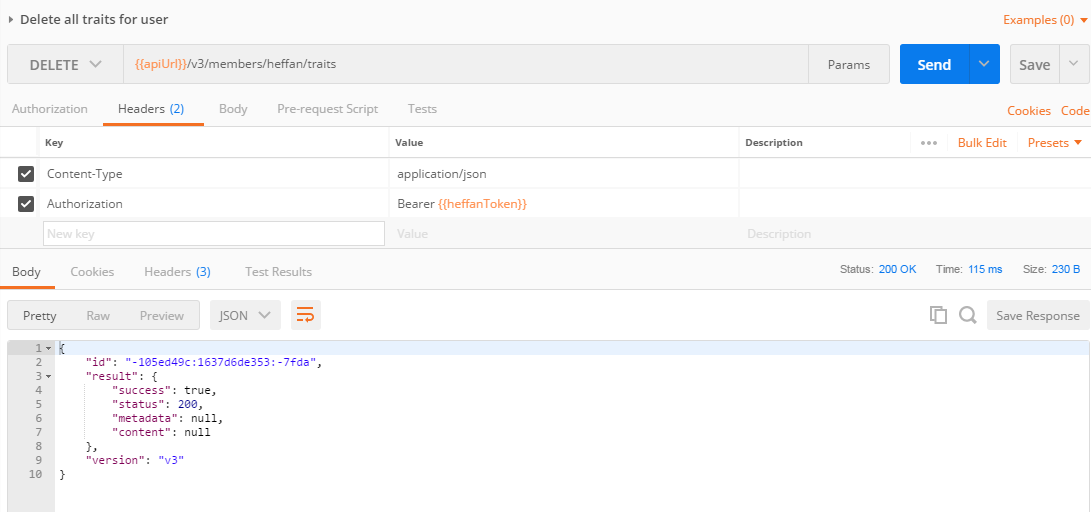
Get all traits for user

This will get all the traits for the user



Delete all traits for user

This will delete all the traits data for the user



After delete all the traits for the user, check Create all traits and Update all traits for the user, it will create all the supported traits for the user.