

# CIS 550 Final Project Milestone 4

---

## ROUTE 1

Request Path: `/hello`

Description : Returns a user name to fill in the greeting on our homepage. Greet the user and welcome the user to our Oeda Platform

Request Parameter(s) : None

Request Description: N/A

Query Parameter(s) : None

Route Handler : `hello(req, res)`

Response Type : string

Response Parameters : None

Response description :

- default: general welcome
- with name param: general welcome concatenated with user name

---

## Sample

Path: <http://localhost:8080/hello>

Response: Hi! Welcome to the Oeda Platform!

Path: <http://localhost:8080/hello?name=Clara>

Response: Hi, Clara! Welcome to the Oeda Platform!

## Dashboard Section

---

## ROUTE 2

Request Path: `/YearlyOrder`

Description : Returns an array of total numbers of orders in 2016, 2017, and 2018, and the differences/increments between two years (2016 v.s. 2017, 2017 v.s. 2018). Display the result in our homepage dashboards

Request Parameter(s) : `year(int)`

Request Description: select from year 2016, 2017, 2018

Query Parameter(s) : None

Route Handler : `yearly_order(req, res)`

Response Type : JSON

Response Parameters : { results (JSON array of {order\_2018 (int), difference\_2017\_2018(string)}) }

Response description :

- default: total numbers of order in 2018, and the growth percentage compared to previous year
- with year param: total numbers of order in the chosen year, and the growth percentage compared to previous year

---

## Sample

Path: <http://localhost:8080/YearlyOrder?year=2018> (default)

<http://localhost:8080/YearlyOrder> (with param)

Response:

```
{
  "results": [
    {
      "order_2018": 52778,
      "difference_2017_2018": "21.58%"
    }
  ]
}
```

```
]
}
```

---

### ROUTE 3

Request Path: [/YearlySales](#)

Description : Returns an array of total sales in 2016, 2017, and 2018, and the differences/increments between two years (2016 v.s. 2017, 2017 v.s. 2018). Display the result in our homepage dashboards

Request Parameter(s) : year(int)

Request Description: select from year 2016, 2017, 2018

Query Parameter(s) : None

Route Handler : yearly\_sales(req, res)

Response Type : JSON

Response Parameters : { results (JSON array of {sales\_2018 (int), difference\_2017\_2018(string)}) }

Response description :

- default: total amount of sales in 2018, and the growth percentage compared to previous year
- with year param: total amount of sales in the chosen year, and the growth percentage compared to previous year

---

#### Sample

Path: <http://localhost:8080/YearlySales> (default)

<http://localhost:8080/YearlyOrder?year=2018> (with param)

Response:

```
{
  "results": [
    {
      "sales_2018": 3785320.43,
      "difference_2017_2018": "22.72%"
    }
  ]
}
```

---

### ROUTE 4

Request Path: [/YearlyReview](#)

Description : Returns an array of average review in 2016, 2017, and 2018, and the differences/increments between two years (2016 v.s. 2017, 2017 v.s. 2018). Display the result in our homepage dashboards

Request Parameter(s) : year(int)

Request Description: select from year 2016, 2017, 2018

Query Parameter(s) : None

Route Handler : yearly\_review(req, res)

Response Type : JSON

Response Parameters : { results (JSON array of {review\_2018 (int), difference\_2017\_2018(string)}) }

Response description :

- default: average review score in 2018, and the growth percentage compared to previous year
- with year param: average review score in the chosen year, and the growth percentage compared to previous year

---

#### Sample

Path: <http://localhost:8080/YearlyReview?year=2017> (default)

<http://localhost:8080/YearlyReview> (with param)

Response:

```
{
  "results": [
    {
      "review_2017": 3.89,
      "difference_2016_2017": null
    }
  ]
}
```

---

#### ROUTE 5

Request Path: [/YearlyState](#)

Description : Returns an array of the number of states participating in the ecommerce in 2016, 2017, and 2018, and the differences/increments between two years (2016 v.s. 2017, 2017 v.s. 2018). Display the result in our homepage dashboards

Request Parameter(s) : year(int)

Request Description: select from year 2016, 2017, 2018

Query Parameter(s) : None

Route Handler : yearly\_state(req, res)

Response Type : JSON

Response Parameters : { results (JSON array of {state\_2018 (int), difference\_2017\_2018(string)}) }

Response description :

- default: total number of states participated in 2018, and the growth percentage compared to previous year
- with year param: total number of states in the chosen year, and the growth percentage compared to previous year

---

#### Sample

Path: <http://localhost:8080/YearlyState> (default)

<http://localhost:8080/YearlyState?year=2017> (with param)

Response:

```
{
  "results": [
    {
      "state_2017": 27,
      "difference_2016_2017": "28.57%"
    }
  ]
}
```

## Analytics Section

---

#### ROUTE 6

Request Path: [/search](#)

Description : Returns details for the transactions that can be filtered by users based on price range, category, and time range

Request Parameter(s) : category(string), low (int), high(int), year(int), month(int)

Request Description: select from year 2016, 2017, 2018

Query Parameter(s) : None  
Route Handler : search(req, res)  
Response Type : JSON  
Response Parameters : { results (JSON array of {customer\_id (int), order\_id(int), price (float), category (string), year (int), month(int), review\_score (float), customer\_city (string)}) }  
Response description :

- default: matching results based on default param values
- with year param: results matching param inputs

---

### Sample

Path: <http://localhost:8080/search> (default category "", low 0 high 10000000 year 2018 month 8 page 1 pagesize 10)  
<http://localhost:8080/search?category=perf&low=0&high=500&year=2018&month=5&page=1&pagesize=2>  
(with param)

Response:

```
{
  "results": [
    {
      "customer_id": "fff5169e583fd07fac9fec88962f189d",
      "order_id": "000aed2e25dbad2f9ddb70584c5a2ded",
      "price": 144,
      "category": "perfumery",
      "year": 2018,
      "month": 5,
      "review_score": 1,
      "customer_city": "santa barbara d'oeste"
    },
    {
      "customer_id": "de0c1a4d8c367c58d66e61dfa379f4cf",
      "order_id": "0094bd07f49fed90209ffa62d1ef26d6",
      "price": 11.53,
      "category": "perfumery",
      "year": 2018,
      "month": 5,
      "review_score": 1,
      "customer_city": "praia grande"
    }
  ]
}
```

## Report Section

---

### Route 7

Request Path: </allmarket>  
Description : Returns a sales report for the top 5 cities with the most Walmart stores  
Request Parameter(s) : None  
Query Parameter(s) : None  
Route Handler : all\_market(req, res)

Return Type : JSON

Return Parameters : { results (JSON array of {City (string), Number of Walmart Store (int), Year (int)Number of Orders (int), Sales (float), Top Selling Product (string)}) }

Response description :

- An array of 5 cities
- 

#### Sample

Path: <http://localhost:8080/allmarket>

Response:

```
{
  "results": [
    {
      "City": "salvador",
      "Number of Walmart Stores": 26,
      "Year": 2018,
      "Number of Orders": 125,
      "Sales": 14312.609946250916,
      "Top Selling Product": "auto"
    },
    {
      "City": "curitiba",
      "Number of Walmart Stores": 16,
      "Year": 2018,
      "Number of Orders": 122,
      "Sales": 14875.089968681335,
      "Top Selling Product": "small appliances"
    },
    {
      "City": "recife",
      "Number of Walmart Stores": 14,
      "Year": 2018,
      "Number of Orders": 49,
      "Sales": 6038.710014343262,
      "Top Selling Product": "auto"
    },
    {
      "City": "porto alegre",
      "Number of Walmart Stores": 12,
      "Year": 2018,
      "Number of Orders": 122,
      "Sales": 12946.850032806396,
      "Top Selling Product": "telephony"
    },
    {
      "City": "são paulo",
      "Number of Walmart Stores": 7,
      "Year": 2018,
      "Number of Orders": 1515,
      "Sales": 164246.76998853683,
      "Top Selling Product": "auto"
    }
  ]
}
```

```
}
]
}
```

---

**Route: 8**

Request Path: `/market`

Description : Returns the market info for a specific city in Brazil

Route Parameter(s) : `city(string)`

Query Parameter(s) : None

Route Handler : `market(req, res)`

Return Type : JSON

Return Parameters : { results (JSON array of {City (string), Number of Walmart Store (int), Year (int), Number of Orders (int), Sales (float), Top Selling Product (string)}) }

Response description :

- Case 1: If the query parameter ( `city, year` ) is defined
    - Return match entries from the specific city and year.
  - Case 2: If the query parameter ( `city, year` ) is not defined
    - Return the query parameter will be set as `salvador` and `2018`
- 

**Sample**

Path: <http://localhost:8080/market> (default city `salvador`, year `2018`)

<http://localhost:8080/market?city=salvador&year=2017> (with param)

Response:

```
{
  "results": [
    {
      "City": "salvador",
      "Number of Walmart Stores": 26,
      "Year": 2017,
      "Number of Orders": 82,
      "Sales": 9299.600039958954,
      "Top Selling Product": "perfumery"
    }
  ]
}
```

---

**ROUTE 9**

Request Path: `/allreview`

Description : We list the top 10 highest average review score product. Users can query and create a review report for the top 10 products with highest average review score and standard deviation score. We also allow users to filter the review by a specific duration of time.

Request Parameter(s) : None

Request Description: N/A

Query Parameter(s) : None  
Route Handler : all\_review(req, res)  
Response Type : JSON  
Response Parameters : { results (JSON array of {product\_category (string), avg\_review\_score(float), std\_dev\_review\_score (float), review\_num (int)}) }  
Response description :  
• default: return top 10 categories and their average and standard deviation results

---

#### Sample

Path: <http://localhost:8080/allreview>

Response:

```
{
  "results": [
    {
      "product_category": "home appliances",
      "avg_review_score": 4.333333333333333,
      "std_dev_review_score": 1.1785113019775793,
      "review_num": 12
    },
    {
      "product_category": "toys",
      "avg_review_score": 4.309523809523809,
      "std_dev_review_score": 1.3360940806284,
      "review_num": 42
    },
    {
      "product_category": "audio",
      "avg_review_score": 4.285714285714286,
      "std_dev_review_score": 1.385051387833237,
      "review_num": 7
    },
    {
      "product_category": "drinks",
      "avg_review_score": 4.222222222222222,
      "std_dev_review_score": 1.227262335243029,
      "review_num": 9
    },
    {
      "product_category": "electronics",
      "avg_review_score": 4.127659574468085,
      "std_dev_review_score": 1.3932515356378636,
      "review_num": 47
    },
    {
      "product_category": "food",
      "avg_review_score": 4.125,
      "std_dev_review_score": 1.2686114456365272,
      "review_num": 8
    },
    {
      "product_category": "stationery",
      "avg_review_score": 4.096774193548387,
```

```

        "std_dev_review_score": 1.5102838594476933,
        "review_num": 31
    },
    {
        "product_category": "perfumery",
        "avg_review_score": 4.0344827586206895,
        "std_dev_review_score": 1.542115846551579,
        "review_num": 29
    },
    {
        "product_category": "small appliances",
        "avg_review_score": 4,
        "std_dev_review_score": 1.5811388300841898,
        "review_num": 12
    },
    {
        "product_category": "computers",
        "avg_review_score": 4,
        "std_dev_review_score": 1.7320508075688772,
        "review_num": 4
    }
]
}

```

#### ROUTE 10

Request Path: `/review`

Description : We list the top 10 highest average review score product. Users can query and create a review report for the top 10 products with highest average review score and standard deviation score. We also allow users to filter the review by a specific duration of time.

Request Parameter(s) : `category(string)`

Request Description: select from product category

Query Parameter(s) : None

Route Handler : `review(req, res)`

Response Type : JSON

Response Parameters : { results (JSON array of {product\_category (string), avg\_review\_score(float), std\_dev\_review\_score (float), review\_num (int)}) }

Response description :

- default: select default category "home appliances"
- with category param: return product review results for the selected category

#### Sample

Path: <http://localhost:8080/review> (default category "home appliances")

<http://localhost:8080/review?category=home appliances> (with category param)

Response:

```

{
  "results": [
    {
      "product_category": "home appliances",
      "avg_review_score": 4.333333333333333,
      "std_dev_review_score": 1.1785113019775793,
      "review_num": 12
    }
  ]
}

```



```

    }
  ]
}

```

#### ROUTE 11

Request Path: `/allhabit`

Description : Returns an array of the payment habits including the differences in total, average, max, and min payment values by credit card users and boleto (bank tickets) users from each state.

Request Parameter(s) : `page(int)`, `pagesize(int)`

Request Description: 27 results in total

Query Parameter(s) : None

Route Handler : `all_habit(req, res)`

Response Type : JSON

Response Parameters : { results (JSON array of {customer\_id (int), total\_paydiff(float), avg\_paydiff (float), max\_paydiff (float), min\_paydiff (float)}) }

Response description :

- default: return the first page of states results (page size = 10)
- with page and pagesize param: return the selected page and pagesize results

#### Sample

Path: <http://localhost:8080/allhabit> (default, page = 1, pagesize = 10)

<http://localhost:8080/allhabit?page=1&pagesize=2> (with page and pagesize param)

Response:

```

{
  "results": [
    {
      "customer_state": "SP",
      "total_paydiff": 266370.54,
      "avg_paydiff": 28.19,
      "max_paydiff": 2561.94,
      "min_paydiff": -14.38
    },
    {
      "customer_state": "RJ",
      "total_paydiff": 102473.09,
      "avg_paydiff": 41.7,
      "max_paydiff": 943.99,
      "min_paydiff": -26
    }
  ]
}

```

#### ROUTE 12

Request Path: `/habit`

Description : Returns an array of the payment habits including the differences in total, average, max, and min payment values by credit card users and boleto (bank tickets) users from each state.

Request Parameter(s) : `state(string)`

Request Description: select from Brazil's state using initials

Query Parameter(s) : None

Route Handler : `habit(req, res)`

Response Type : JSON

Response Parameters : { results (JSON array of {customer\_id (int), total\_paydiff(float), avg\_paydiff (float), max\_paydiff (float), min\_paydiff (float)}) }

Response description :

- default: return the result for state SP
- with state param: return the habit result for the specific state input

---

**Sample**

Path: <http://localhost:8080/habit> (default state SP)  
<http://localhost:8080/habit?state=SP> (with param)

Response:

```
{
  "results": [
    {
      "customer_state": "SP",
      "total_paydiff": 266370.54,
      "avg_paydiff": 28.19,
      "max_paydiff": 2561.94,
      "min_paydiff": -14.38
    }
  ]
}
```

**ROUTE 13**

Request Path: [/transaction](#)

Description : Returns 10 of the most recent transactions

Request Parameter(s) : None

Request Description: N/A

Query Parameter(s) : None

Route Handler : transaction(req, res)

Response Type : JSON

Response Parameters : { results (JSON array of {, order\_id(int),customer\_id (int), order\_purchase\_timestamp (timestamp), payment\_value (float)}) }

Response description :

- return the order id, customer id, transaction time and the transaction value for 10 of the most recent transactions happened in Brazil ecommerce platform.

---

**Sample**

Path: <http://localhost:8080/transaction>

Response:

```
{
  "results": [
    {
      "order_id": "35a972d7f8436f405b56e36add1a7140",
      "customer_id": "898b7fee99c4e42170ab69ba59be0a8b",
      "order_purchase_timestamp": "2018-08-29T19:00:37.000Z",
      "Payment_value": 93.75
    },
    {
      "order_id": "912859fef5a0bd5059b6d48fa79d121a",
      "customer_id": "b8c19e70d00f6927388e4f31c923d785",
      "order_purchase_timestamp": "2018-08-29T13:48:09.000Z",
      "Payment_value": 178.25
    }
  ]
}
```

```

    },
    {
      "order_id": "fb393211459aac00af932cd7ab4fa2cc",
      "customer_id": "54365416b7ef5599f54a6c7821d5d290",
      "order_purchase_timestamp": "2018-08-29T13:14:11.000Z",
      "Payment_value": 106.95
    },
    {
      "order_id": "bee12e8653a04e76786e8891cfb6330a",
      "customer_id": "448945bc713d98b6726e82eda6249b9e",
      "order_purchase_timestamp": "2018-08-29T12:46:11.000Z",
      "Payment_value": 497.25
    },
    {
      "order_id": "4c867c2aac81653679eff07e922441a0",
      "customer_id": "8ffe52ac7d480a27180338e697eec534",
      "order_purchase_timestamp": "2018-08-29T02:51:54.000Z",
      "Payment_value": 38.36
    },
    {
      "order_id": "e7c290bfc31d7eed478c3d3d2d4d2953",
      "customer_id": "004440537b68545ca3c341d7279bc4c0",
      "order_purchase_timestamp": "2018-08-29T02:30:32.000Z",
      "Payment_value": 98.04
    },
    {
      "order_id": "dbb786f88b6d4e52fe3cb5d771b979d6",
      "customer_id": "478778636c75019554439f75286a22e3",
      "order_purchase_timestamp": "2018-08-29T01:56:30.000Z",
      "Payment_value": 161.97
    },
    {
      "order_id": "00b44ba3d7c4a5e9a9ebafef9150781d",
      "customer_id": "0b5f6687d659478f1747caed607c4ec5",
      "order_purchase_timestamp": "2018-08-29T01:10:46.000Z",
      "Payment_value": 67.58
    },
    {
      "order_id": "0ac69790e2a6e4c075edbd78648a3594",
      "customer_id": "1c3d8766b5f8b24d7e95001ce31d1d38",
      "order_purchase_timestamp": "2018-08-28T22:49:20.000Z",
      "Payment_value": 14.89
    },
    {
      "order_id": "93d7e1f0c5415c1a0679635187700f1d",
      "customer_id": "1288df1d9da5e124c94b06a6ba3ca50b",
      "order_purchase_timestamp": "2018-08-28T20:33:22.000Z",
      "Payment_value": 216.58
    }
  ]
}

```