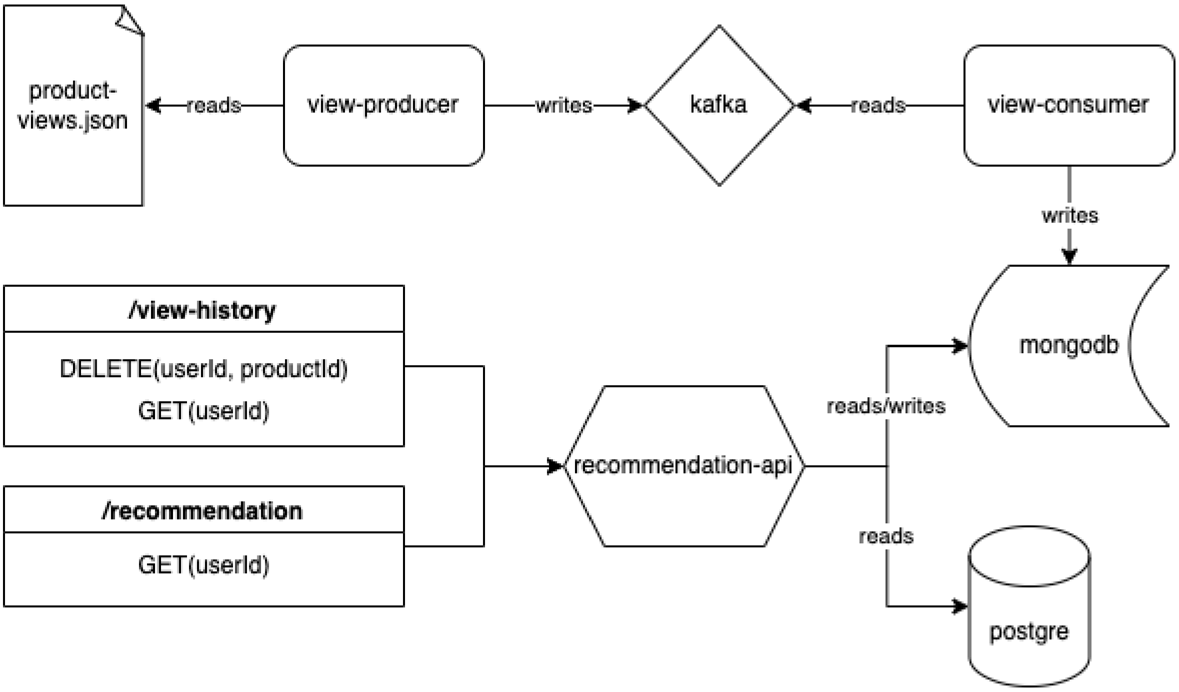
Hepsiburada
Recommendation Team
Software/Data Engineer Assignment Documentation

Ataberk Gürel

System Design



view-producer: It reads a line every second from product-views.json file and sends it to a topic called "views" in kafka.
view-consumer: It reads the "views" topic of kafka and writes every product view event into mongodb.
mongodb: All product view events are stored here.
postgre: It contains products, orders and order_items tables. (No changes made.)
recommendation-api: It can recommend 10 different products depending on the product view history of a user. It can also delete a product from a users view history or return last 10 visited products of a user.

How to run?

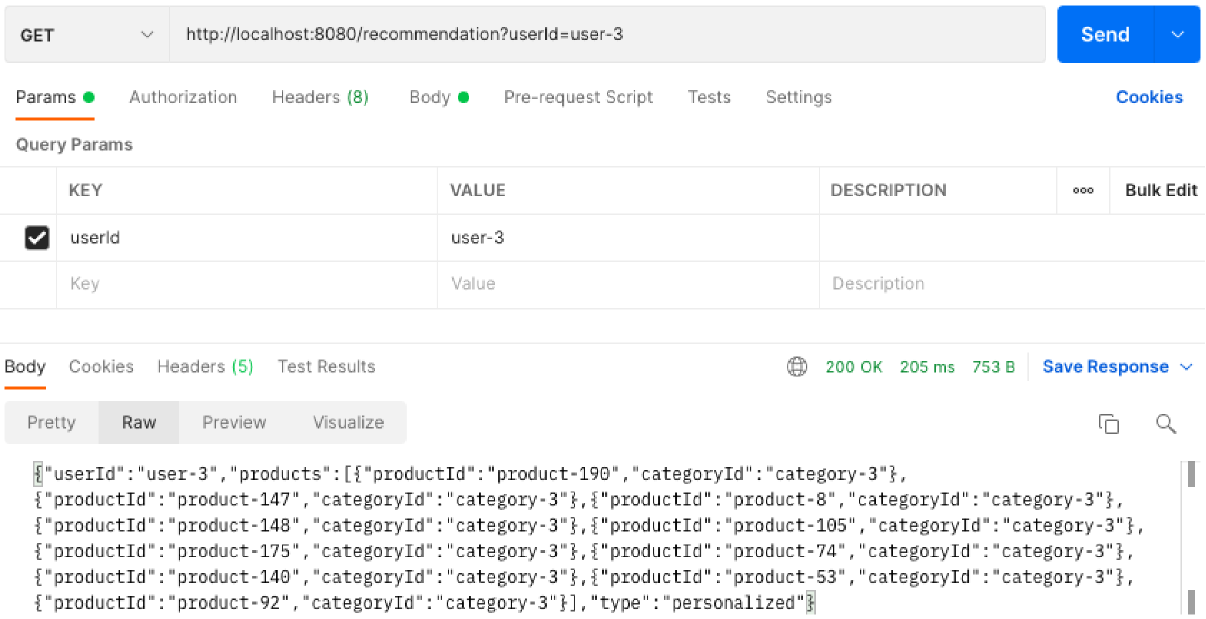
To setup whole system you just have to run following command:

```
docker-compose -f ./docker-compose.yml up -d
```

API Endpoints

<u>/recommendation</u>

GET(userId)



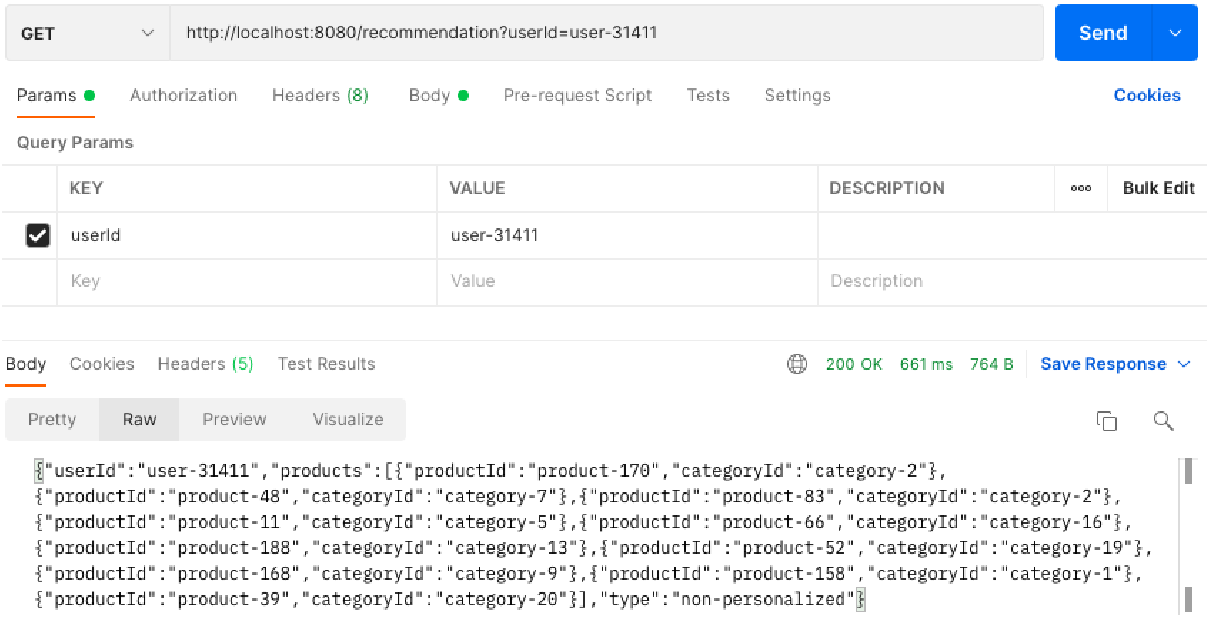
GET http://localhost:8080/recommendation?userId=user-3

	KEY	VALUE	DESCRIPTION
☑	userId	user-3	
	Key	Value	Description

200 OK 205 ms 753 B

```
{"userId":"user-3","products":[{"productId":"product-190","categoryId":"category-3"},
{"productId":"product-147","categoryId":"category-3"},{"productId":"product-8","categoryId":"category-3"},
{"productId":"product-148","categoryId":"category-3"},{"productId":"product-105","categoryId":"category-3"},
{"productId":"product-175","categoryId":"category-3"},{"productId":"product-74","categoryId":"category-3"},
{"productId":"product-140","categoryId":"category-3"},{"productId":"product-53","categoryId":"category-3"},
{"productId":"product-92","categoryId":"category-3"}],"type":"personalized"}
```

For a user that has a view history, it selects 3 categories of the last 10 products visited by this user and recommends top ten bought products by distinct users from these categories.



GET http://localhost:8080/recommendation?userId=user-31411

	KEY	VALUE	DESCRIPTION
☑	userId	user-31411	
	Key	Value	Description

200 OK 661 ms 764 B

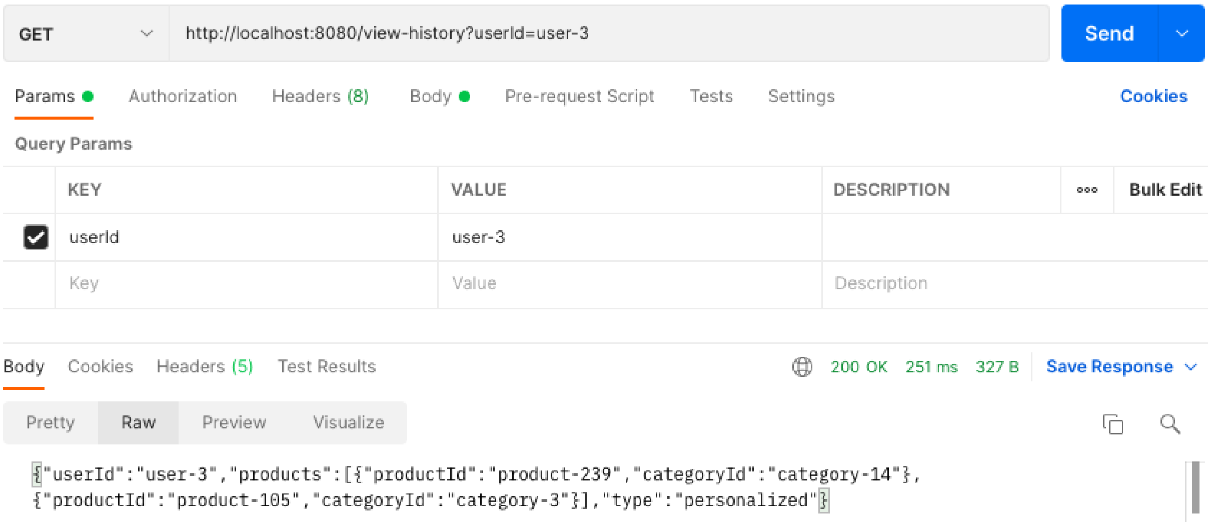
```
{"userId":"user-31411","products":[{"productId":"product-170","categoryId":"category-2"},
{"productId":"product-48","categoryId":"category-7"},{"productId":"product-83","categoryId":"category-2"},
{"productId":"product-11","categoryId":"category-5"},{"productId":"product-66","categoryId":"category-16"},
{"productId":"product-188","categoryId":"category-13"},{"productId":"product-52","categoryId":"category-19"},
{"productId":"product-168","categoryId":"category-9"},{"productId":"product-158","categoryId":"category-1"},
{"productId":"product-39","categoryId":"category-20"}],"type":"non-personalized"}
```

If a user does not have any product view event it will recommend the top ten bought products without considering a category.

You can check the queries at the ProductRepository interface.

/view-history

GET(userId)



GET http://localhost:8080/view-history?userId=user-3

Query Params

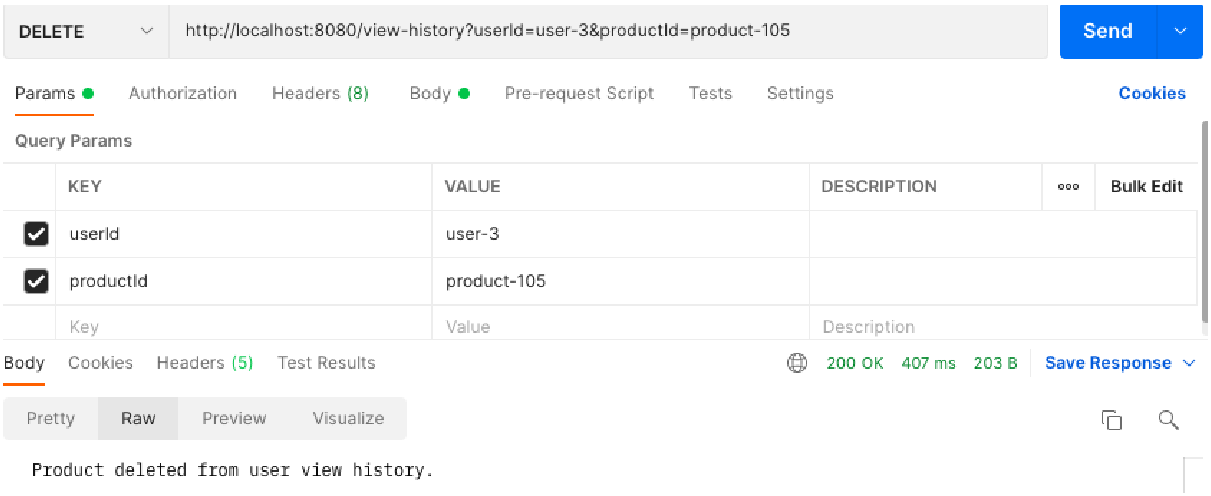
	KEY	VALUE	DESCRIPTION
☑	userId	user-3	
	Key	Value	Description

200 OK 251 ms 327 B

```
{"userId":"user-3","products":[{"productId":"product-239","categoryId":"category-14"},
{"productId":"product-105","categoryId":"category-3"}],"type":"personalized"}
```

GET method returns the last visited 10 products of a user.

DELETE(userId, productId)



DELETE http://localhost:8080/view-history?userId=user-3&productId=product-105

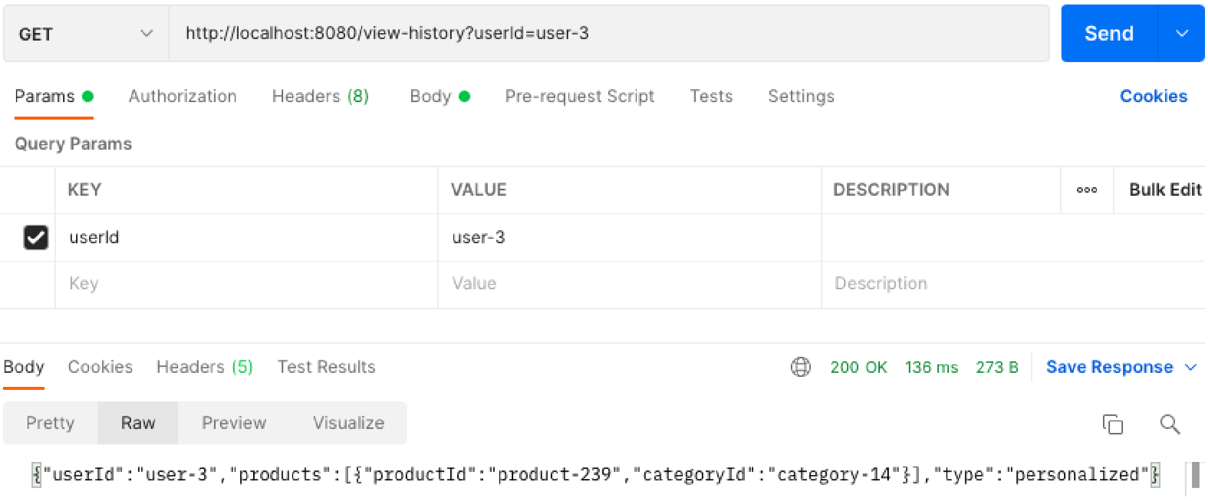
Query Params

	KEY	VALUE	DESCRIPTION
☑	userId	user-3	
☑	productId	product-105	
	Key	Value	Description

200 OK 407 ms 203 B

```
Product deleted from user view history.
```

DELETE method removes every view event of a product from a user's view history.



http://localhost:8080/view-history?userId=user-3

Query Params

	KEY	VALUE	DESCRIPTION
☑	userId	user-3	
	Key	Value	Description

```
{"userId":"user-3","products":[{"productId":"product-239","categoryId":"category-14"}],"type":"personalized"}
```

As you can see, the product-105 gets deleted from the view-history of user-3 after the DELETE method of this endpoint is invoked.