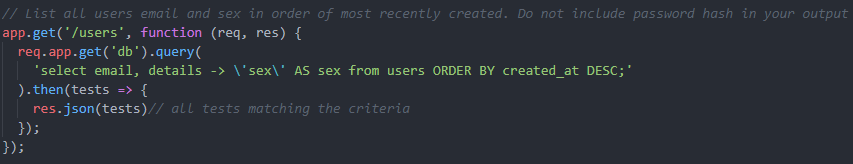
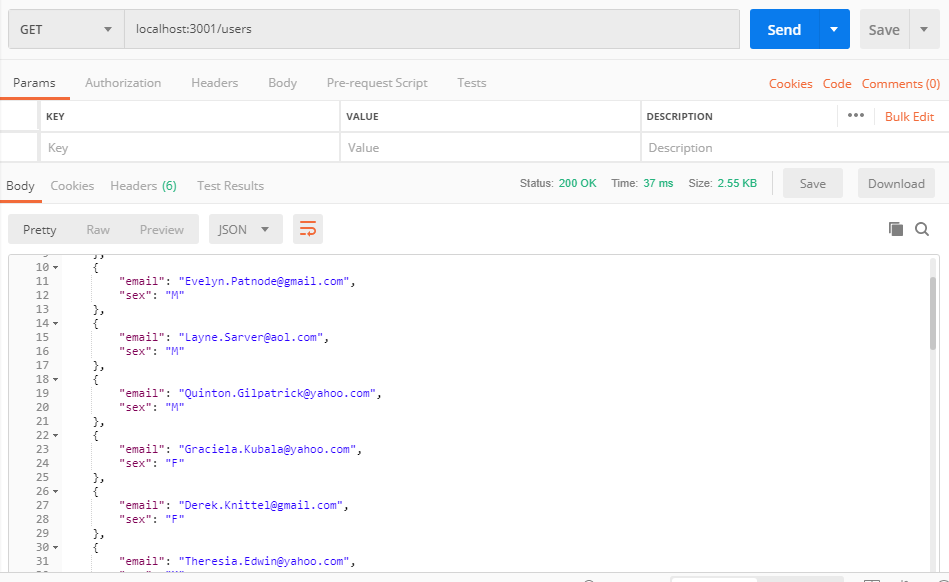
Enterprise Application Development  
Lab 1   
Mohamad Zabad  
C15745405

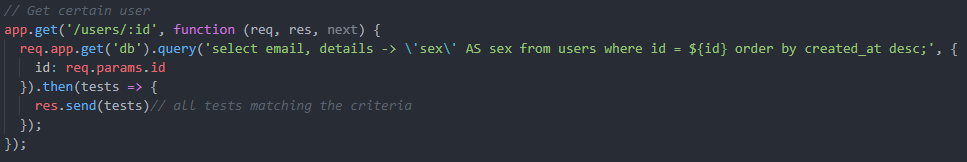
Using Node, Express and Massive create the following HTTP API endpoints serving the following resources as JSON documents

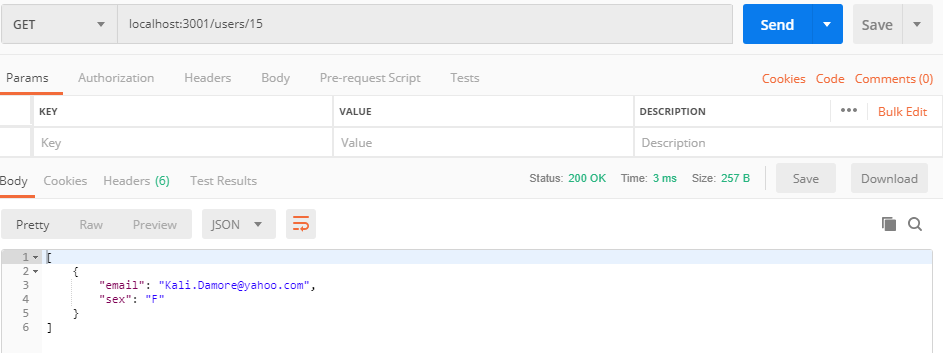
|  |  |
| --- | --- |
| GET /users | List all users email and sex in order of most recently created. Do not include password hash in your output |

1. 

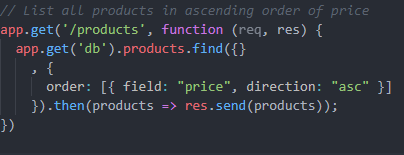


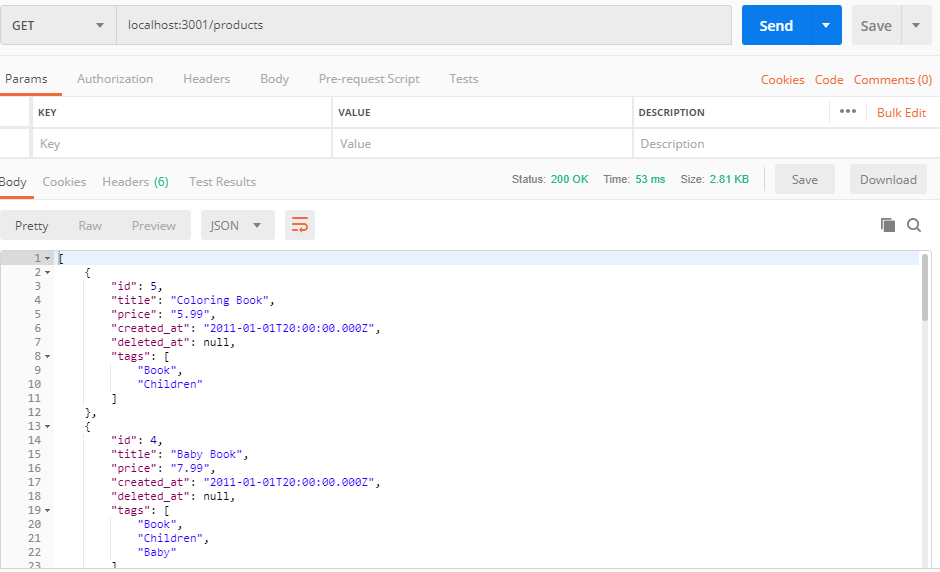
|  |  |
| --- | --- |
| GET /users/:id | Show above details of the specified user |



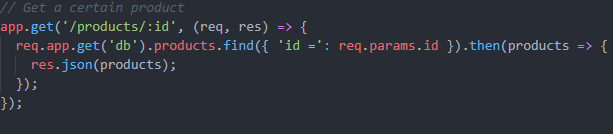


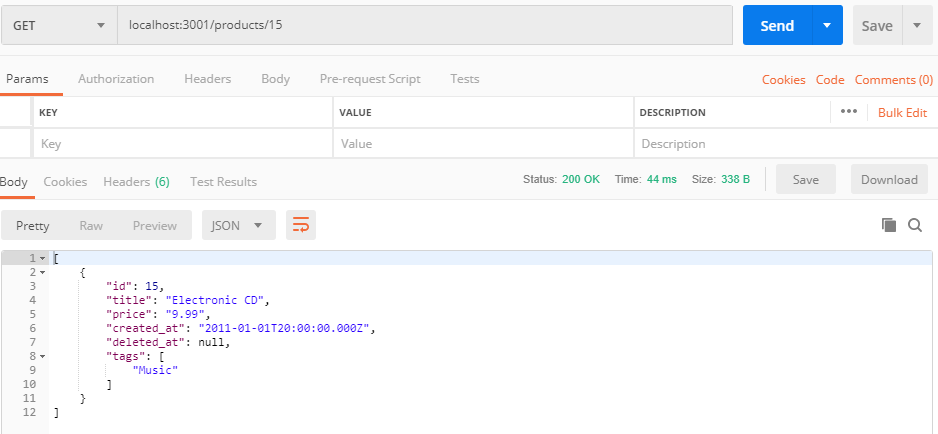
|  |  |
| --- | --- |
| GET /products | List all products in ascending order of price |



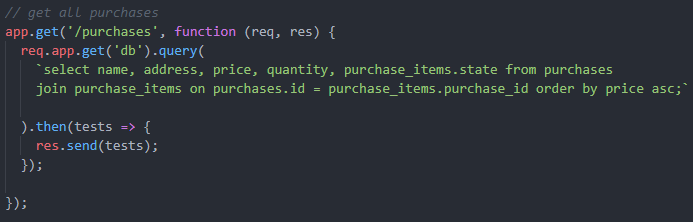


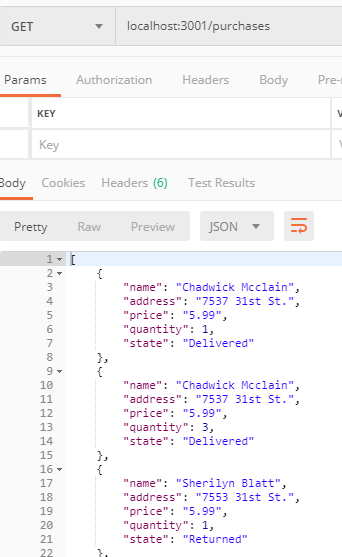
|  |  |
| --- | --- |
| GET /products/:id | Show details of the specified products |



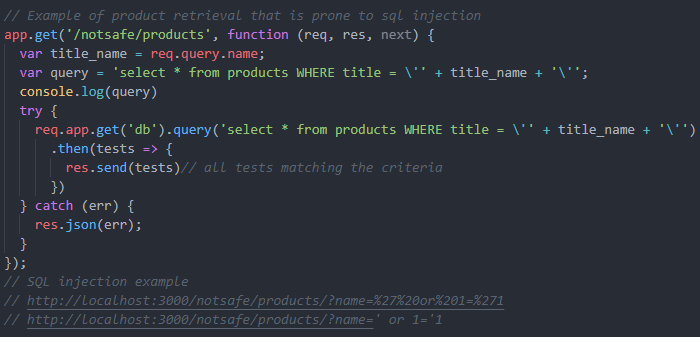


|  |  |
| --- | --- |
| GET /purchases | List purchase items to include the receiver’s name and, address, the purchaser’s email address and the price, quantity and delivery status of the purchased item. Order by price in descending order |

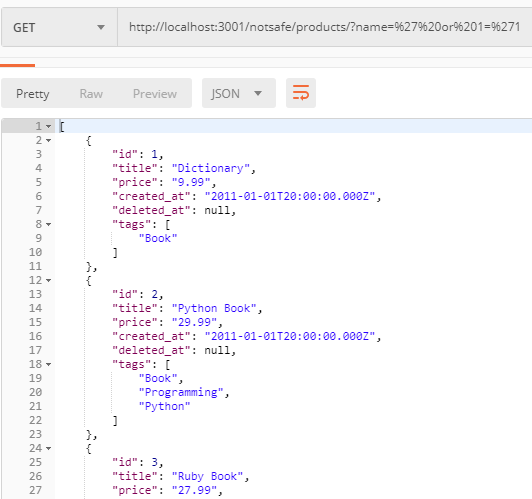




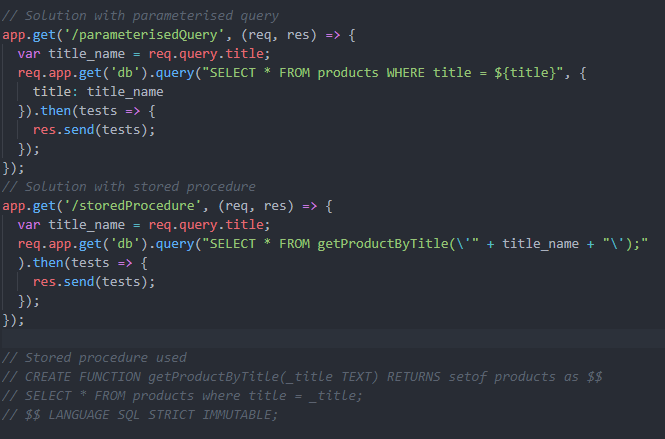
1.  For your solution you should implement the query (badly)

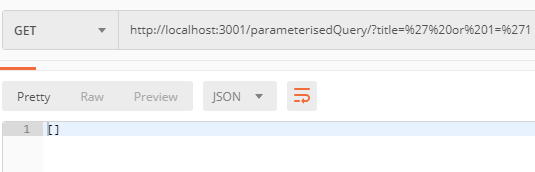


SQL Injection that displays all products instead of only one



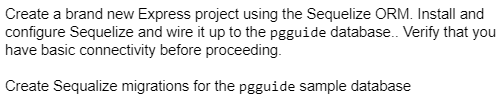
1. 

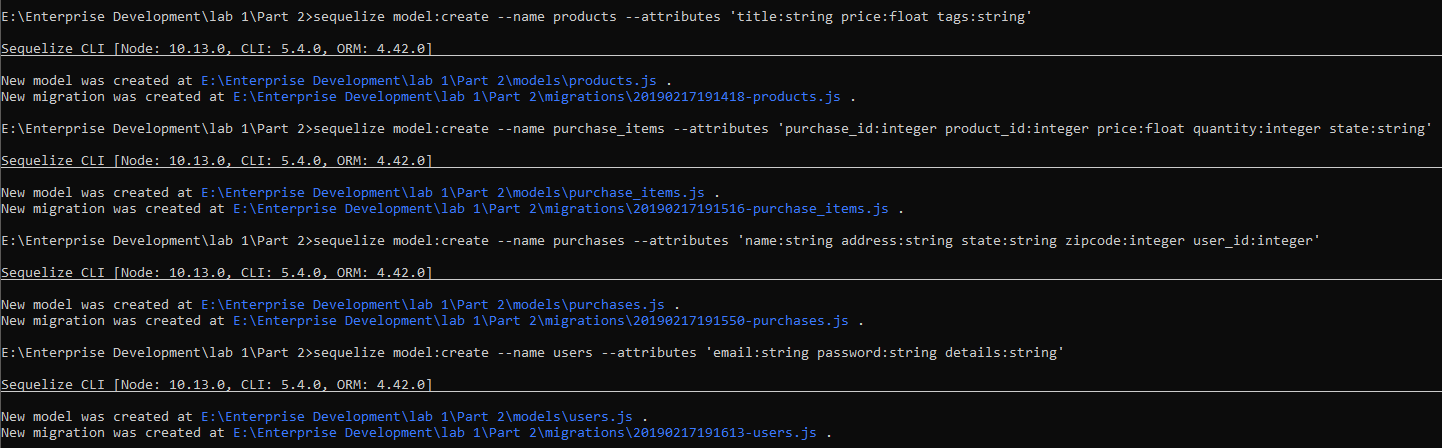




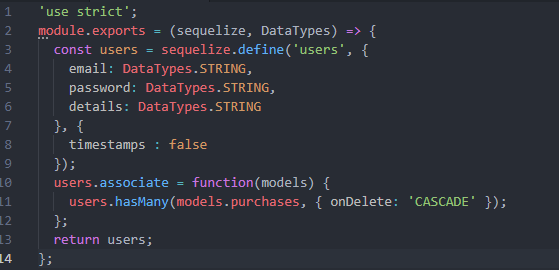
Stored Procedure response for the SQL Injection:



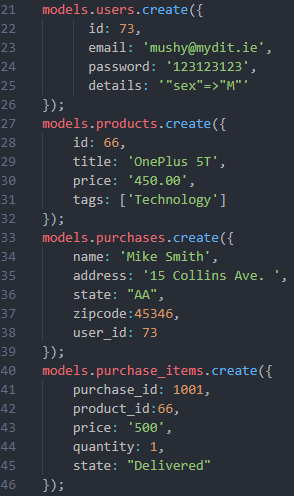
1. 

The following commands create a model that represents the existing table in the database we created:

Example of generated model:



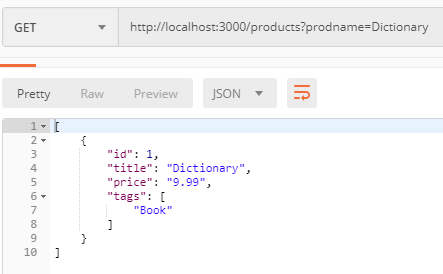
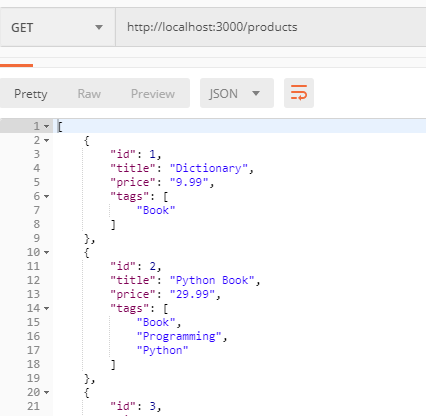
1. Populating the tables with new examples:



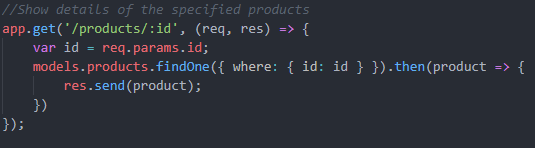


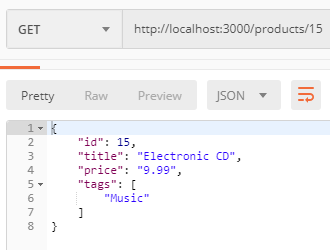
What I understood from this, is that I have to display all results for a matching name, or everything if no name is given



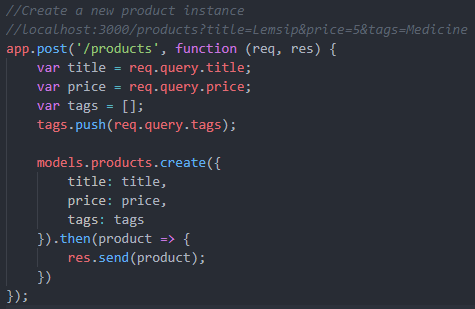
 

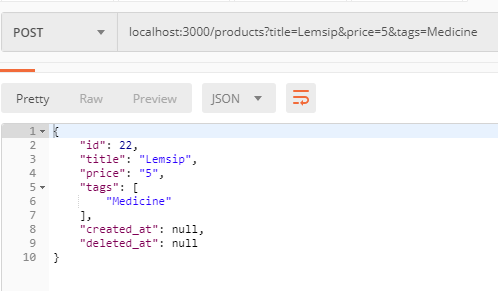






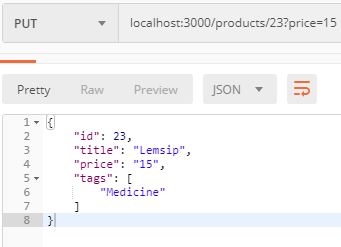
















On Success it displays which product was deleted

