**Enterprise Application Development**

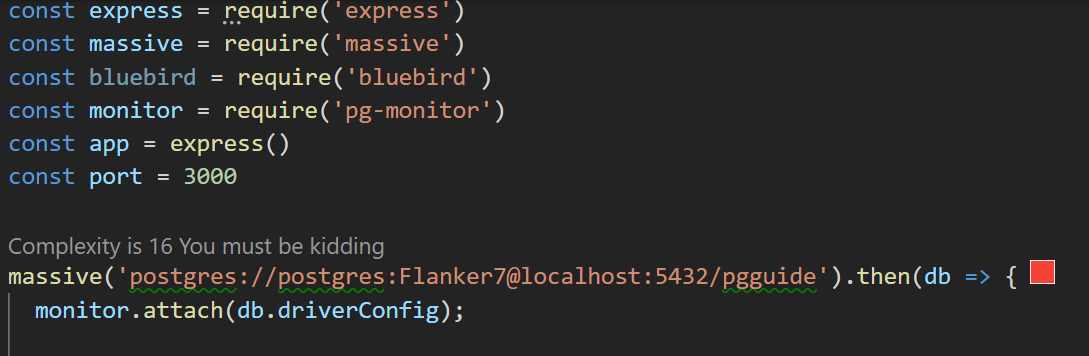
Lab 1

C15410232-wks-1

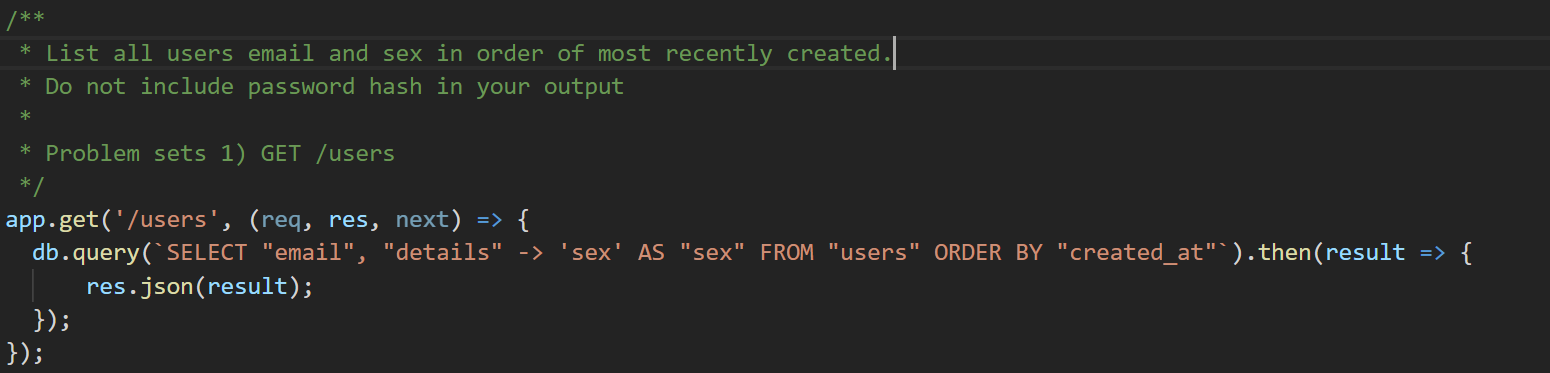
Problem Sets 1 – GET /users

List all users email and sex in order of most recently created. Do not include password hash in your output

Connection



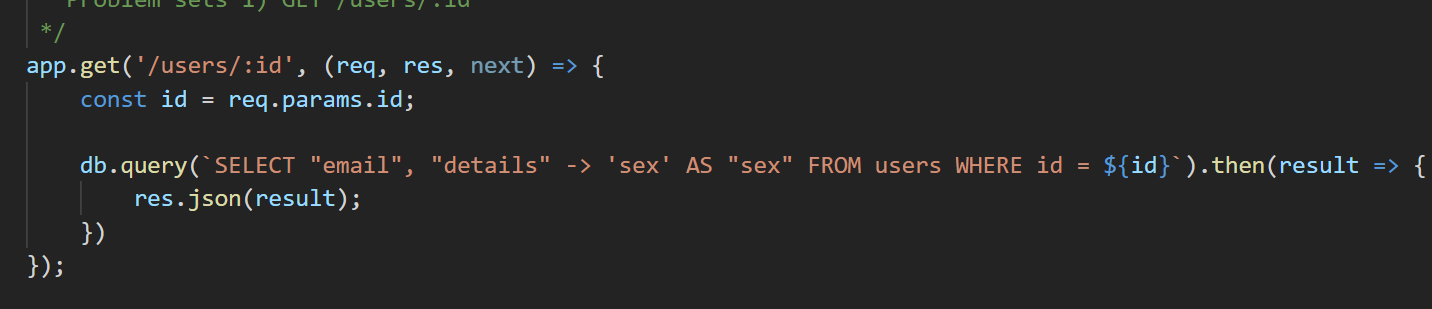
Answer to question





Problem Sets 1 – GET /users/:id

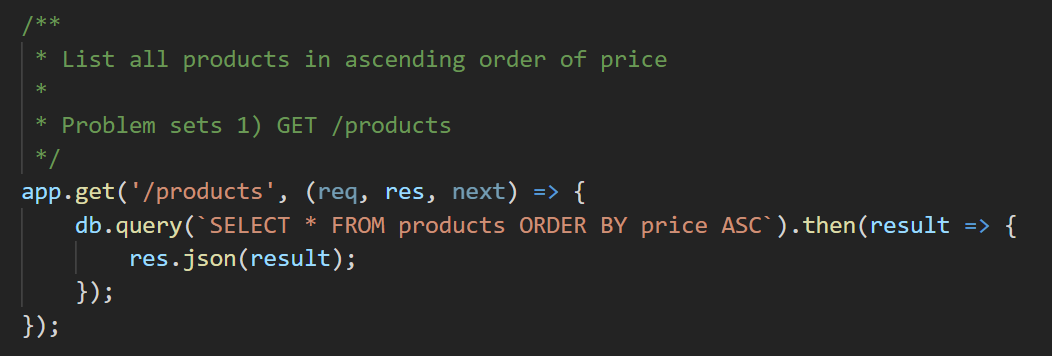
Show above details of the specified user





Problem Sets – GET /products

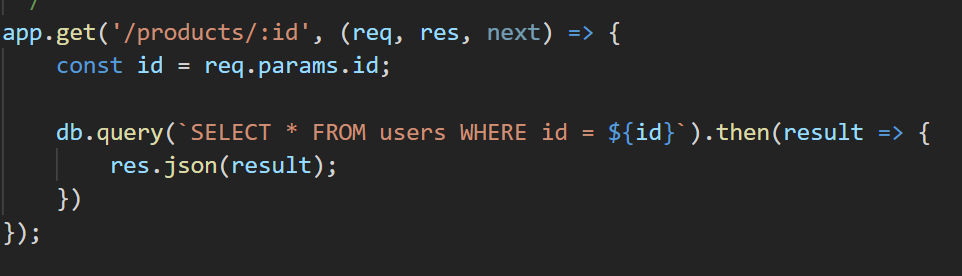
List all products in ascending order of price

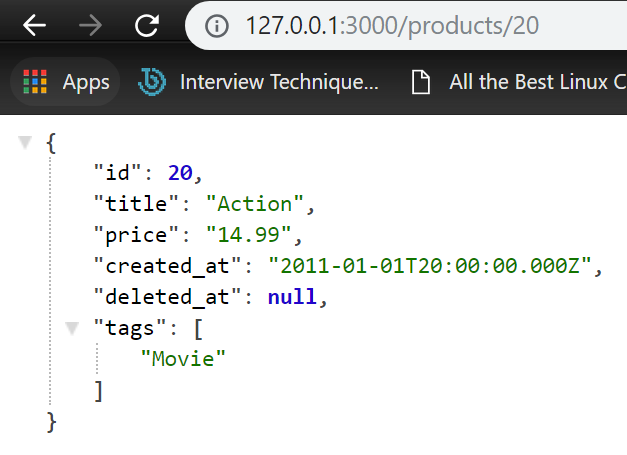




Problem Sets – GET /products/:id

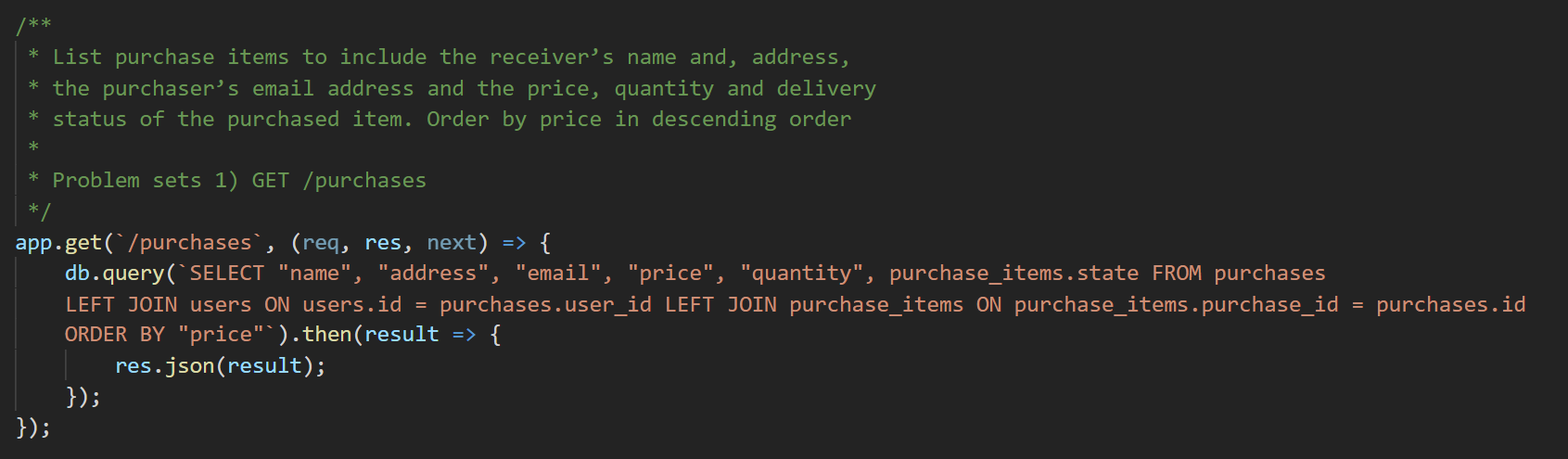
Show details of the specified products





Problem sets – GET /purchases

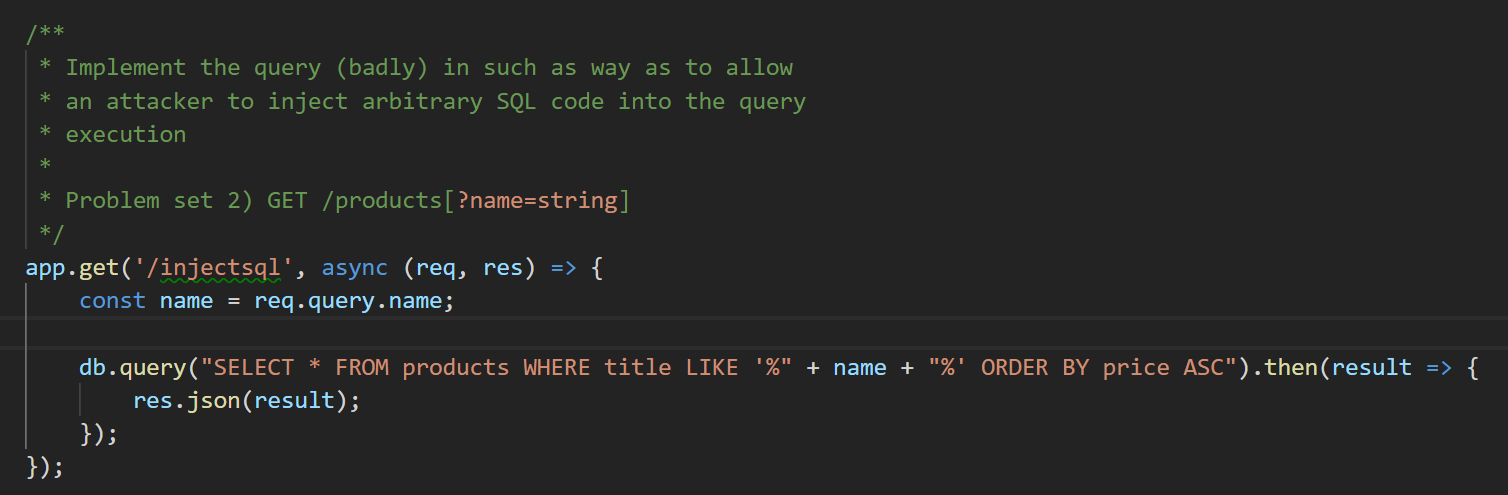
List purchase items to include the receiver’s name and, address, the purchaser’s email address and the price, quantity and delivery status of the purchased item. Order by price in descending order

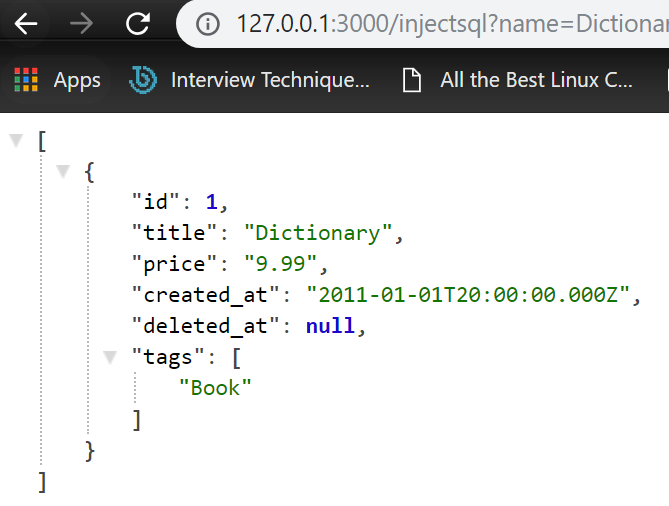


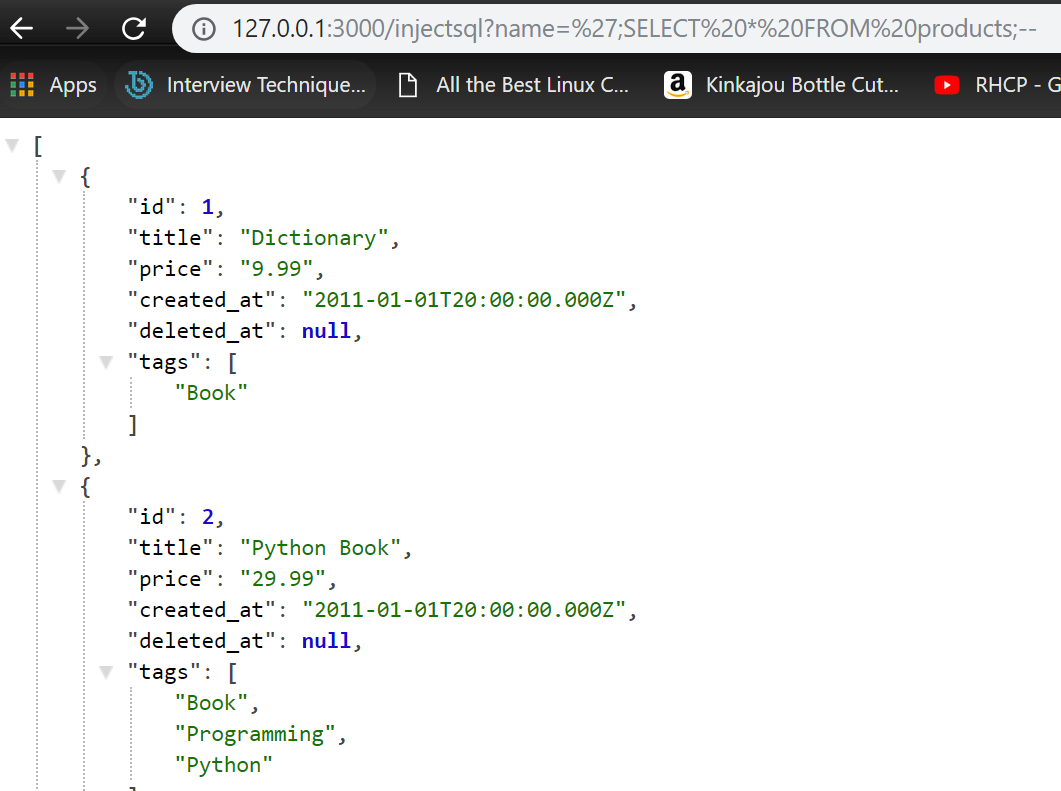


Problem Sets 2 - GET /products[?name=string]

Building on your solution to part 1 for the API to the products resource from the pgguide database, extend the product indexing endpoint to allow the filtering of products by name as follows



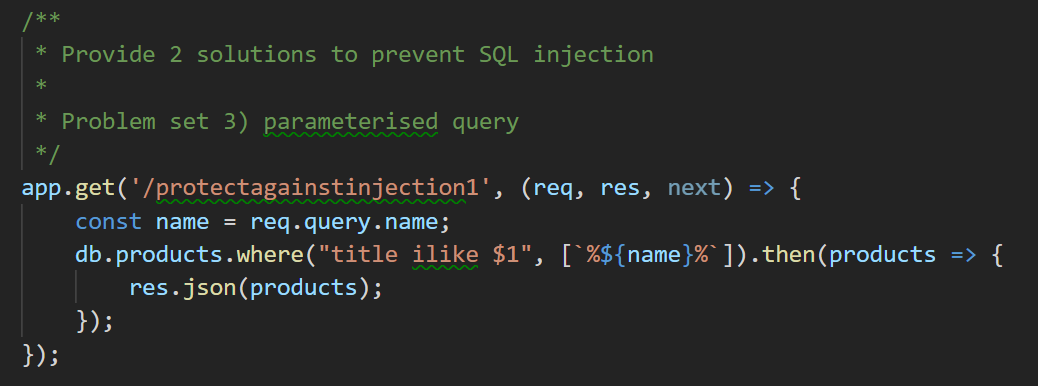


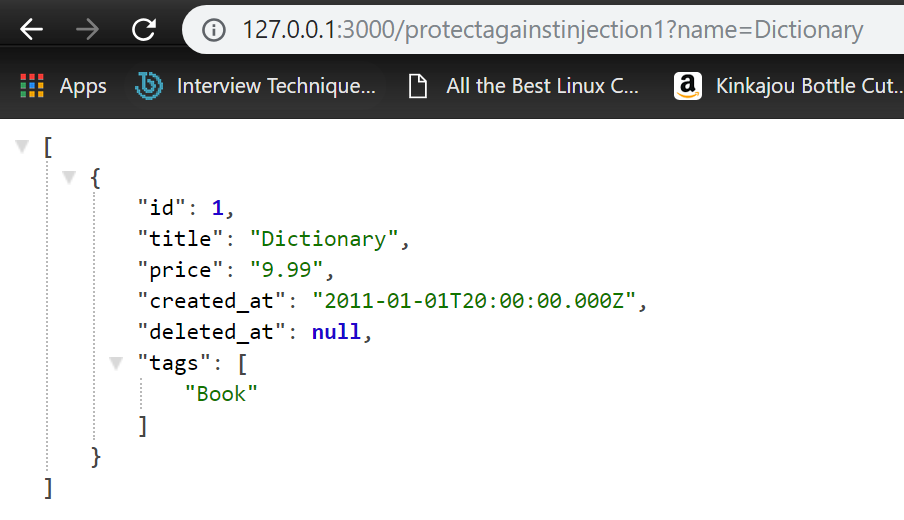


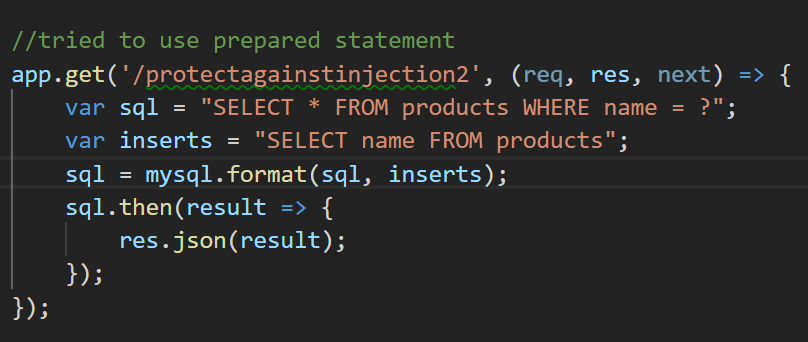
Problem Sets 3

Provide two solutions to eliminate the security hole in your approach from the previous section as follows:

* Using a parameterised query
* Using a stored procedure using SQL or PLPGSQL whichever you prefer







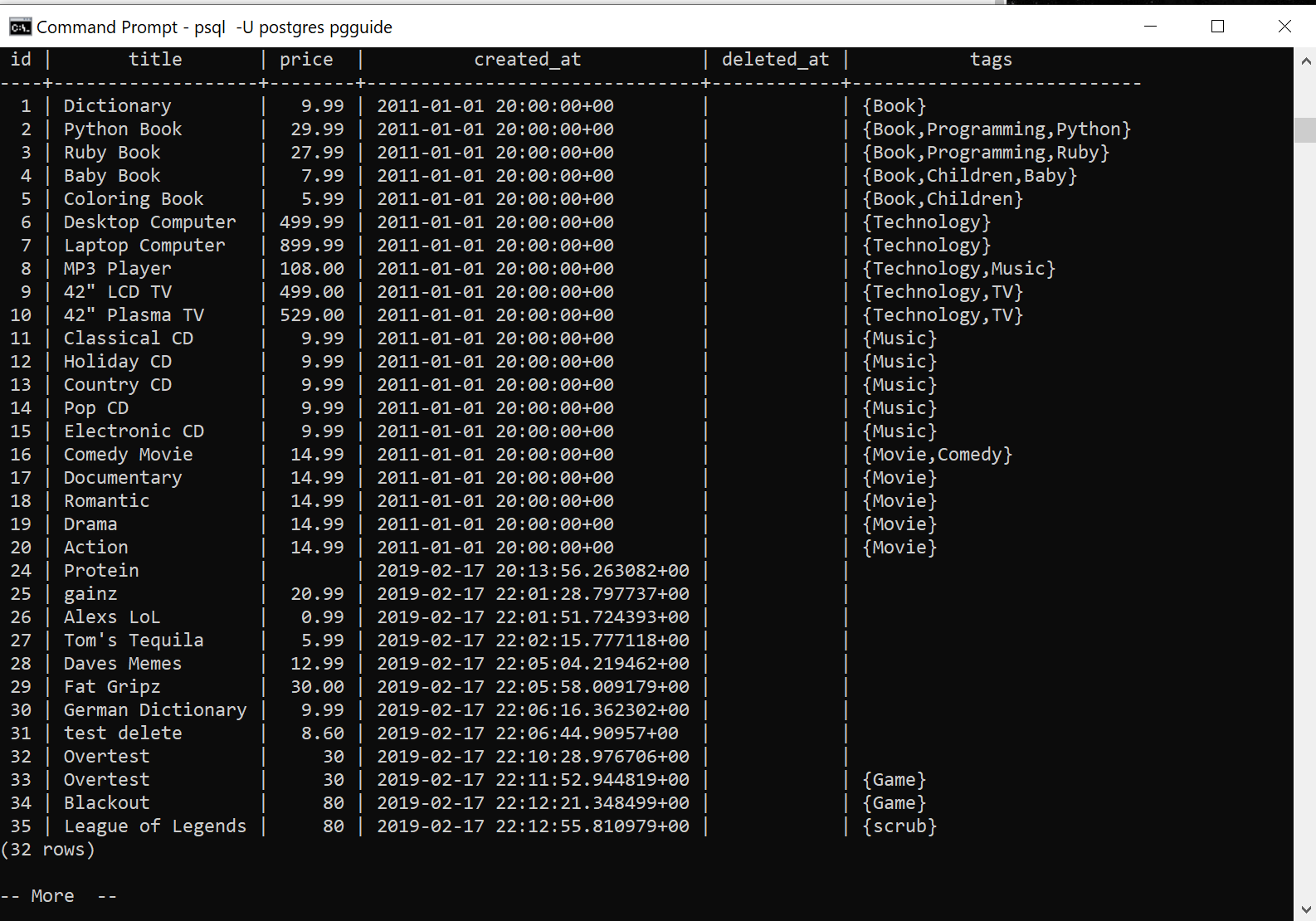
Problem Sets 4

Create a brand new Express project using the Sequelize ORM. Install and configure Sequelize and wire it up to the pgguide database.. Verify that you have basic connectivity before proceeding.



Problem sets 5

Use your models and Javascript code to populate the database with some additional test data for all of the models above



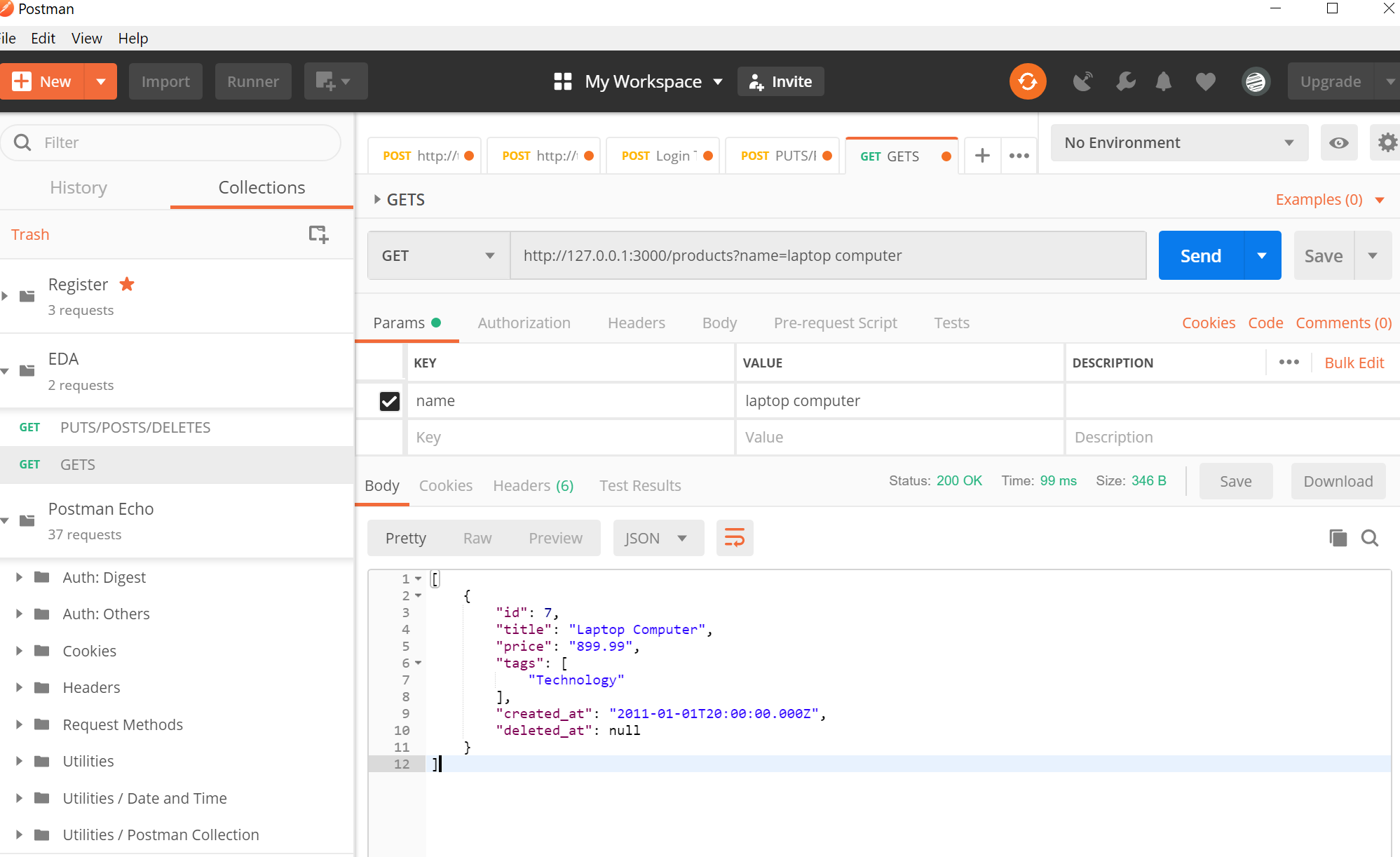
Problem Sets 6

Reimplement the RESTful API using Sequelize and Express for your system. Your API should support the following CRUD operations as follows, returning JSON responses

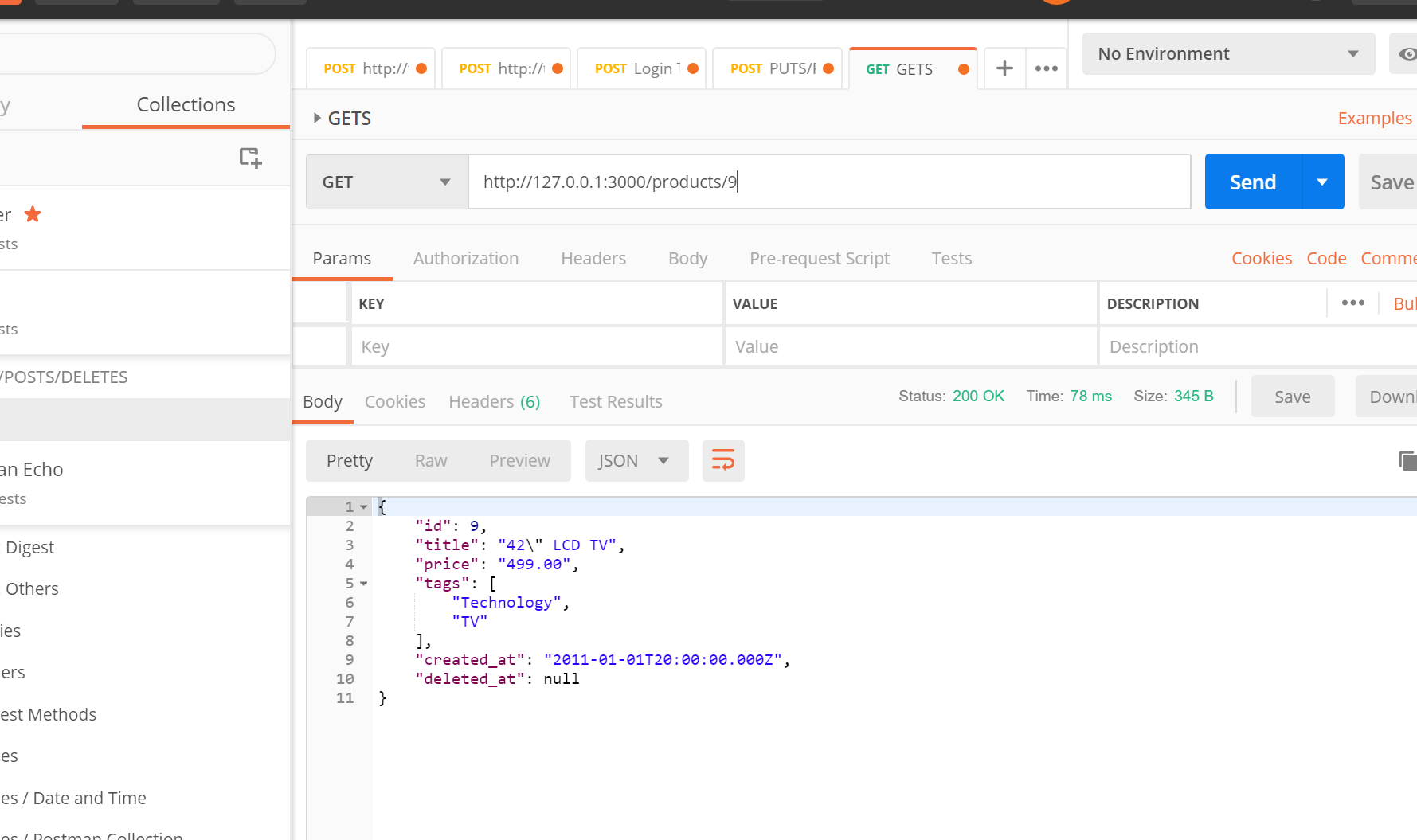
|  |  |
| --- | --- |
| GET /products[?name=string] | List all products |
| GET /products/:id | Show details of the specified products |
| POST /products | Create a new product instance |
| PUT /products/:id | Update an existing product |
| DELETE /products/:id | Remove an existing product |

Show test cases for each of the API endpoint REST operations

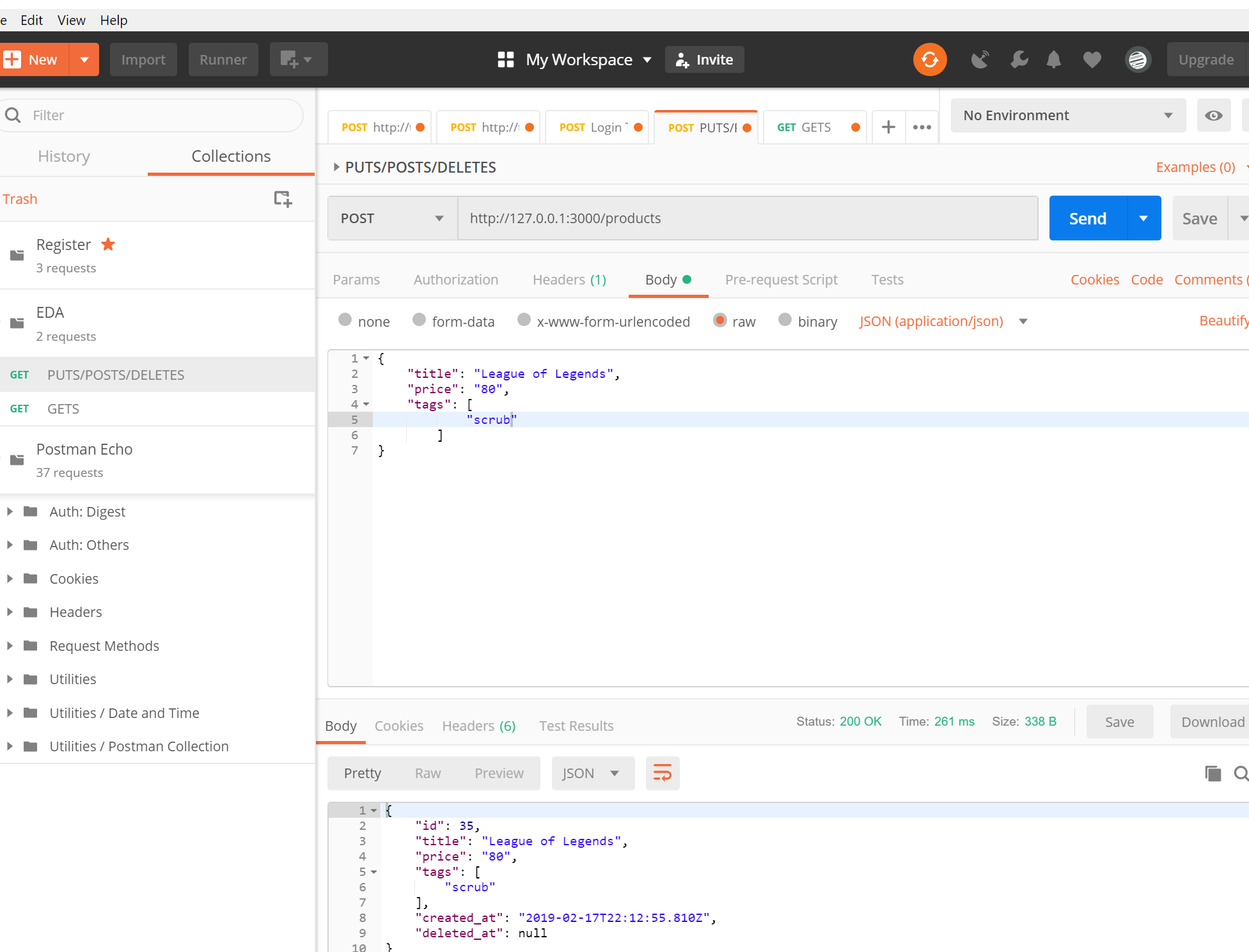
GET /products[?name=string]



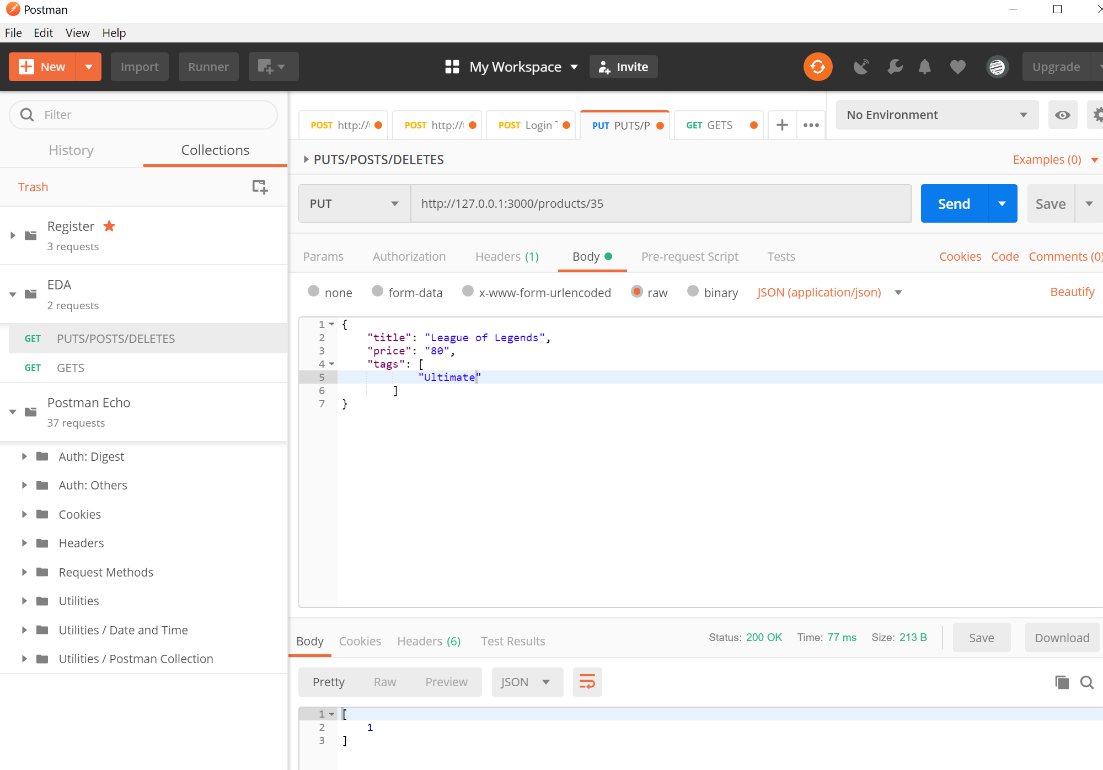
GET /products/:id



POST /products



PUT /products/:id



DELETE /products/:id

