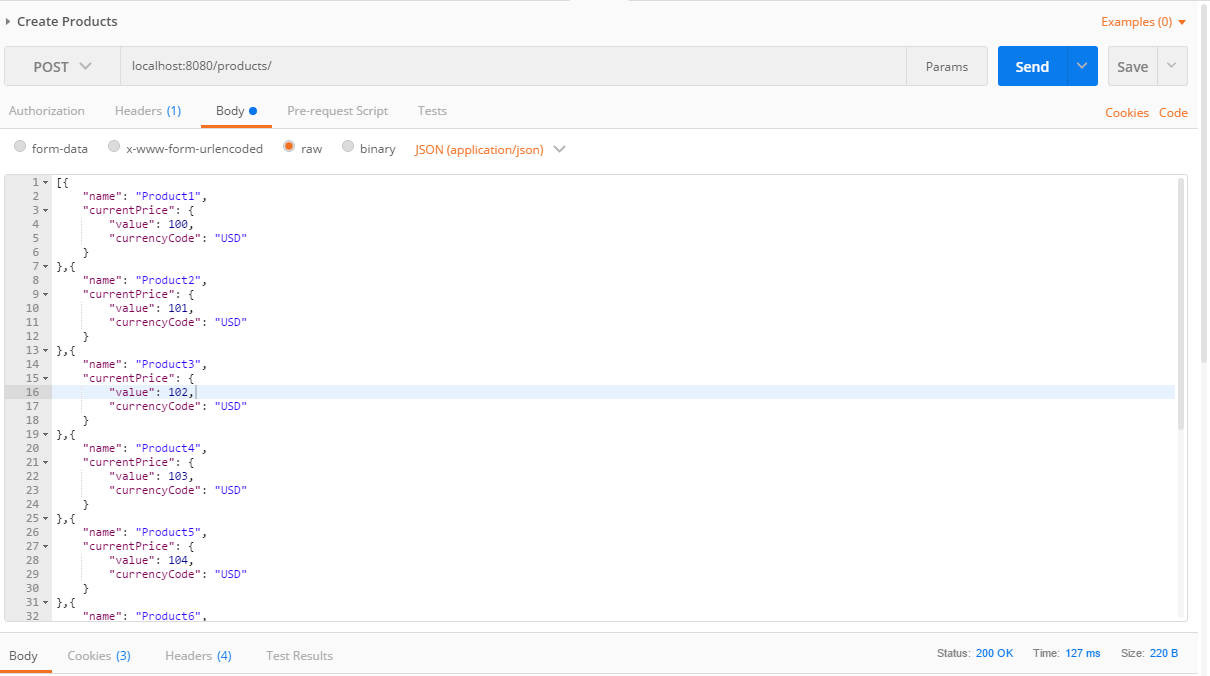
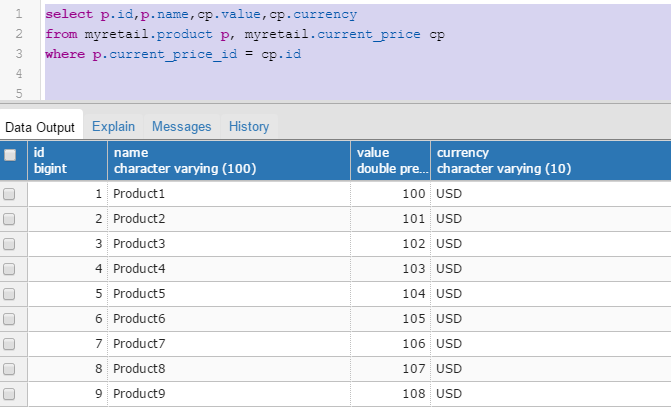
Created a POST API of products creation to facilitate testing for other API’s.

Endpoint : “/products”

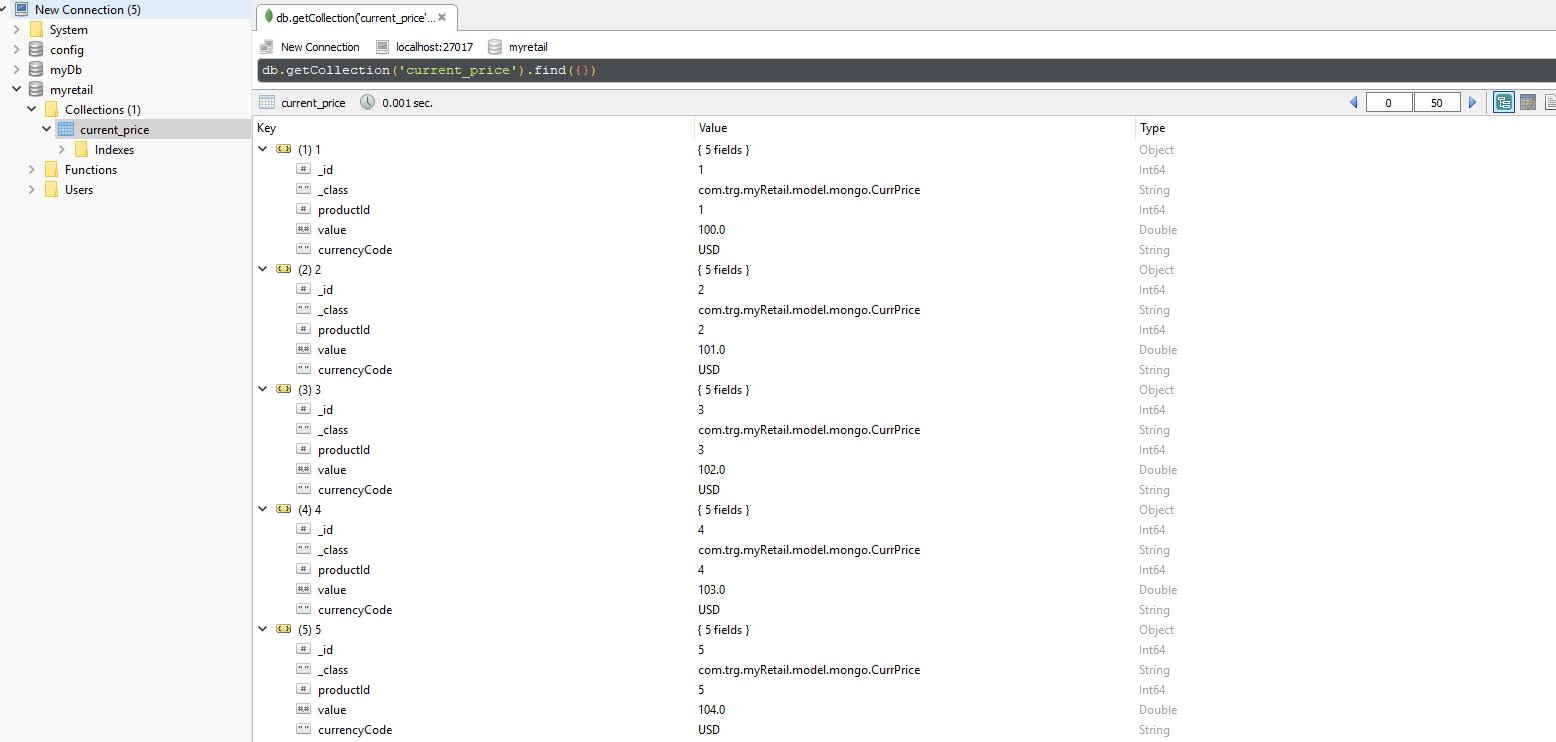
Method : POST



Products Data persisted into postgres table *product* and *currency\_price*



Price Data persisted into MongoDB as well



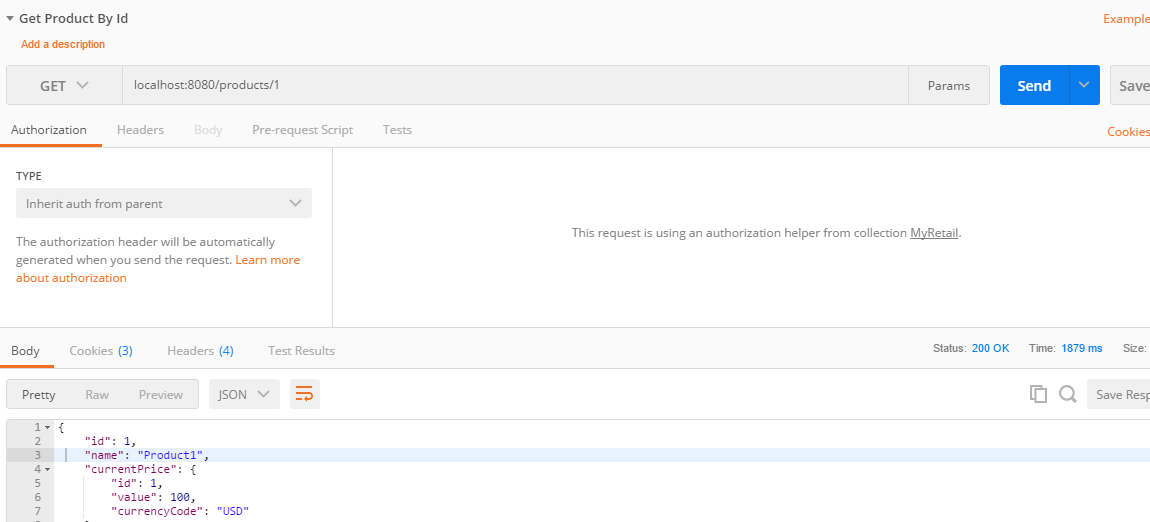
Use Case:: Responds to an HTTP GET request at /products/{id} and delivers product data as JSON (where {id} will be a number.

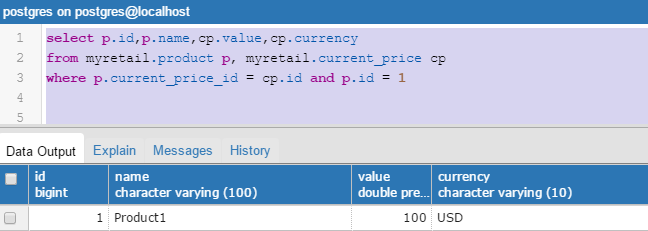
Assumption: Product data is fetched from RDBMS (postgres)

GET API to fetch Product from Product Id

Endpoint: “/products/{id}”

Method: GET





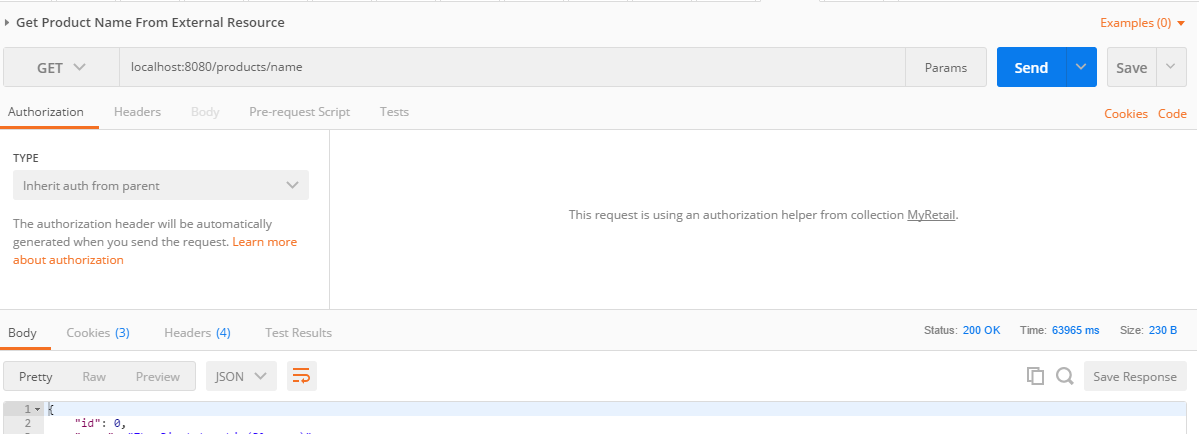
Use case :: Performs an HTTP GET to retrieve the product name from an external API. (For this exercise the data will come from redsky.target.com, but let’s just pretend this is an internal resource hosted by myRetail)

Example : http://redsky.target.com/v2/pdp/tcin/13860428?excludes=taxonomy,price,promotion, bulk\_ship,rating\_and\_review\_reviews,rating\_and\_review\_statistics,question\_answer\_statistics

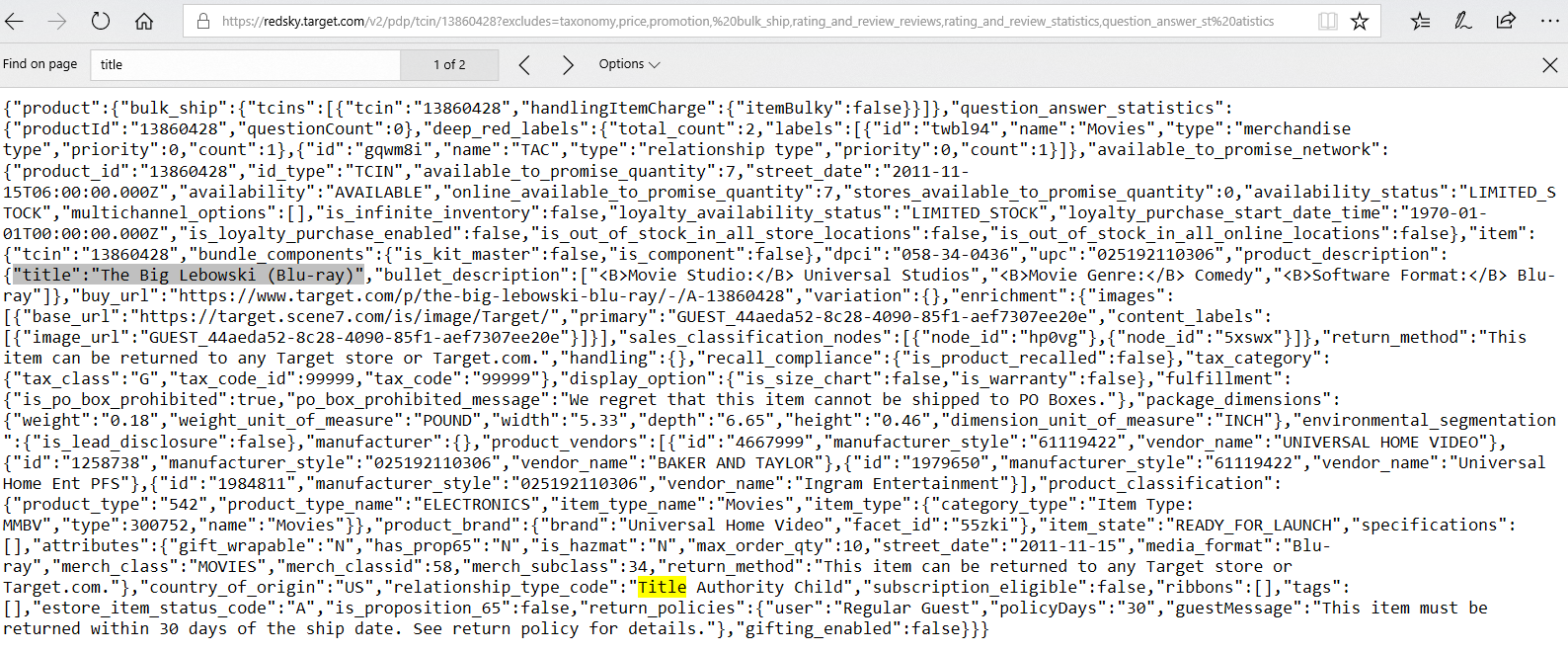
GET API to fetch the product name from external API, hosted by MyRetail

Endpoint : “/products/name”

Method : POST



Product title highlighted below

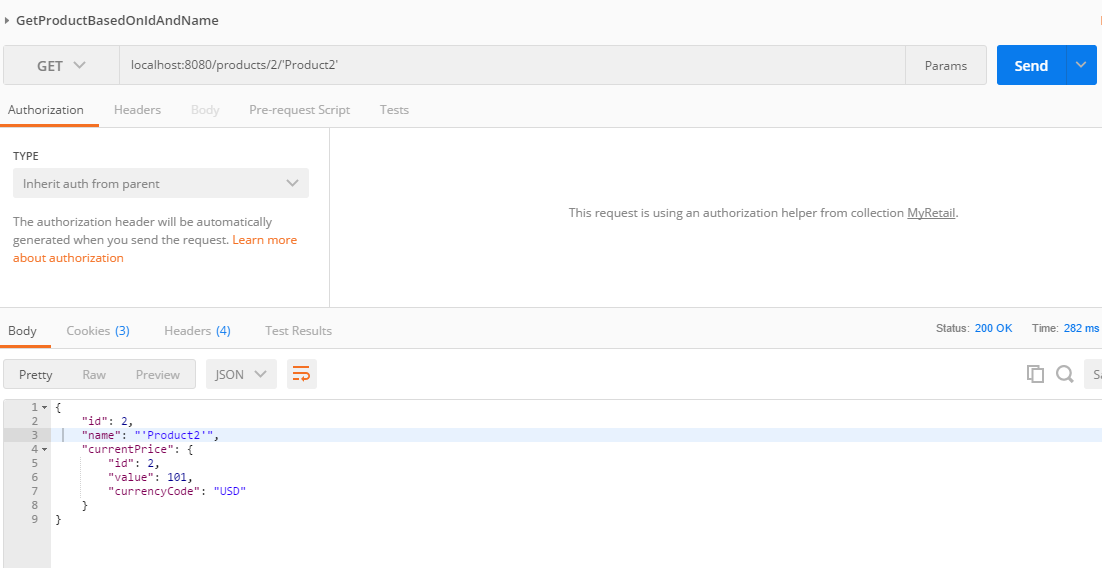


Use Case : Reads pricing information from a NoSQL data store and combines it with the product id and name from the HTTP request into a single response.

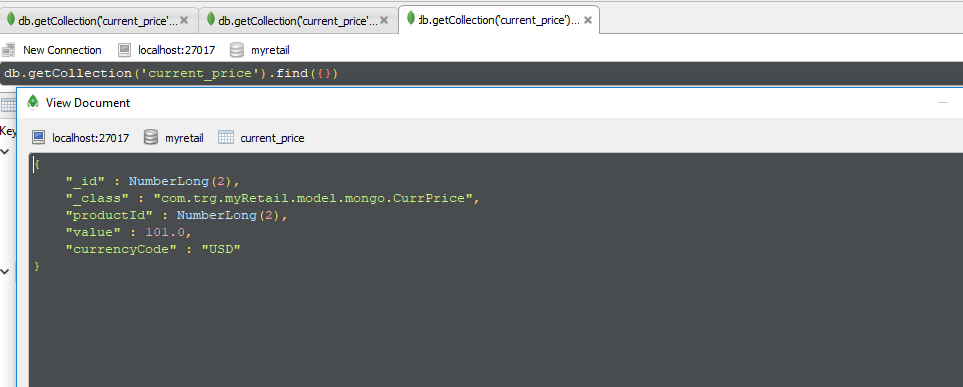
GET API to fetch price information from mongodb based on product id

Endpoint : “/products/{id}/{name}”

Method : GET



Data fetched from mongodb



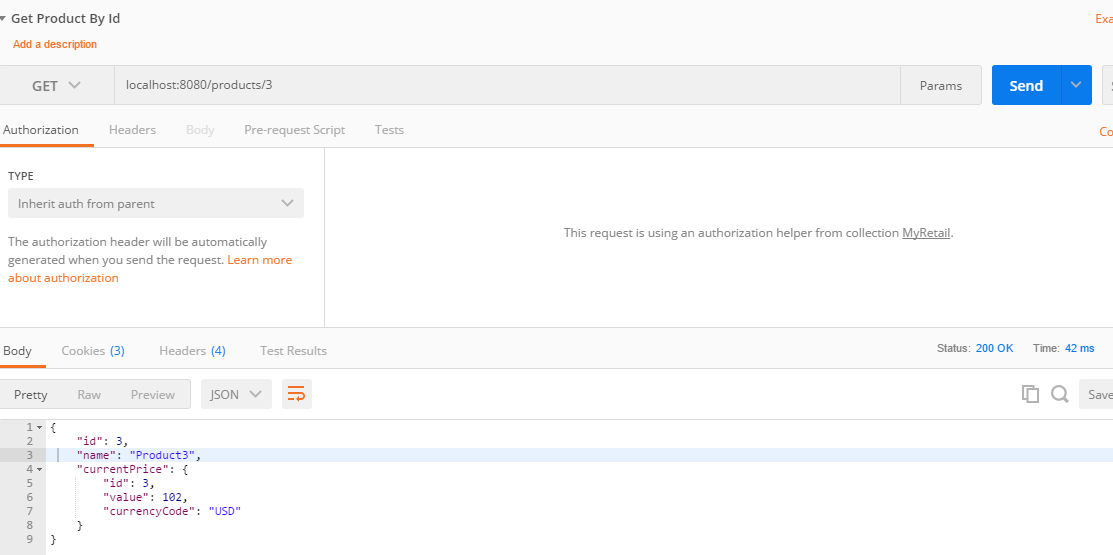
Use Case : Accepts an HTTP PUT request at the same path (/products/{id}), containing a JSON request body similar to the GET response, and updates the product’s price in the data store.

Updates the price info into RDBMS(postgres)

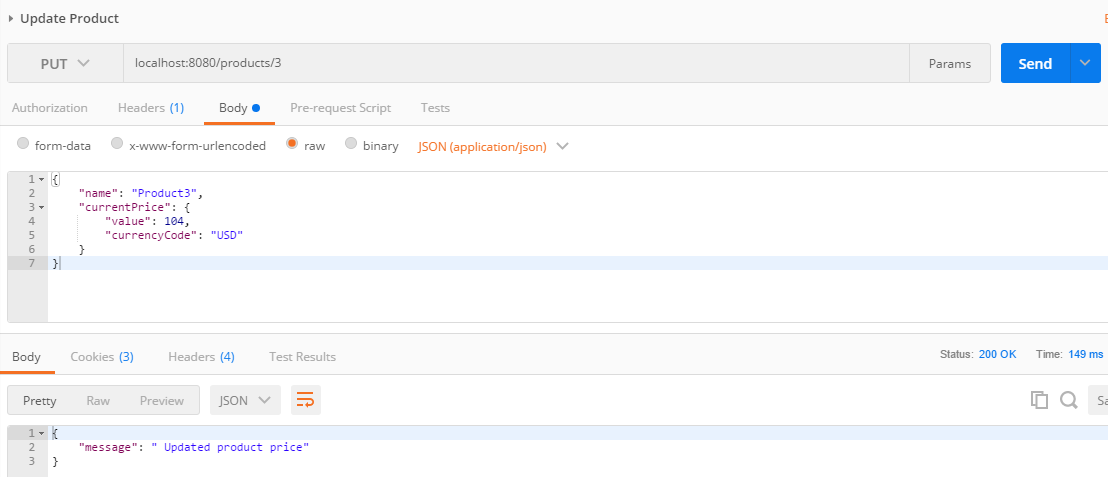
Endpoint : “products/{id}”

Method : PUT

Existing product price



Updated product price



Fetching updated price

