

Welcome to your first group project, you will be broken up into groups of either 5 or 6 people. Together over the next few days you will build a RESTful service for a banking application.

Task 1: Create the restful endpoint for an account object.

Account
<ul style="list-style-type: none"> - id: Long - type: Enum - nickname: String - rewards: Integer - balance: Double - customer: Customer

- **id** - is an auto generated value which should be assigned by the database
- **type** - is an enumerated value which when serialized should print out as a string
 - Savings
 - Checking
 - Credit
- **nickname** - is a string value assigned at creation to describe the account i.e “Leon’s savings account”
- **rewards** - is a integer value which represents the reward points assigned to an account
- **balance** - is the amount of money available in the account which can be applied to purchases.
- **customer**- customer object associated with account

Method: GET	URI: /accounts	Action: Get all accounts
Method: GET	URI: /accounts/{customerId}	Action: Get account by id
Method: GET	URI: /customers/{customerId}/accounts	Action: Get account by id
Method: POST	/customers/{customerId}/accounts	Create an account
Method: PUT	/accounts/{accountId}	Update a specific existing account
Method:DELETE	/accounts/{accountId}	Update a specific existing account

Account: List all of the accounts available		
Method: GET	URI: /accounts	Action: Get all accounts
Example Failed Response: Status (404) <pre> { "code" : 404, "message": "error fetching accounts" } </pre>		
Example Successful Response: Status (200) <pre> { "code" : 200, "message": "Success", "data": [{ "id": 1, "type": "Savings", "nickname": "Leon's Account", "rewards": 37605, "balance": 19351, "customer_id": 10 }, { "id": 2, "type": "Savings", "nickname": "Zach Account", "rewards": 37605, "balance": 19351, "customer_id": 11 }] } </pre>		

Account: Get one account by customer id		
Method: GET	URI: /accounts/{customerId}	Action: Get account by id
Example Failed Response: Status (404) <pre>{ "code" : 404, "message": "error fetching account" }</pre>		
Example Successful Response: Status (200) <pre>{ "code" : 200, "message": "error fetching account", "data": { "id": 1, "type": "Savings", "nickname": "Leon's Account", "rewards": 37605, "balance": 19351, "customer_id": 10 } }</pre>		

Account : Update a account by account id		
PUT	/accounts/{accountId}	Update a specific existing account
Example Failed Response: Status (404) <pre>{ "code" : 404, "message": "Error ", }</pre>		
Example Successful Response: Status (200) <pre>{ "code": 200, "message": "Customer account updated" }</pre>		

Account: Get all accounts belonging to one customer by customer ID

Method: GET

URI: /customers/{customerId}/accounts

Action: Get account by id

Example Failed Response: Status (404)

```
{
  "code" : 404,
  "message": "Error fetching customers accounts"
}
```

Example Successful Response: Status (200)

```
{
  "code" : 200,
  "message": "Error fetching customers accounts",
  "Data":
    [
      {
        "id": 1,
        "type": "Savings",
        "nickname": "Leon's Saving Account",
        "rewards": 37605,
        "balance": 19351,
        "customer_id": 10
      },
      {
        "id": 32,
        "type": "Checking",
        "nickname": "Leon's Checking Account",
        "rewards": 37605,
        "balance": 22,
        "customer_id": 10
      }
    ]
}
```

Account: Create a new account and assign it to a customer by the customer id		
POST	/customers/{customerId}/accounts	Create an account
Example Failed Response: Status (404) <pre>{ "code" : 404, "message": "error fetching creating customers account" }</pre>		
Example Successful Response: Status (201) <pre>{ "code": 201, "message": "Account created", "data": { "type": "Credit Card", "nickname": "string", "rewards": 0, "balance": 0, "customer_id": 10, "_id": 42 } }</pre>		

Account: Delete a account by account id		
DELETE	/accounts/{accountId}	Update a specific existing account
Example Failed Response: Status (404) <pre>{ "code" : 404, "message": "Account does not exist" }</pre>		
Example Successful Response: Status (201) <pre>{ "code": 202, "message": "Account successfully deleted" }</pre>		

Task 2: Create the restful endpoint for a Customer object.

Customer
+ id:Long + first_name:String + last_name:String + address: Set<Address>

- **id** - is an auto generated value which should be assigned by the database
- **first_name** - first name of customer
- **last_name** - last name of customer
- **address** - A set of addresses associated with customer

Address
+ id:Long + street_number:String + street_name:String + city:String + state:String + zip: String

- **id** - is an auto generated value which should be assigned by the database
- **street_name** - name of street
- **street_number** - street number
- **city** - city of customer
- **state** - state customer lives in
- **zip** - zip code of customer

Method: GET	/accounts/{customerId}/customer	Get customer that owns the specified account
Method: GET	/customers	Get all customers
Method: GET	/customers/{id}	Get customer id
Method: POST	/customers	Create a customer
Method: PUT	/customers/{id}	Update a specific existing bill

Customer: List all of the accounts for a specified customer

Method: GET

/accounts/{customerId}/customer

Get customer that owns the specified account

Example Failed Response: Status (404)

```
{
  "code" : 0,
  "message": "error fetching customers accounts"
}
```

Example Successful Response: Status (200)

```
{
  "code": 200,
  "message": "Success",
  "Data":[
    {
      "id": 1,
      "type": "Savings",
      "nickname": "Leon's Savings Account",
      "rewards": 37605,
      "balance": 19351,
      "customer_id": 10
    },
    {
      "id": 2,
      "type": "Checking",
      "nickname": "Leon's Checking Account",
      "rewards": 37605,
      "balance": 19351,
      "customer_id": 10
    }
  ]
}
```

Customer: List all customers

Method: GET

/customers

Get all customers

Example Failed Response: Status (404)

```
{
  "code" : 404,
  "message": "error fetching accounts"
}
```

```
{
  "code": 200,
  "message": "Success",
  "Data":
  [
    {
      "id": 1,
      "first_name": "Leon",
      "last_name": "Hunter",
      "address": {
        "street_number": "902",
        "street_name": "Walker Road",
        "city": "Clearfield",
        "state": "Pennsylvania",
        "zip": "16830"
      }
    },
    {
      "id": 2,
      "first_name": "David",
      "last_name": "Ginzburgh",
      "address": {
        "street_number": "40",
        "street_name": "Pitt Bridge",
        "city": "Branchland",
        "state": "West Virginia",
        "zip": "25506"
      }
    }
  ]
}
```


Customer: List all customers

Method: GET

/customers/{id}

Get customer id

Example Failed Response: Status (404)

```
{
  "code" : 404,
  "message": "error fetching account"
}
```

```
{
  "code": 200,
  "message": "Success",
  "data":
  {
    "id": 1,
    "first_name": "Leon",
    "last_name": "Hunter",
    "address": {
      "street_number": "902",
      "street_name": "Walker Road",
      "city": "Clearfield",
      "state": "Pennsylvania",
      "zip": "16830"
    }
  }
}
```

Customer : Create a new customer account

Method: POST

/customers

Create a customer

Example Failed Response: Status (404)

```
{
  "code" : 404,
  "message": "Error "
}
```

Example Successful Response: Status (200)

```
{
  "code": 200,
  "message": "Customer account updated",
  "Data":
  {
    "id": 3,
    "firstName": "Tariq",
    "lastName": "Hook",
    "address":
    [
      {
        "id": 5,
        "streetNumber": "123",
        "streetName": "Sesame Street",
        "city": "Wilmington",
        "state": "DE",
        "zipcode": "19801"
      }
    ]
  }
}
```

Customer: Get all accounts belonging to one customer by customer ID

Method: GET

URI: /customers/{customerId}/accounts

Action: Get account by id

Example Failed Response: Status (404)

```
{
  "code" : 404,
  "message": "Error fetching customers accounts"
}
```

Example Successful Response: Status (200)

```
{
  "code": 200,
  "message": "Customer account updated",
  "Data":
    [
      {
        "id": 1,
        "type": "Savings",
        "nickname": "Leon's Saving Account",
        "rewards": 37605,
        "balance": 19351,
        "customer_id": 10
      },
      {
        "id": 32,
        "type": "Checking",
        "nickname": "Leon's Checking Account",
        "rewards": 37605,
        "balance": 22,
        "customer_id": 10
      }
    ]
}
```

Customer: Create a new account and assign it to a customer by the customer id

POST

/customers/{customerId}/accounts

Create an account

Example Failed Response: Status (404)

```
{
  "code" : 404,
  "message": "error fetching creating customers account"
}
```

Example Successful Response: Status (201)

```
{
  "code": 201,
  "message": "Account created",
  "data": {
    "type": "Credit Card",
    "nickname": "string",
    "rewards": 0,
    "balance": 0,
    "customer_id": 10,
    "_id": 42
  }
}
```