Kevin Smith
SWDV 691 – Software Development Capstone
Week 3 – Re-Design: Service Layer(s)
8/10/2020

I haven't needed to add any new service layers yet. So far, I have one service layer that is active in my project. As I come across more in the next week, I will add them to this file. I believe next thing I will be using is the add to customers api. Below I will add numbers and descriptors for the planned order of events.

Some Examples for the rest api's are as follows:
CREATE:
POST http://localhost:5000/api/cust/add
READ:
GET http://localhost:5000/api/cust/all
UPDATE:
PUT http://localhost:5000/api/cart/update
DELETE: - (don't really plan to delete anything but as an example to delete a specific inactive customer)
DELETE http://api/cust/delete/<int:id>

Erroneous error example: - (When there's a typo in the api request – ie.. api/carittem/all)

```
<h1>404</h1>
<p>The resource could not be found.</p>
```

The following is a more in-depth example of each rest api planned for regular use:

Add customers by json(NEXT - #1): made username,password,email unique – gets error code 1062
Error Code: 1062. Duplicate entry 'batman' for key 'customers.username_3'

Body example:

```
[{
    "address": "wayne manor",
    "address2": null,
    "address3": null,
    "fName": "bruce",
    "lName": "wayne",
    "password": "robin",
    "phone": "1-800-bmobile",
    "phone2": null,
    "username": "batman"
    "email": "bwisbm@wayneenterprises.com"
}]
```

CLI - curl -X POST -H "Content-Type:application/json" -d "[{\"address\":\"wayne manor\",\"address2\":null,\"address3\":null,\"fName\":\"bruce\",\"lName\":\"wayne\",\"password\":\"robin\",\"phone\":\"1-800-bmobile\",\"phone2\":null,\"username\":\"batman\"}]" http://localhost:5000/api/cust/add

Postman – POST http://localhost:5000/api/cust/add
Response:

Status: 200 OK
Body example: "Customer added successfully!"

Add products by json(EMPLOYEE INTERFACE – FUTURE WORK):

Body example:
```
[{
    "description": "bottled water",
    "imageFile": "c:/imaeg10",
    "price": 1.00,
    "prodName": "Water"
}]
```

CLI - curl -X POST -H "Content-Type:application/json" -d "[{\"description\":\"bottled water\",\"imageFile\":\"c:/imaeg10\",\"price\":1.00,\"prodName\":\"Water\"}]" http://localhost:5000/api/products/add

Postman – POST http://localhost:5000/api/products/add

Response:

Status: 200 OK

Body example: "Product added successfully!"

Add cart by json(#2):

Body example:
```
[{
    "CustID": 9
},
{
    "CustID": 10
}]
```

CLI - curl -X POST -H "Content-Type:application/json" -d "[{\"CustID\":9},{\"CustID\":10}]" http://localhost:5000/api/carts/add

Postman – POST http://localhost:5000/api/carts/add

Response:

Status: 200 OK

Body example: "Cart added successfully!"

Add cart items by json(#3):

Body example:
```
[
  {
    "custid": 9,
    "prodID": 4,
    "qty": 3
  }]
```

CLI - curl -X POST -H "Content-Type:application/json" -d "[{\"custid\":9,\"prodID\":4,\"qty\":3}]" http://localhost:5000/api/cartitems/add

Postman – POST http://localhost:5000/api/cartitems/add

Response:

Status: 200 OK

Body example: "Cart added successfully!"

Request all customers(EMPLOYEE INTERFACE – FUTURE WORK):

CLI - curl http://localhost:5000/api/cust/all -H "Content-Type:application/json"

Postman – GET http://localhost:5000/api/cust/all

Response:

Status: 200 OK

Body: json file

Body example: [


```
{
    "CustID": 1,
    "address": "streets",
    "address2": null,
    "address3": null,
    "fName": "kevin",
    "lName": "smith",
    "password": "kpassword",
    "phone": "5555555555",
    "phone2": null,
    "username": "kusername"
  },
```


Request all customers by username and password(USING FOR LOGIN):

CLI - curl http://localhost:5000/api/cust/ripley/ripspasswd -H "Content-Type:application/json"

Postman – GET http://localhost:5000/api/cust/ripley/ripspasswd

Response:

Status: 200 OK

Body: json file

Body example: [


```
{
  "CustID": 8,
  "address": "U.S.S Sulaco",
  "address2": "",
  "address3": "",
  "email": null,
  "fName": "ellen",
  "lName": "ripley",
  "password": "ripspasswd",
  "phone": "phone#",
  "phone2": "",
  "username": "ripley"
}
```


Request all products: (ACTIVE)

CLI - curl http://localhost:5000/api/products/all -H "Content-Type:application/json"

Postman – GET http://localhost:5000/api/products/all

Response:

Status: 200 OK

Body: json file

Body example: [


```
{
    "description": "King Sized Candy",
```

```
    "imageFile": "c:/imaeg1",
    "price": 1.99,
    "prodID": 1,
    "prodName": "Peanut M&M's"
  },
```


Request all carts(EMPLOYEE INTERFACE – FUTURE WORK):

CLI - curl http://localhost:5000/api/carts/all -H "Content-Type:application/json"

Postman – GET http://localhost:5000/api/carts/all

Response:

Status: 200 OK

Body: json file

Body example:


```
[
  {
    "CartID": 1,
    "CustID": 1,
    "dateCreated": "Wed, 22 Jul 2020 21:20:27 GMT",
    "state": 0
  },
```


Request all cart items(#4 - ALSO EMPLOYEE INTERFACE – FUTURE WORK):

CLI - curl http://localhost:5000/api/cartitems/all -H "Content-Type:application/json"

Postman – GET http://localhost:5000/api/cartitems/all

Response:

Status: 200 OK

Body: json file

Body example:


```
[
  {
    "cartItemID": 2,
    "custid": 1,
    "prodID": 1,
    "qty": 3
  },
```


Request customer by id(#5 – ALSO EMPLOYEE INTERFACE – FUTURE WORK):

CLI - curl http://localhost:5000/api/cust/8 -H "Content-Type:application/json"

Postman – GET http://localhost:5000/ api/cust/8

Response:

Status: 200 OK

Body: json file

Body example:


```
{
  "CustID": 8,
  "address": "98 add st",
  "address2": "",
  "address3": "",
  "fName": "ellen",
  "lName": "ripley",
  "password": "ripspasswd",
  "phone": "phone#",
```

```
  "phone2": "",
  "username": "ripley"
}
```


Request product by id(EMPLOYEE INTERFACE – FUTURE WORK):

CLI - curl http://localhost:5000/api/products/6 -H "Content-Type:application/json"

Postman – GET http://localhost:5000/ api/products/6

Response:

Status: 200 OK

Body: json file

Body example:


```
{
  "description": "coffee mug",
  "imageFile": "c:/imaeg6",
  "price": 10.99,
  "prodID": 6,
  "prodName": "best dad mug"
}
```

Request cart by cust id(#6 – ALSO EMPLOYEE INTERFACE – FUTURE WORK):

CLI - curl http://localhost:5000/api/cart/view/1 -H "Content-Type:application/json"

Postman – GET http://localhost:5000/ api/cart/view/1

Response:

Status: 200 OK

Body: json file

Body example:


```
{
  "CartID": 1,
  "CustID": 1,
  "Total": 58.90,
  "dateCreated": "Wed, 22 Jul 2020 21:20:27 GMT",
  "fname": "kevin",
  "lname": "smith",
  "state": 0
}
```


Request cart items by cust id(#7 – ALSO EMPLOYEE INTERFACE – FUTURE WORK):

CLI - curl http://localhost:5000/api/cartitems/1 -H "Content-Type:application/json"

Postman – GET http://localhost:5000/ api/cartitems/1

Response:

Status: 200 OK

Body: json file

Body example:


```
[
  {
    "cartItemID": 2,
    "custid": 1,
    "prodID": 1,
    "qty": 3
  }]
```

Update customer by cust id with json file(EMPLOYEE INTERFACE – FUTURE WORK):

Body example:

```
[{
    "CustID": 8,
    "address": "U.S.S Sulaco",
    "address2": "",
    "address3": "",
    "fName": "ellen",
    "lName": "ripley",
    "password": "ripspasswd",
    "phone": "phone#",
    "phone2": "",
    "username": "ripley"
}]
```

CLI - curl -H "Content-Type:application/json" -X PUT -d "[{\"CustID\":8,\"address\":\"U.S.S Sulaco\",\"address2\":\"\",\"address3\":\"\",\"fName\":\"ellen\",\"lName\":\"ripley\",\"password\":\"ripspasswd\",\"phone\":\"phone#\",\"phone2\":\"\",\"username\":\"ripley\"}]" http://localhost:5000/api/cust/update

Postman – PUT http://localhost:5000/ api/cart/view/1

Response:

Status: 200 OK

Body example: `"Customer Table updated successfully!"`

Update cart state by cust with json file(#8 – TURN CART INTO ORDER BY BOOLEAN VALUE):

Body example:

```
[{
    "CartID": 1,
    "state": "1",
}]
```

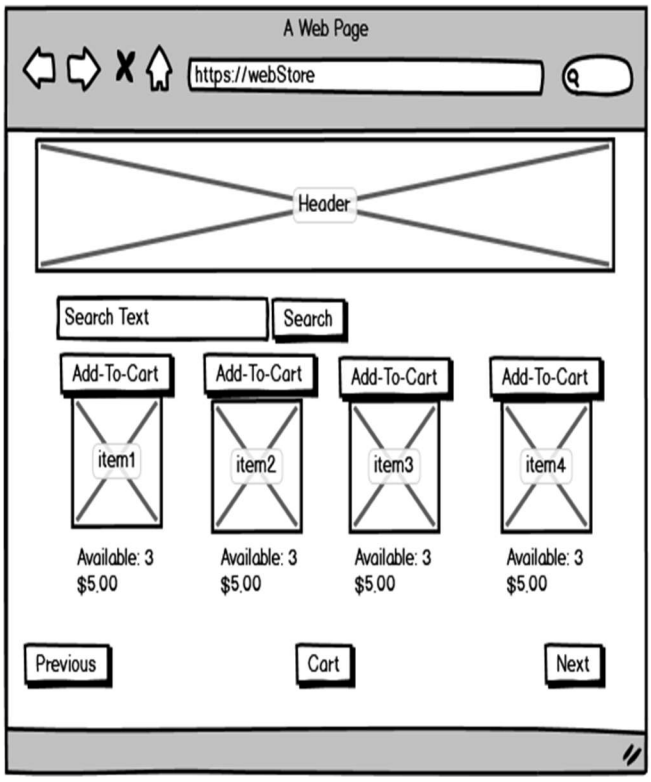
CLI - curl -H "Content-Type:application/json" -X PUT -d "[{\"CartID\":1,\"state\":1}]" http://localhost:5000/api/cart/update

Postman – PUT http://localhost:5000/ api/cart/update

Response:

Status: 200 OK

Body example: `"Cart updated successfully!"`

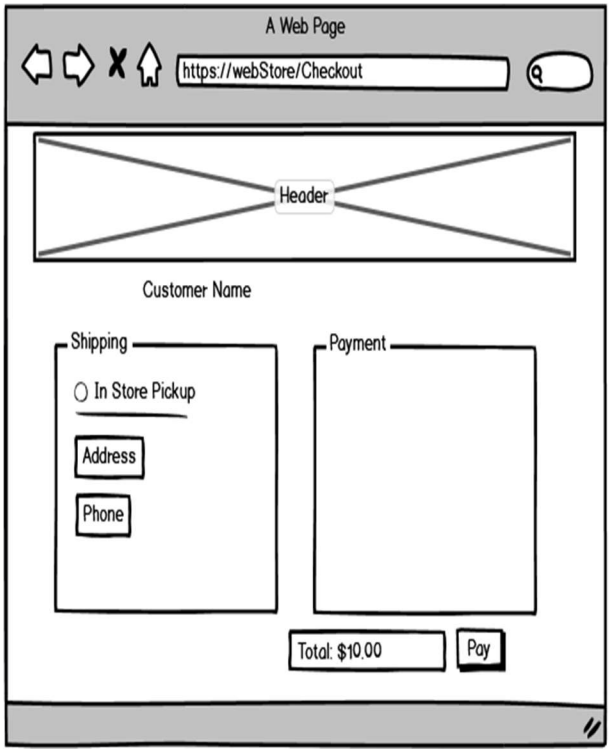


Add to temp cart items api - POST http://..../api/tempcart/add

When user clicks add-to-cart this api will take the json data associated with the button and will add it to the temporary cart items list. After either saving cart or during checkout (user will need to create account) this data will be pushed to the permanent tables - add new customer, add cart, add products to cart items.

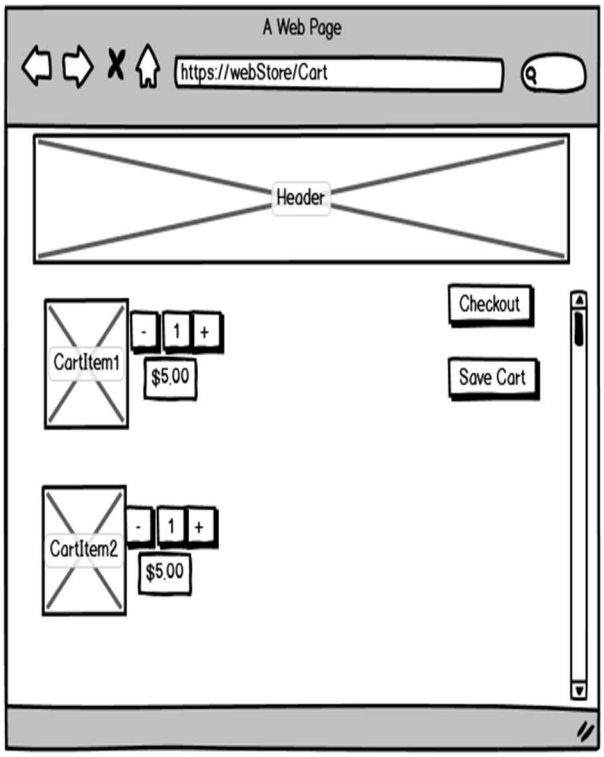
Get cart items api - POST http://..../api/tempcart

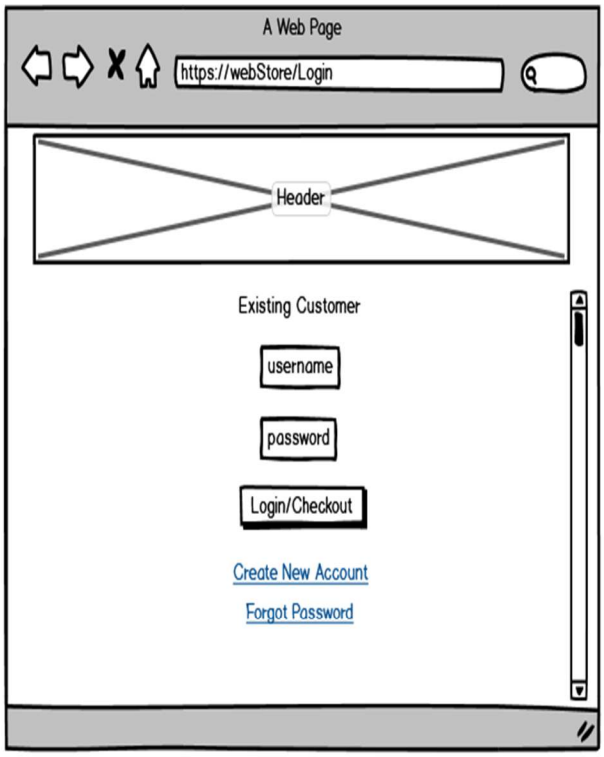
By the time the user clicks on cart, there will be either an empty temp cart or a temp cart with items in it. The cart button will pull the jason data associated with the temp cart and the temp cart items to show on the cart page.

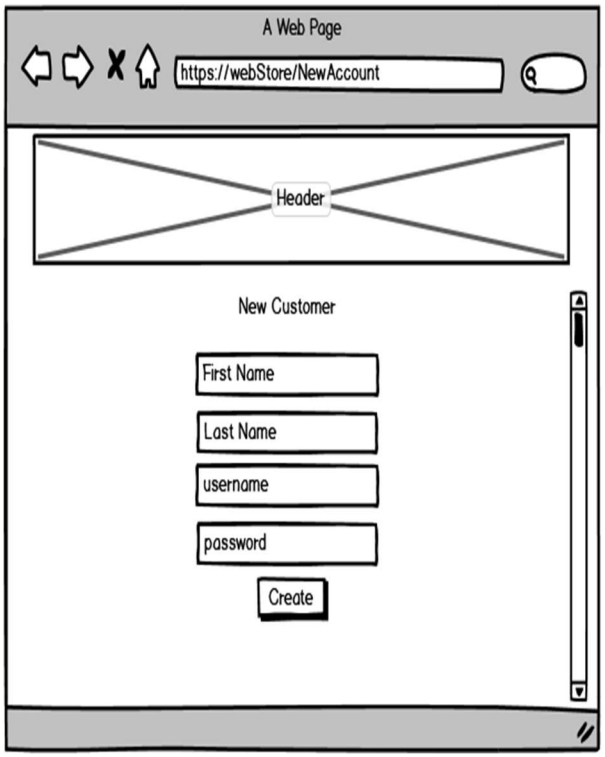


Update customer api - PUT http://..../api/cust - with json file

By the time the customer has clicked pay, the customer information will be complete. This will update the customer by their id which they should have by this point.







Add to customer api - POST http://..../api/cust/add

After clicking create, a new customer will be added the customers table.

Get customer api - GET http://..../api/cust/<int:id>

After clicking login/checkout, the api will pull the customer info in hopes of autopopulating the customer info.