

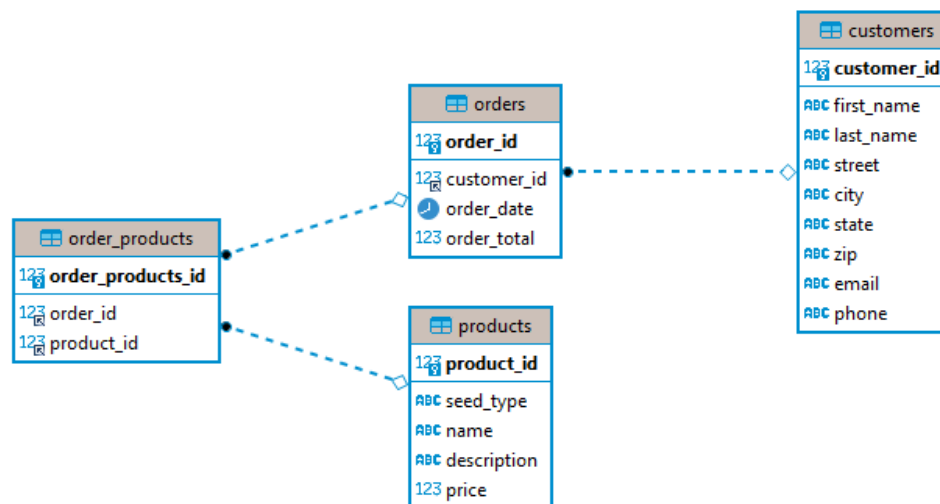
GitHub Link:

<https://github.com/AlishaThomas/Seed-StoreFinalProject.git>

Seed Store API:

This is a web-based heirloom seed store where customers can browse and choose from a variety of products and create orders. The many to many relationship is between the products and orders table where many products can be included on many orders and many orders can contain many products. The API allows CRUD operations on all the entities. On the customers entity, all four CRUD operations can be applied. The products entity is capable of the read operation when supplied with product name and type parameters and can also read and return a list of all available products which requires no parameters. The orders entity has a create operation which still needs some work but will eventually create an order. Also, I plan to work out how to properly save an order and calculate an order total.

ERD:



Customer CRUD operations:

Update

Updates customer information requiring customer Id, first name and last name as parameters and returns 200 OK status.

basic-customer-controller

PUT /seedstore/update-customer Updates a customer.

Updates a customer using customer id, first name, last name street, city, state, zip, email and phone number.

Parameters Cancel

Name	Description
customer_id	
integer(\$int32)	The customer id.
(query)	
	<input type="text" value="1"/>
first_name	
string	The customer first name.
(query)	
	<input type="text" value="Alisha"/>
last_name	
string	The customer last name.
(query)	
	<input type="text" value="Thomas"/>
street	
string	The customer's street.
(query)	
	<input type="text" value="707 S Forest Ave"/>
city	
string	The customer's city.

Curl

```
curl -X 'PUT' \
'http://localhost:8080/seedstore/update-customer?customer_id=1&first_name=Alisha&last_name=Thomas&street=707%20S%20Forest%20Ave&city=Springfield&state=MO&zip=65802&email=adthomas02840@gmail.com&phone=141742'
-H 'accept: application/json'
```

Request URL

```
http://localhost:8080/seedstore/update-customer?customer_id=1&first_name=Alisha&last_name=Thomas&street=707%20S%20Forest%20Ave&city=Springfield&state=MO&zip=65802&email=adthomas02840@gmail.com&phone=1417425544
```

Server response

Code	Details
200	<div>Response body</div> <pre>{ "first_name": "Alisha", "last_name": "Thomas", "street": "707 S Forest Ave", "city": "Springfield", "state": "MO", "zip": "65802", "email": "adthomas028@gmail.com", "phone": "1417425544" }</pre> <div> Download</div> <div>Response headers</div> <pre>connection: keep-alive content-type: application/json date: Fri, 18 Mar 2022 18:30:52 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

Responses

Code	Description	Links
------	-------------	-------

Create:

Creates a new customer when all customer parameters are supplied. Returns 201 Created Status.

POST /seedstore/add-customer/ Creates a new customer.

Returns the created customer

Parameters

No parameters

Cancel

Reset

Request body required

application/json

```
{
  "first_name": "Alisha",
  "last_name": "Thomas",
  "street": "Street",
  "city": "City",
  "state": "MO",
  "zip": "65432",
  "email": "email@email.com",
  "phone": "654798351"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/seedstore/add-customer/' \
  -H 'accept: Application/json' \
  -H 'Content-type: application/json' \
  -d '{
    "first_name": "Alisha",
    "last_name": "Thomas",
    "street": "Street",
    "city": "City",
    "state": "MO",
    "zip": "65432",
    "email": "email@email.com",
    "phone": "654798351"
  }'
```

Request URL

http://localhost:8080/seedstore/add-customer/

Server response

Code

Details

201

Response body

```
{
  "first_name": "Alisha",
  "last_name": "Thomas",
  "street": "MO",
  "city": "City",
  "state": "MO",
  "zip": "65432",
  "email": "email@email.com",
  "phone": "654798351"
}
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Fri, 18 Mar 2022 18:38:21 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Reads list of customers:

Takes customer Id, first name or last name as parameters and returns selected customer. Returns 200 OK status.

GET

/seedstore/customers

Returns a list of customers.

Returns a list of customers given customer id and/or first name and/or last name.

Parameters

Cancel

Name	Description
customer_id integer(\$int32) (query)	The customer id.
<input type="text" value="1"/>	
first_name string (query)	The customer first name.
<input type="text" value="Alisha"/>	
last_name string (query)	The customer last name.
<input type="text" value="Thomas"/>	

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/seedstore/customers?customer_id=1&first_name=Alisha&last_name=Thomas' \
  -H 'accept: application/json'
```

Request URL

http://localhost:8080/seedstore/customers?customer_id=1&first_name=Alisha&last_name=Thomas

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "first_name": "Alisha", "last_name": "Thomas", "street": "707 S Forest Ave", "city": "Springfield", "state": "MO", "zip": "65802", "email": "adthomas028@gmail.com", "phone": "3417425544" }</pre></div><div>Download</div></div> <div><div>Response headers</div><div><pre>connection: keep-alive content-type: application/json date: Fri, 18 Mar 2022 18:39:21 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre></div></div>

Responses

Code	Description	Links
200	A customer has been selected	No links

Delete

Requires customer Id as a parameter to delete an existing customer and returns 200 OK status.

DELETE /seedstore/delete-customer Deletes a customer.

Deletes a customer using customer id, first name, last name, street, city, state, zip, email and phone number.

Parameters Cancel

Name	Description
customer_id integer(\$Int32) (query)	The customer id.
<input type="text" value="1"/>	
first_name string (query)	The customer first name.
<input type="text" value="Alisha"/>	
last_name string (query)	The customer last name.
<input type="text" value="Thomas"/>	
street string (query)	The customer's street.
<input type="text" value="707 S Forest Ave"/>	
city string (query)	The customer's city.
<input type="text" value="Springfield"/>	
state string (query)	The customer's state.

Curl

```
curl -X 'DELETE' \
  "http://localhost:8080/seedstore/delete-customer?customer_id=1&first_name=Alisha&last_name=Thomas&street=707%20S%20Forest%20Ave&city=Springfield&state=MO&zip=65802&email=adthomas028@gmail.com&phone=1417425447" \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/seedstore/delete-customer?customer_id=1&first_name=Alisha&last_name=Thomas&street=707%20S%20Forest%20Ave&city=Springfield&state=MO&zip=65802&email=adthomas028@gmail.com&phone=1417425447
```

Server response

Code	Details
200	<div>Response body</div> <pre>{ "first_name": "Alisha", "last_name": "Thomas", "street": "707 S Forest Ave", "city": "Springfield", "state": "MO", "zip": "65802", "email": "adthomas028@gmail.com", "phone": "1417425447" }</pre> <div>Download</div> <div>Response headers</div> <pre>connection: keep-alive content-type: application/json date: Fri, 18 Mar 2022 18:40:42 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

Responses

Code	Description	Links
200	A customer is deleted.	No links

Products CRUD Operations:

Reads and returns product when product type and name are supplied. Returns 200 OK status:

basic-products-controller

GET /seedstore/products/get-seeds Returns a product.

Returns a product given type and name or type.

Parameters

Name

Description

seedType

The product type.

string

(query)

Broccoli

name

The product name.

string

(query)

Rapini Broccoli

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/seedstore/products/get-seeds?seedType=Broccoli&name=Rapini%20broccoli' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/seedstore/products/get-seeds?seedType=Broccoli&name=Rapini%20broccoli
```

Server response

Code

Details

200

Response body

```
[
  {
    "productId": 1,
    "seedType": "Broccoli",
    "name": "Rapini Broccoli",
    "price": 1,
    "description": "This is a non-heading, Italian broccoli with asparagus-like shoots and leaves."
  },
  {
    "productId": 21,
    "seedType": "Broccoli",
    "name": "Rapini Broccoli",
    "price": 1,
    "description": "This is a non-heading, Italian broccoli with asparagus-like shoots and leaves."
  }
]
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Fri, 18 Mar 2022 19:09:15 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Read

Returns a product list and takes no parameters. Returns 200 OK status:

GET /seedstore/products/get-all-seeds Returns a list of seeds.

Returns a list of seeds.

Parameters

No parameters

Cancel

Execute

Clear

Responses

Curl

curl -X 'GET' \ 'http://localhost:8080/seedstore/products/get-all-seeds' \ -H 'accept: Application/json'

Request URL

http://localhost:8080/seedstore/products/get-all-seeds

Server response

Code

Details

200
Undocumented

Response body

```
{
  "seedType": "BROCCOLI",
  "name": "Rapini Broccoli",
  "price": 3,
  "description": "This is a non-heading, Italian Broccoli with asparagus-like shoots and leaves."
},
{
  "seedType": "BROCCOLI",
  "name": "Early Purple Sprouting Broccoli",
  "price": 3,
  "description": "A UK Broccoli ideal for mild climates. Requires overwintering to produce florets."
},
{
  "seedType": "BROCCOLI",
  "name": "Waltham 29 Broccoli",
  "price": 3,
  "description": "This is a standard type Broccoli that green heads and some side shoots."
},
{
  "seedType": "BRUSSELS_SPROUTS",
  "name": "Groninger Brussels Sprouts",
  "price": 3,
  "description": "Easily grown and steady producing. Good for first time growers. Can handle growing in light frost."
},
{
  "seedType": "BRUSSELS_SPROUTS",
  "name": "Red Rutine Brussels Sproute",

```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Sat, 19 Mar 2022 01:14:24 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Orders CRUD Operations:

Supposed to create a new order and calculate an order total. This operation still needs some work.

POST /seedstore/orders Creates an order.

Creates an order.

Parameters

Cancel Reset

No parameters

Request body required

application/json

```
{
  "customer": "1",
  "productId": 1,
  "seedType": "Broccoli",
  "name": "Rapini Broccoli",
  "price": 3,
  "orderDate": "2022-03-18T19:41:59.485Z"
}
```

Execute

Clear

Curl

```
curl -X 'POST' \
  'http://localhost:8080/seedstore/orders' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "customer": "1",
    "productId": 1,
    "seedType": "Broccoli",
    "name": "Rapini Broccoli",
    "price": 3,
    "orderDate": "2022-03-18T19:41:59.485Z"
  }'
```

Request URL

http://localhost:8080/seedstore/orders

Server response

Code

Details

500

Error: response status is 500

Response body

```
{
  "timestamp": "2022-03-18T19:42:27.714+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "trace": "org.springframework.dao.TransientDataAccessResourceException: PreparedStatementCallback; SQL [SELECT * FROM customers WHERE customer_id = ?]; Illegal operation on empty result set.; nested exception is java.sql.SQLException: Illegal operation on empty result set.\r\n\tat org.springframework.jdbc.support.SQLExceptionTranslator.doTranslate(SQLExceptionTranslator.java:110)\r\n\tat org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTranslator.java:70)\r\n\tat org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTranslator.java:79)\r\n\tat org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTranslator.java:79)\r\n\tat org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTranslator.java:79)\r\n\tat org.springframework.jdbc.core.JdbcTemplate.translateException(JdbcTemplate.java:1541)\r\n\tat org.springframework.jdbc.core.JdbcTemplate.execute(JdbcTemplate.java:667)\r\n\tat org.springframework.jdbc.core.JdbcTemplate.query(JdbcTemplate.java:713)\r\n\tat org.springframework.jdbc.core.JdbcTemplate.query(JdbcTemplate.java:738)\r\n\tat org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate.query(NamedParam
```