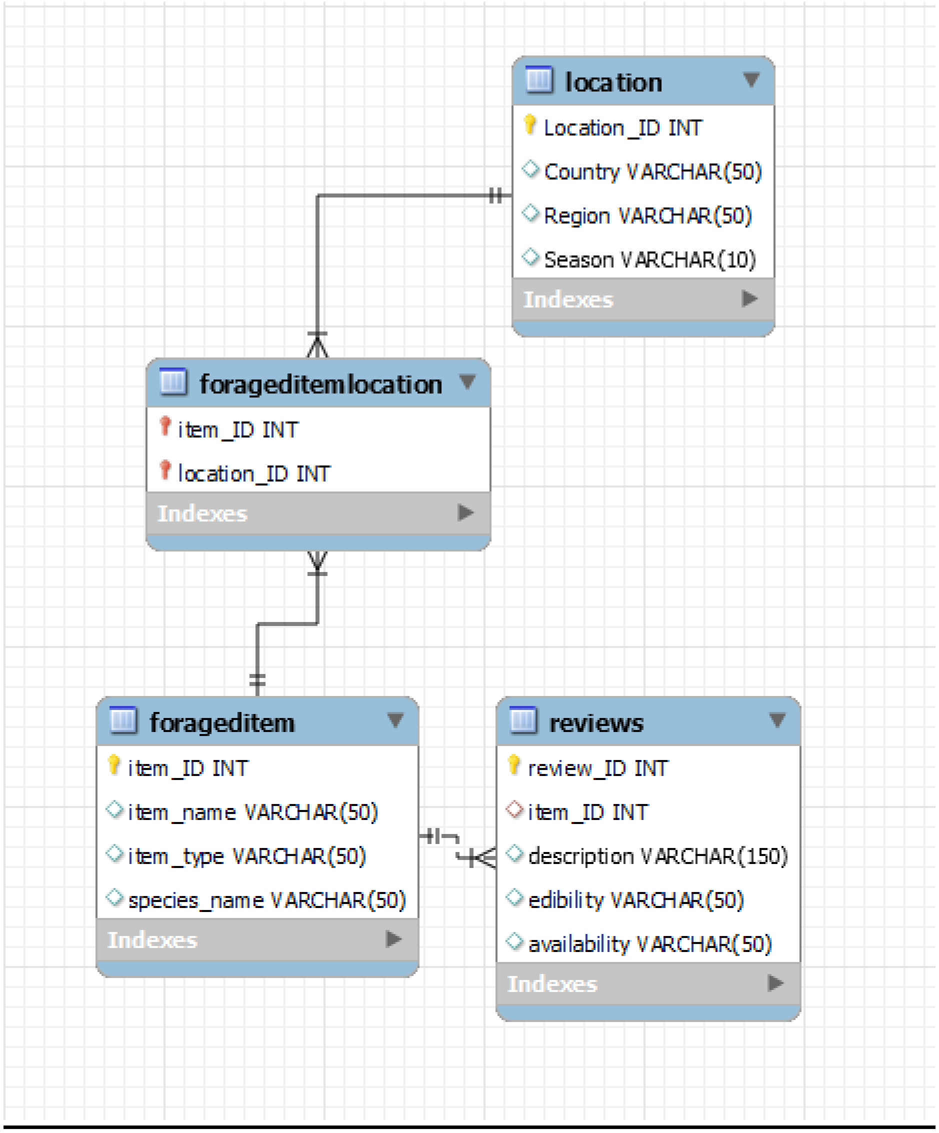
Final project submission

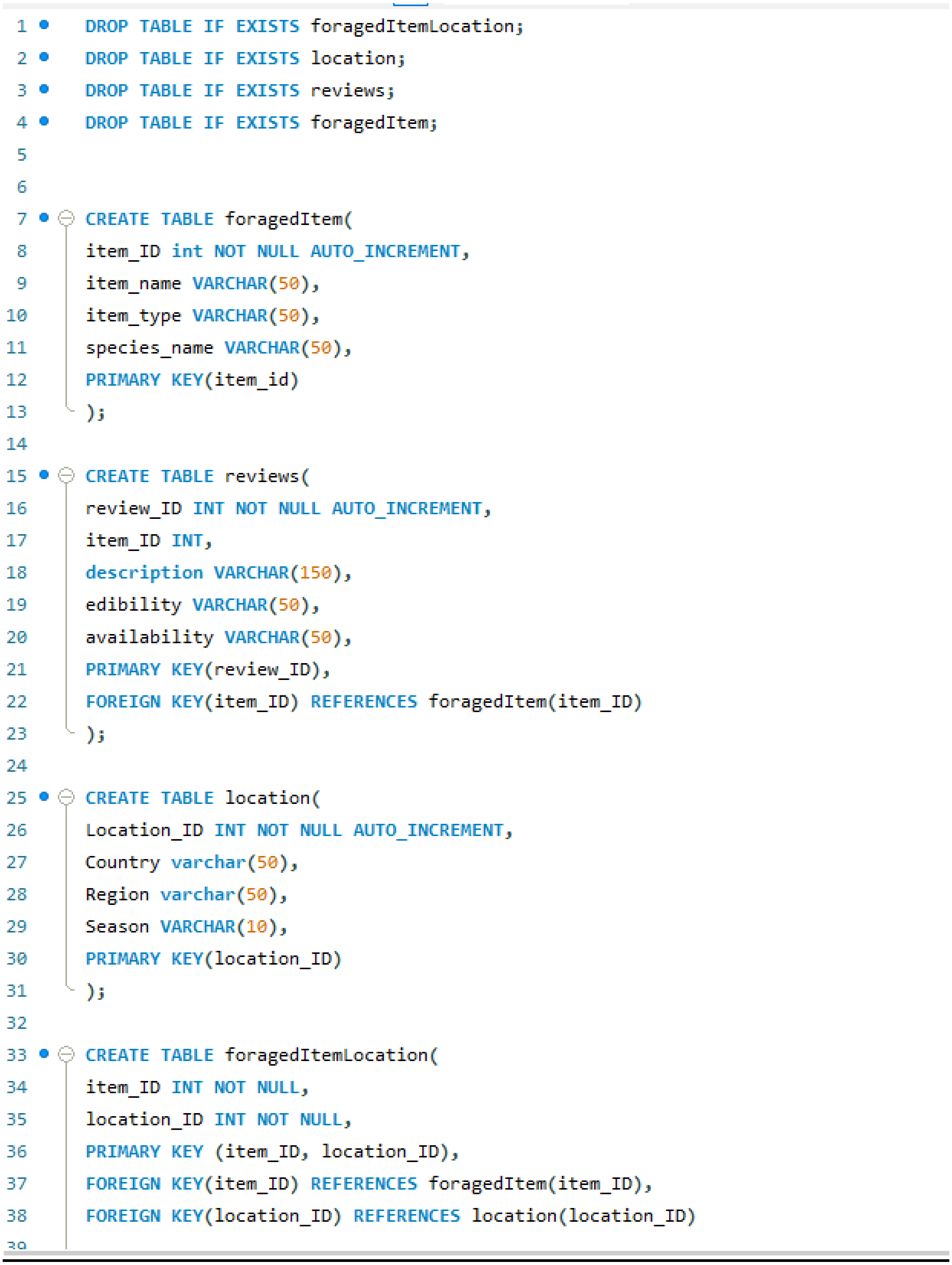
sql scripts and ERD (will also be located on my github):

ERD:



Foraging DB script:

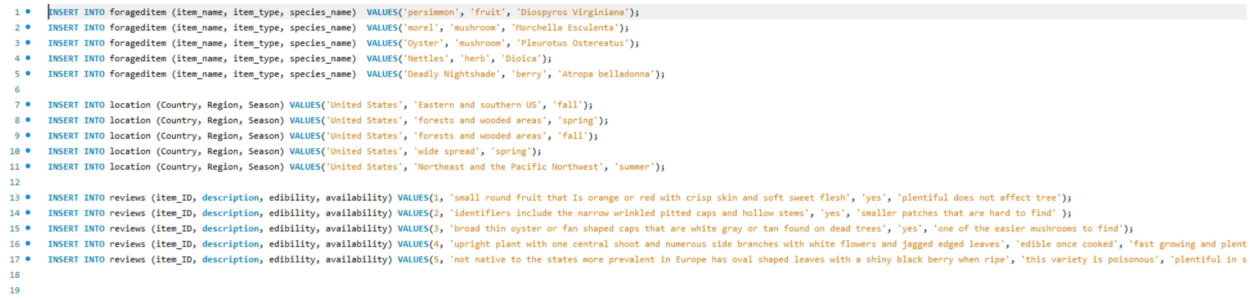
Final project submission



```sql
DROP TABLE IF EXISTS foragedItemLocation;
DROP TABLE IF EXISTS location;
DROP TABLE IF EXISTS reviews;
DROP TABLE IF EXISTS foragedItem;


CREATE TABLE foragedItem(
item_ID int NOT NULL AUTO_INCREMENT,
item_name VARCHAR(50),
item_type VARCHAR(50),
species_name VARCHAR(50),
PRIMARY KEY(item_id)
);

CREATE TABLE reviews(
review_ID INT NOT NULL AUTO_INCREMENT,
item_ID INT,
description VARCHAR(150),
edibility VARCHAR(50),
availability VARCHAR(50),
PRIMARY KEY(review_ID),
FOREIGN KEY(item_ID) REFERENCES foragedItem(item_ID)
);

CREATE TABLE location(
Location_ID INT NOT NULL AUTO_INCREMENT,
Country varchar(50),
Region varchar(50),
Season VARCHAR(10),
PRIMARY KEY(location_ID)
);

CREATE TABLE foragedItemLocation(
item_ID INT NOT NULL,
location_ID INT NOT NULL,
PRIMARY KEY (item_ID, location_ID),
FOREIGN KEY(item_ID) REFERENCES foragedItem(item_ID),
FOREIGN KEY(location_ID) REFERENCES location(location_ID)
```

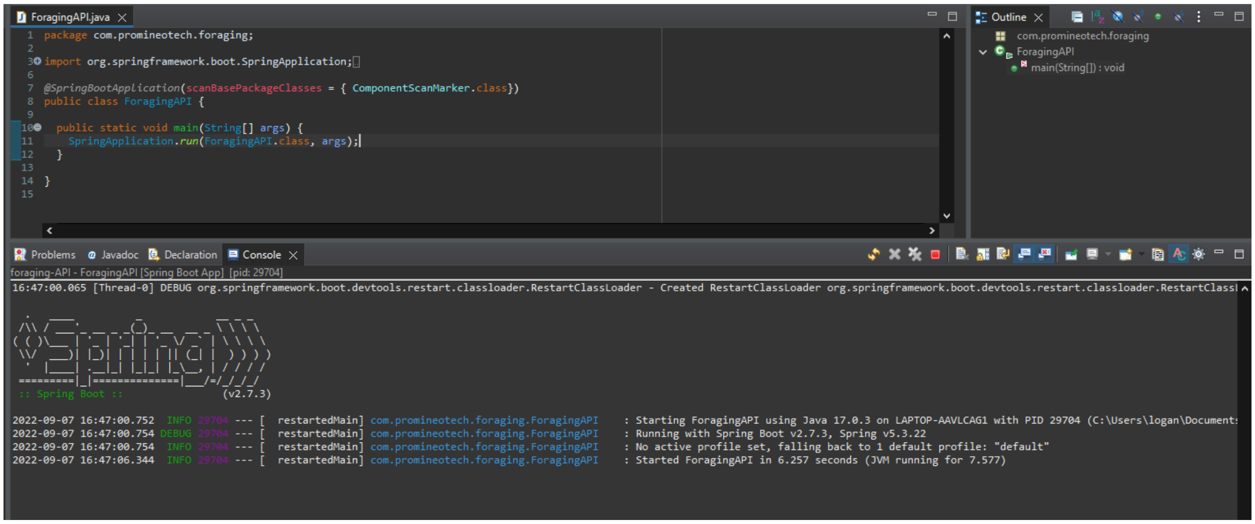
Final project submission

Foraging data script:



```sql
INSERT INTO forageditem (item_name, item_type, species_name)  VALUES('persimmon', 'fruit', 'Diospyros Virginiana');
INSERT INTO forageditem (item_name, item_type, species_name)  VALUES('morel', 'mushroom', 'Morchella Esculenta');
INSERT INTO forageditem (item_name, item_type, species_name)  VALUES('Oyster', 'mushroom', 'Pleurotus Ostereatus');
INSERT INTO forageditem (item_name, item_type, species_name)  VALUES('Nettles', 'herb', 'Dioica');
INSERT INTO forageditem (item_name, item_type, species_name)  VALUES('Deadly Nightshade', 'berry', 'Atropa belladonna');

INSERT INTO location (Country, Region, Season) VALUES('United States', 'Eastern and southern US', 'fall');
INSERT INTO location (Country, Region, Season) VALUES('United States', 'forests and wooded areas', 'spring');
INSERT INTO location (Country, Region, Season) VALUES('United States', 'forests and wooded areas', 'fall');
INSERT INTO location (Country, Region, Season) VALUES('United States', 'wide spread', 'spring');
INSERT INTO location (Country, Region, Season) VALUES('United States', 'Northeast and the Pacific Northwest', 'summer');

INSERT INTO reviews (item_ID, description, edibility, availability) VALUES(1, 'small round fruit that Is orange or red with crisp skin and soft sweet flesh', 'yes', 'plentiful does not affect tree');
INSERT INTO reviews (item_ID, description, edibility, availability) VALUES(2, 'identifiers include the narrow wrinkled pitted caps and hollow stems', 'yes', 'smaller patches that are hard to find' );
INSERT INTO reviews (item_ID, description, edibility, availability) VALUES(3, 'broad thin oyster or fan shaped caps that are white gray or tan found on dead trees', 'yes', 'one of the easier mushrooms to find');
INSERT INTO reviews (item_ID, description, edibility, availability) VALUES(4, 'upright plant with one central shoot and numerous side branches with white flowers and jagged edged leaves', 'edible once cooked', 'fast growing and plent
INSERT INTO reviews (item_ID, description, edibility, availability) VALUES(5, 'not native to the states more prevalent in Europe has oval shaped leaves with a shiny black berry when ripe', 'this variety is poisonous', 'plentiful in s
```

Pictures of Foraging-API running in springboot:

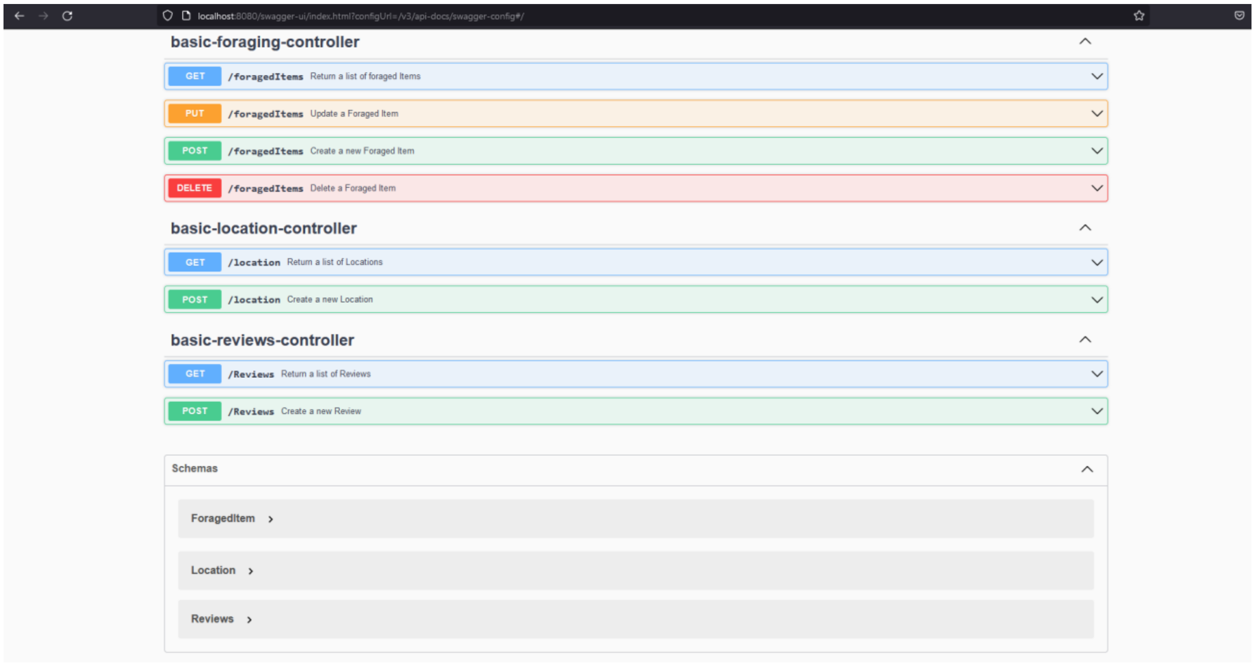


```java
package com.promineotech.foraging;

import org.springframework.boot.SpringApplication;

@SpringBootApplication(scanBasePackageClasses = { ComponentScanMarker.class})
public class ForagingAPI {

  public static void main(String[] args) {
    SpringApplication.run(ForagingAPI.class, args);
  }

}
```

```
foraging-API - ForagingAPI [Spring Boot App]  [pid: 29704]
16:47:00.065 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created RestartClassLoader org.springframework.boot.devtools.restart.classloader.RestartClass

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.7.3)

2022-09-07 16:47:00.752  INFO 29704 --- [  restartedMain] com.promineotech.foraging.ForagingAPI    : Starting ForagingAPI using Java 17.0.3 on LAPTOP-AAVLCAG1 with PID 29704 (C:\Users\logan\Documents
2022-09-07 16:47:00.754 DEBUG 29704 --- [  restartedMain] com.promineotech.foraging.ForagingAPI    : Running with Spring Boot v2.7.3, Spring v5.3.22
2022-09-07 16:47:00.754  INFO 29704 --- [  restartedMain] com.promineotech.foraging.ForagingAPI    : No active profile set, falling back to 1 default profile: "default"
2022-09-07 16:47:06.344  INFO 29704 --- [  restartedMain] com.promineotech.foraging.ForagingAPI    : Started ForagingAPI in 6.257 seconds (JVM running for 7.577)
```

Pictures of foraging running in swagger:

Final project submission



localhost:8080/swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config#/

basic-foraging-controller

GET /foragedItems Return a list of foraged Items

PUT /foragedItems Update a Foraged Item

POST /foragedItems Create a new Foraged Item

DELETE /foragedItems Delete a Foraged Item

basic-location-controller

GET /location Return a list of Locations

POST /location Create a new Location

basic-reviews-controller

GET /Reviews Return a list of Reviews

POST /Reviews Create a new Review

Schemas

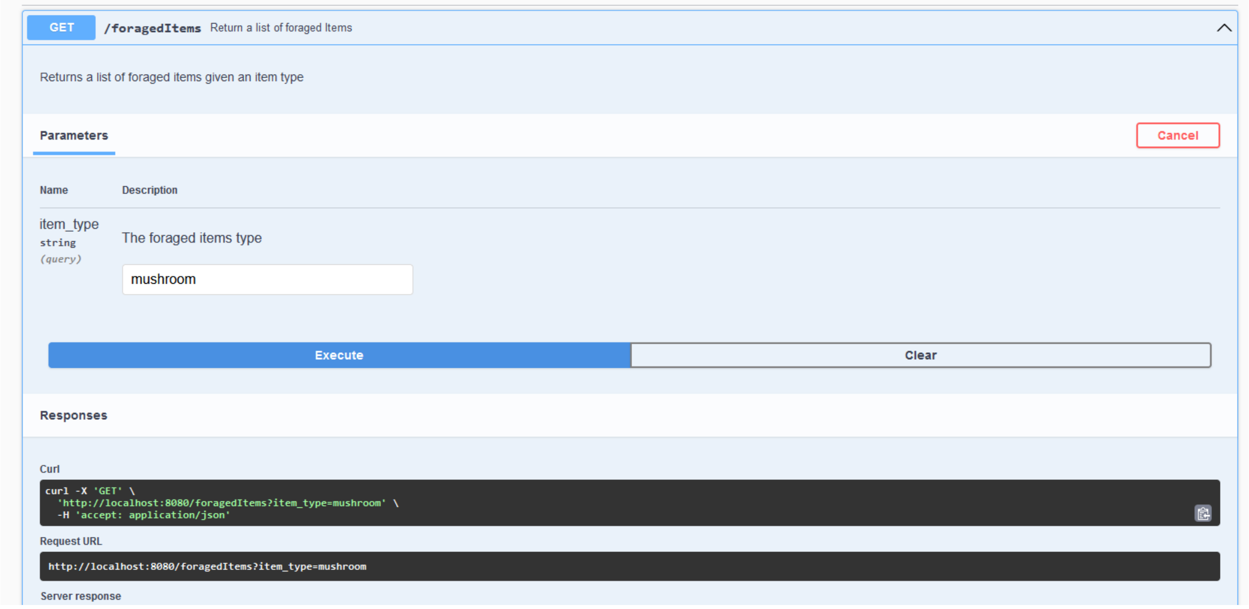
ForagedItem ›

Location ›

Reviews ›

Basic-foraging-controller:

GET:



GET /foragedItems Return a list of foraged Items

Returns a list of foraged items given an item type

Name	Description
item_type string *(query)*	The foraged items type mushroom

Responses

Curl

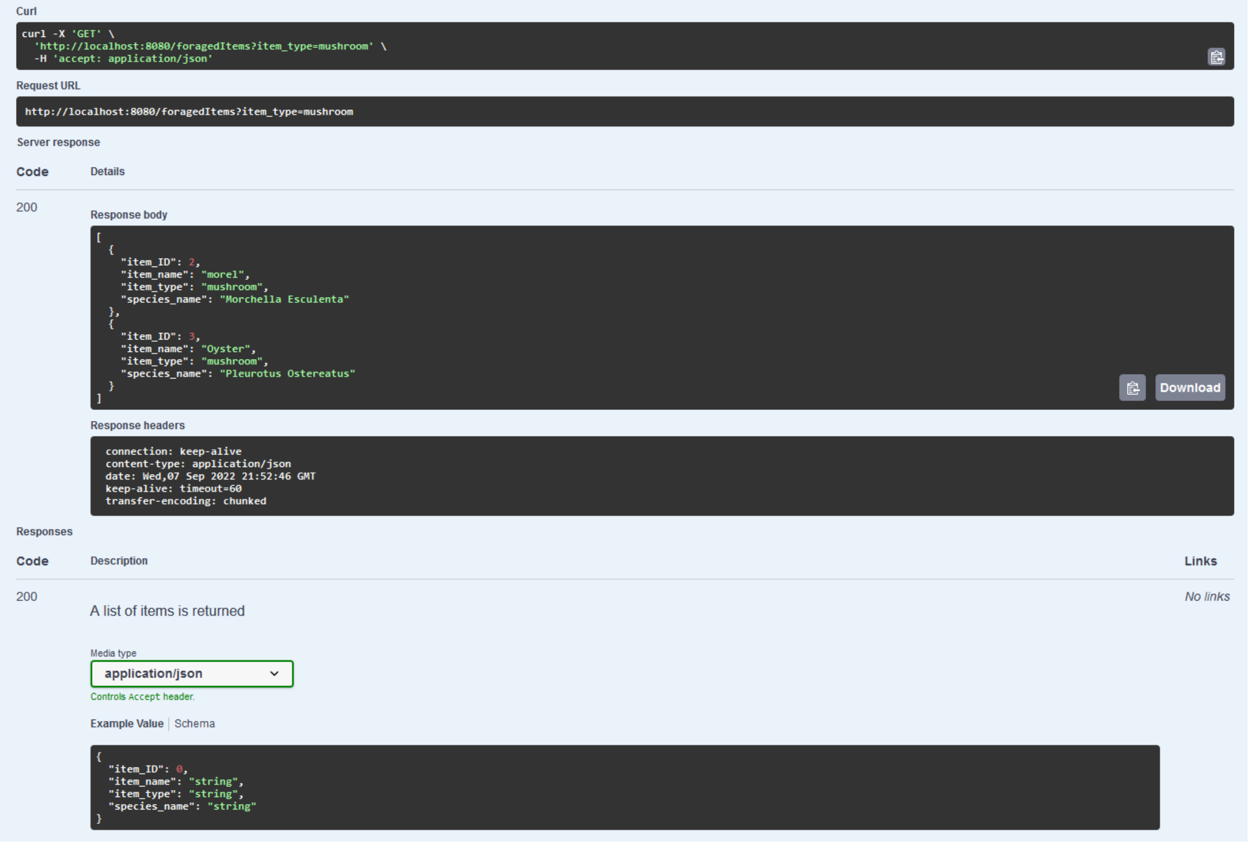
```
curl -X 'GET' \
  'http://localhost:8080/foragedItems?item_type=mushroom' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/foragedItems?item_type=mushroom
```

Server response

Final project submission



Curl

```
curl -X 'GET' \
  'http://localhost:8080/foragedItems?item_type=mushroom' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/foragedItems?item_type=mushroom
```

Server response

Code Details

200

Response body

```
[
  {
    "item_ID": 2,
    "item_name": "morel",
    "item_type": "mushroom",
    "species_name": "Morchella Esculenta"
  },
  {
    "item_ID": 3,
    "item_name": "Oyster",
    "item_type": "mushroom",
    "species_name": "Pleurotus Ostereatus"
  }
]
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,07 Sep 2022 21:52:46 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```

Responses

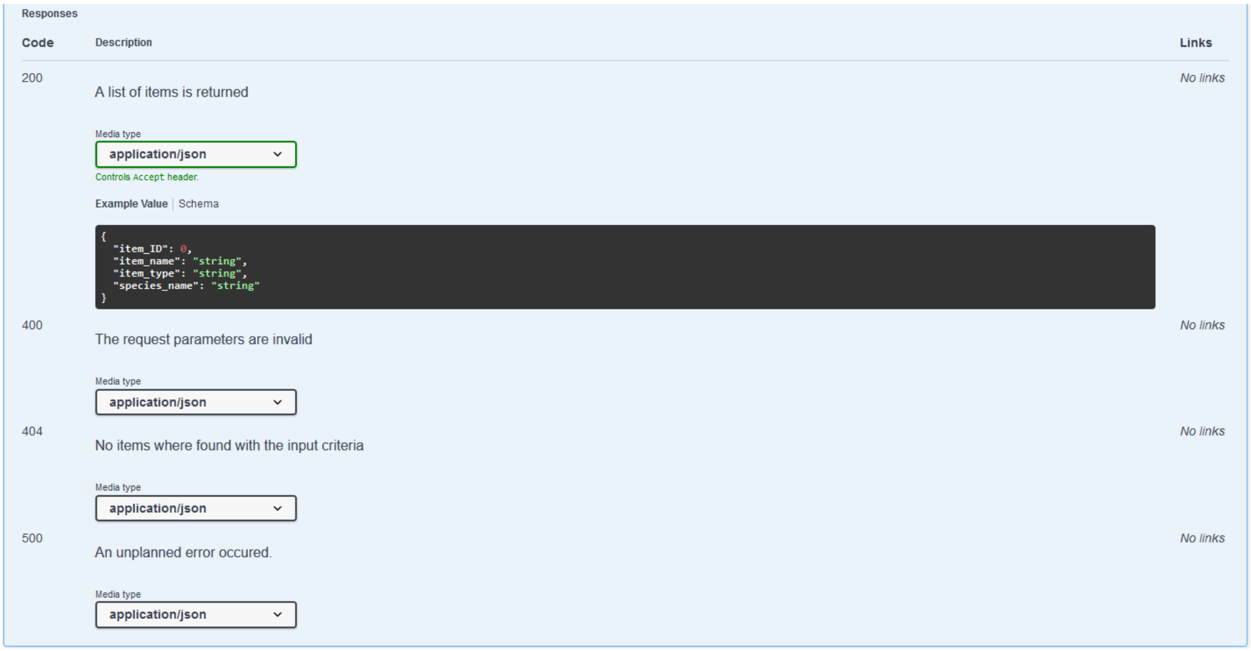
Code	Description	Links
200	A list of items is returned	*No links*

Media type: application/json

Controls Accept header.

Example Value | Schema

```
{
  "item_ID": 0,
  "item_name": "string",
  "item_type": "string",
  "species_name": "string"
}
```



Responses

Code	Description	Links
200	A list of items is returned	*No links*

Media type: application/json

Controls Accept header.

Example Value | Schema

```
{
  "item_ID": 0,
  "item_name": "string",
  "item_type": "string",
  "species_name": "string"
}
```

Code	Description	Links
400	The request parameters are invalid	*No links*

Media type: application/json

Code	Description	Links
404	No items where found with the input criteria	*No links*

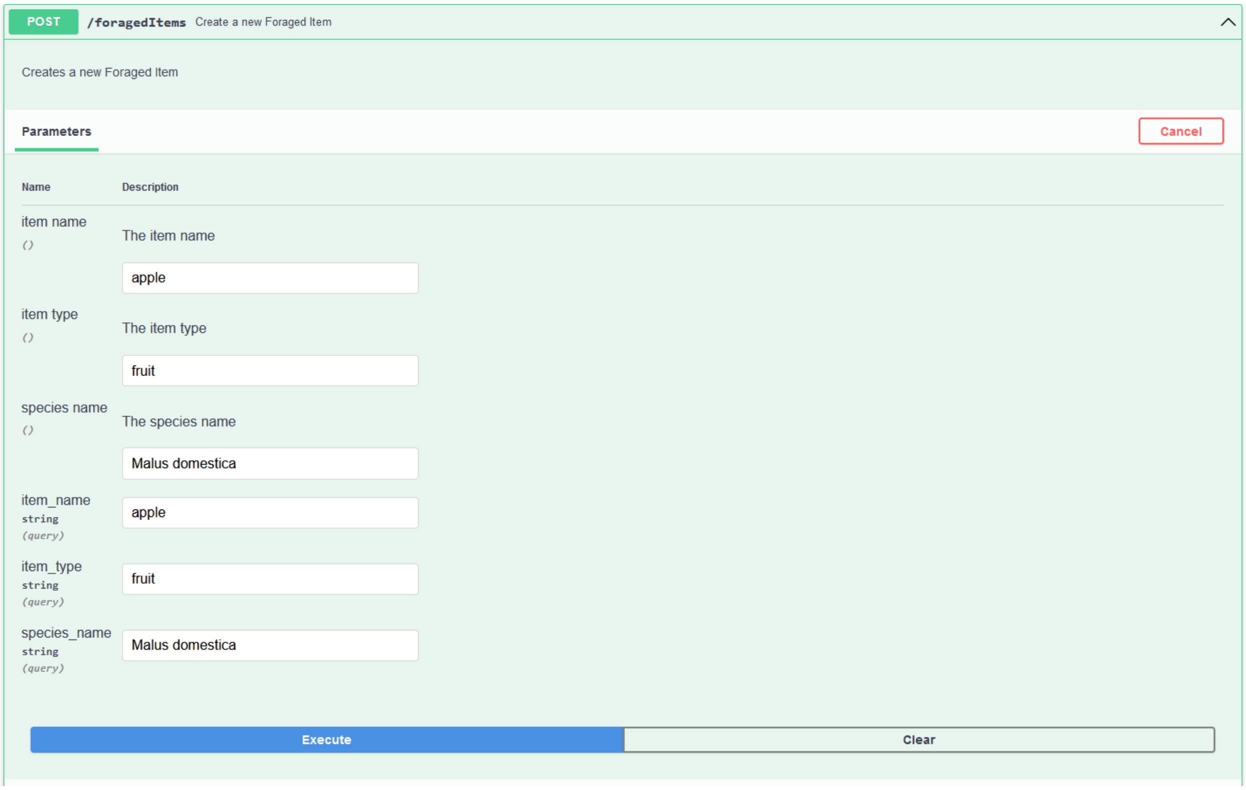
Media type: application/json

Code	Description	Links
500	An unplanned error occured.	*No links*

Media type: application/json

Final project submission

POST:



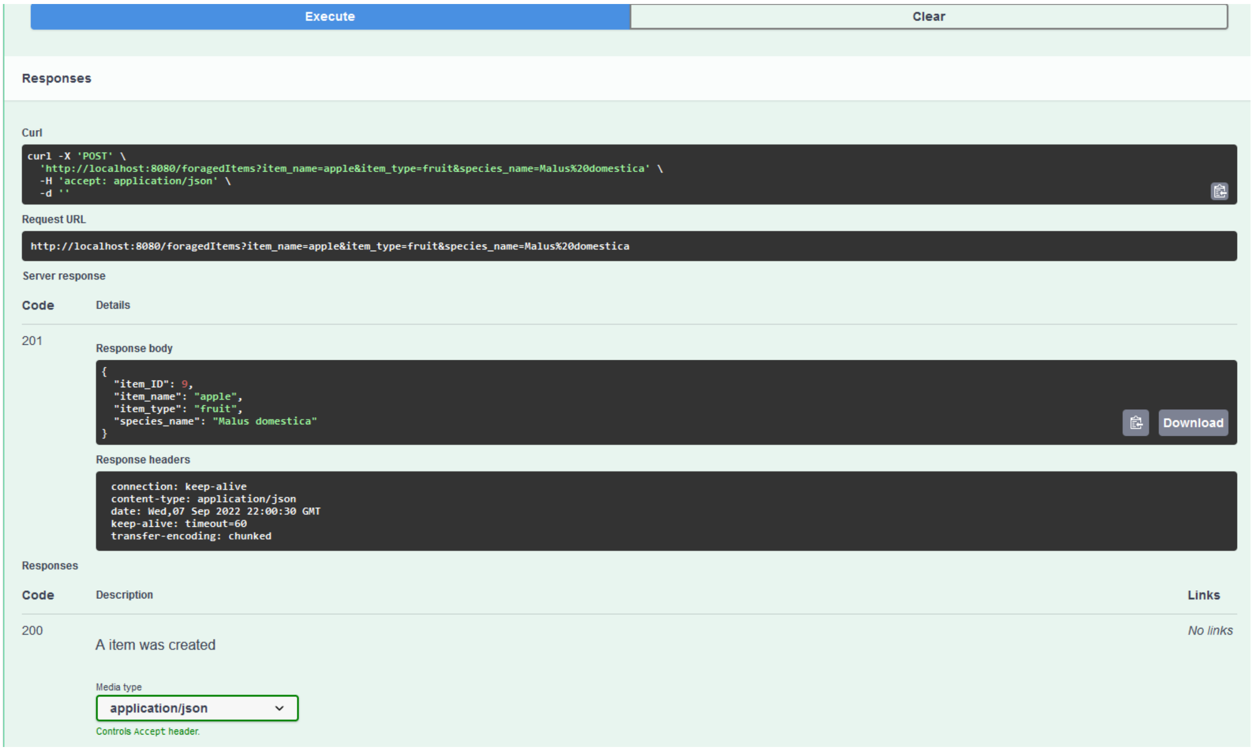
/foragedItems Create a new Foraged Item

Creates a new Foraged Item

Parameters

Name	Description
item name ()	The item name apple
item type ()	The item type fruit
species name ()	The species name Malus domestica
item_name string *(query)*	apple
item_type string *(query)*	fruit
species_name string *(query)*	Malus domestica

Final project submission



Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/foragedItems?item_name=apple&item_type=fruit&species_name=Malus%20domestica' \
  -H 'accept: application/json' \
  -d ''
```

Request URL

```
http://localhost:8080/foragedItems?item_name=apple&item_type=fruit&species_name=Malus%20domestica
```

Server response

Code Details

201

Response body

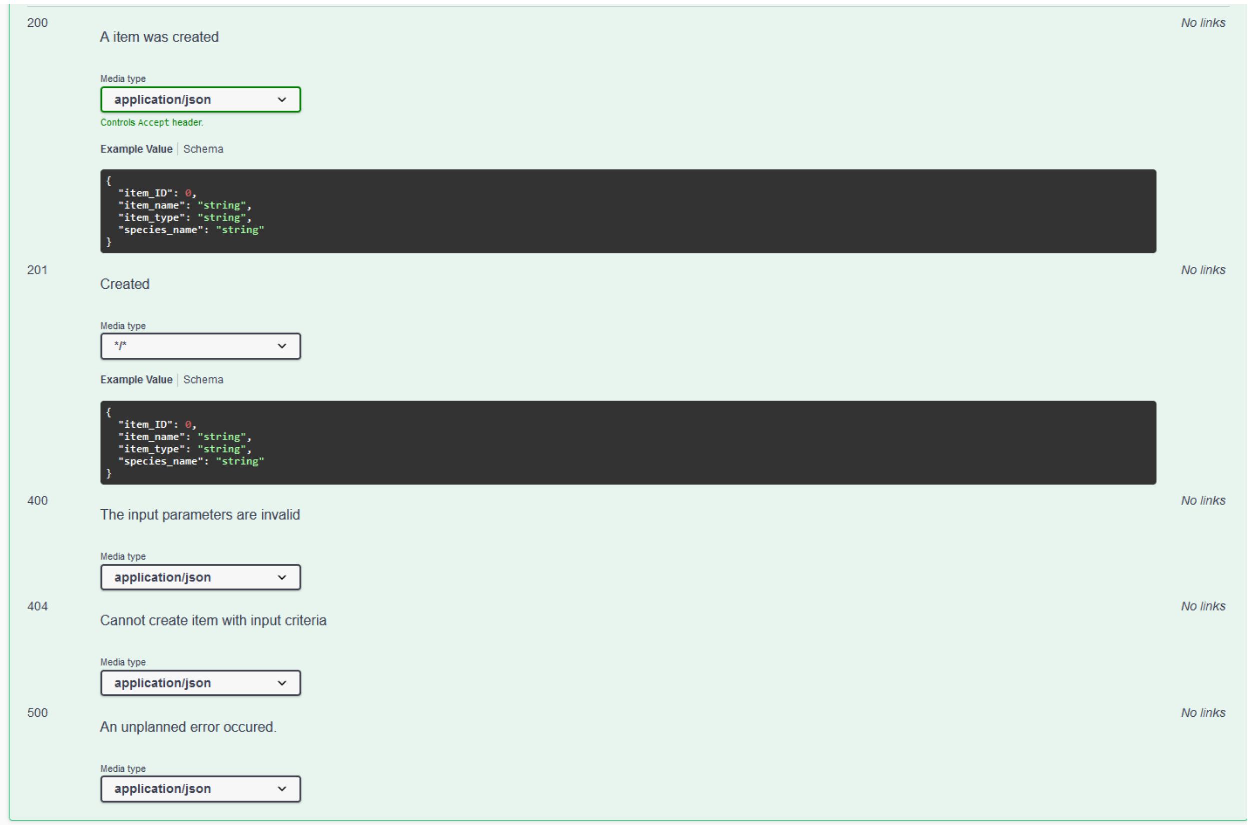
```
{
  "item_ID": 9,
  "item_name": "apple",
  "item_type": "fruit",
  "species_name": "Malus domestica"
}
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,07 Sep 2022 22:00:30 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```

Responses

Code	Description	Links
200	A item was created Media type application/json Controls Accept header.	*No links*



Code	Description	Links
200	A item was created Media type application/json Controls Accept header. **Example Value** | Schema	*No links*

```
{
  "item_ID": 0,
  "item_name": "string",
  "item_type": "string",
  "species_name": "string"
}
```

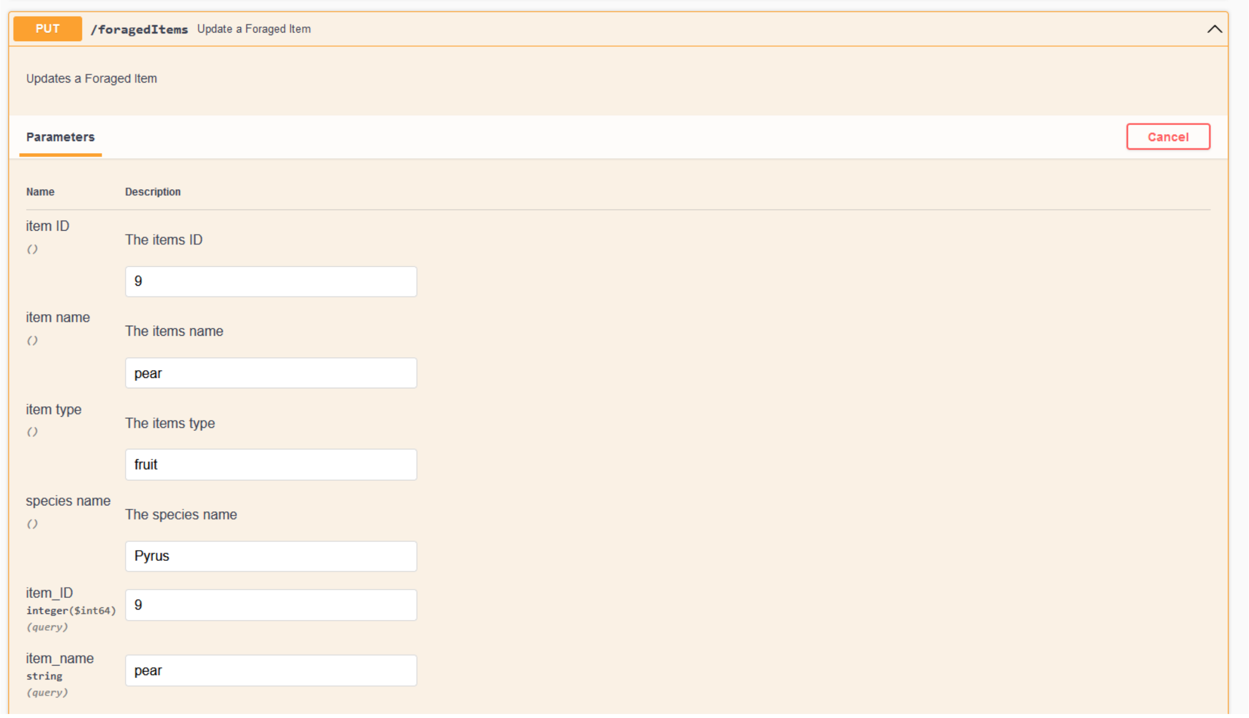
Code	Description	Links
201	Created Media type */* **Example Value** | Schema	*No links*

```
{
  "item_ID": 0,
  "item_name": "string",
  "item_type": "string",
  "species_name": "string"
}
```

Code	Description	Links
400	The input parameters are invalid Media type application/json	*No links*
404	Cannot create item with input criteria Media type application/json	*No links*
500	An unplanned error occured. Media type application/json	*No links*

PUT:

Final project submission



/foragedItems Update a Foraged Item

Updates a Foraged Item

Name Description

item ID
()
The items ID
9

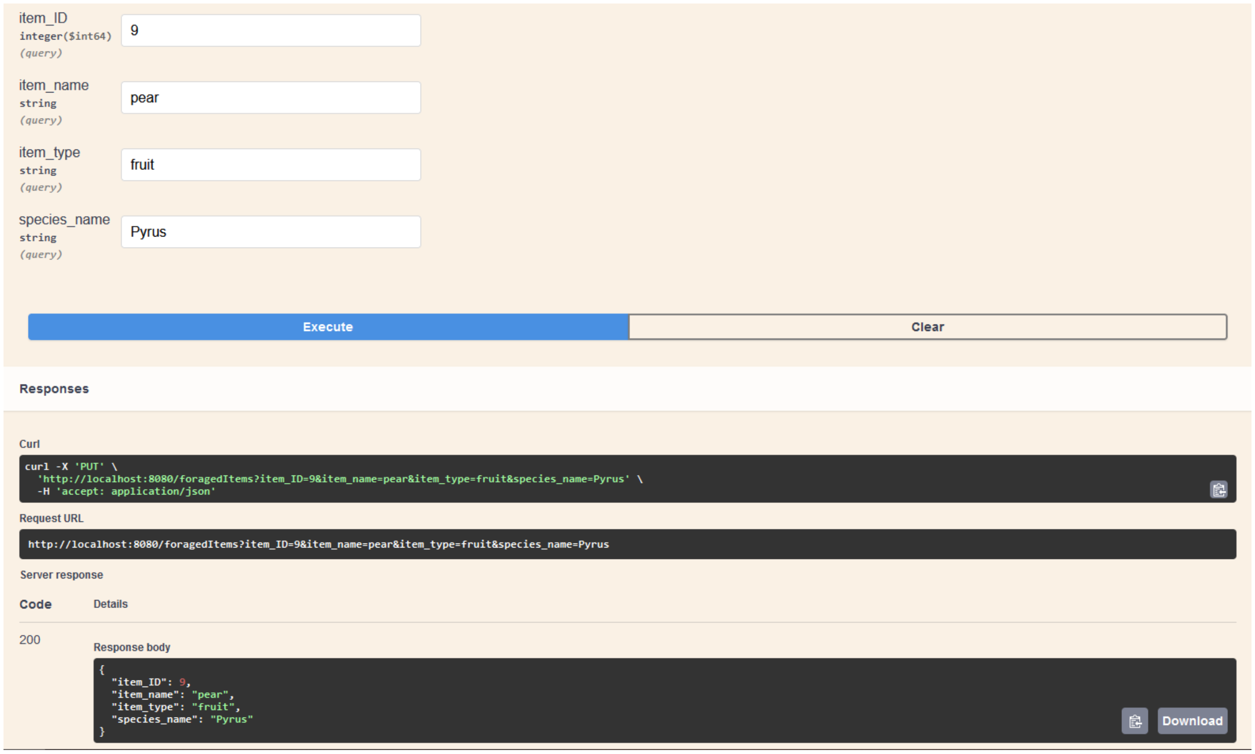
item name
()
The items name
pear

item type
()
The items type
fruit

species name
()
The species name
Pyrus

item_ID
integer($int64)
(query)
9

item_name
string
(query)
pear


item_ID
integer($int64)
(query)
9

item_name
string
(query)
pear

item_type
string
(query)
fruit

species_name
string
(query)
Pyrus

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:8080/foragedItems?item_ID=9&item_name=pear&item_type=fruit&species_name=Pyrus' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/foragedItems?item_ID=9&item_name=pear&item_type=fruit&species_name=Pyrus
```

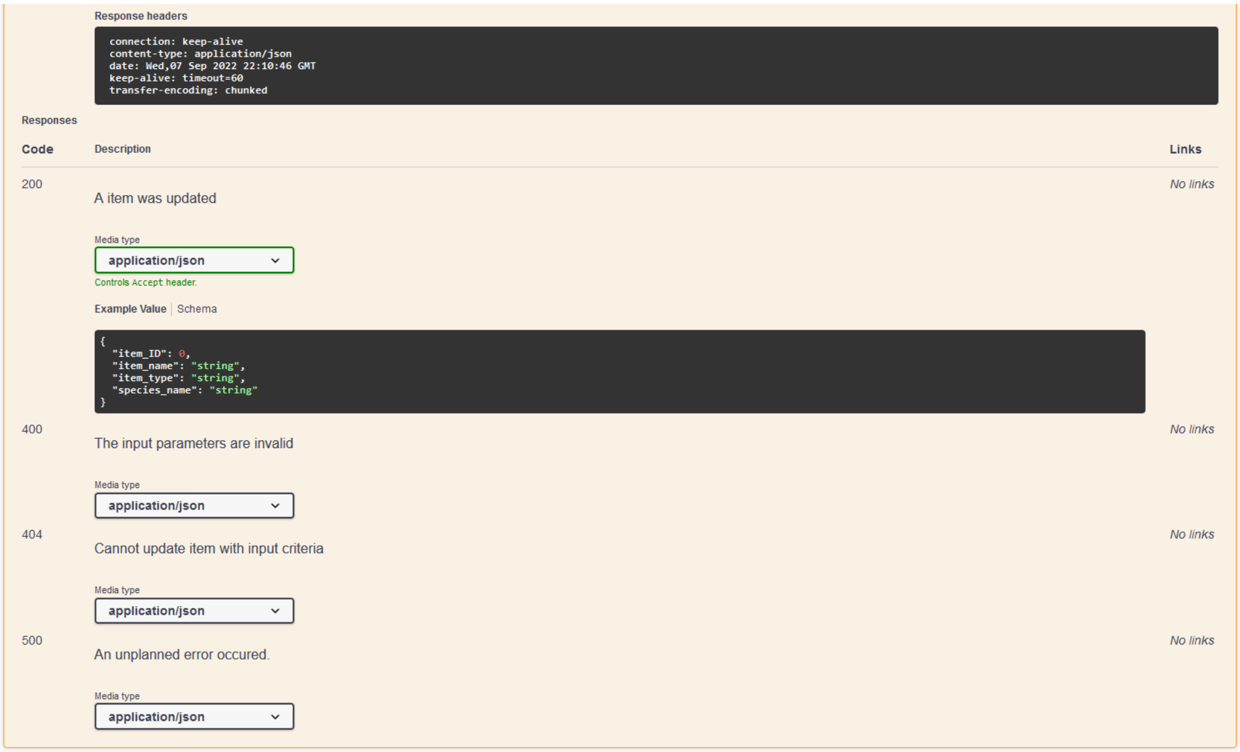
Server response

Code Details

200

Response body

```
{
  "item_ID": 9,
  "item_name": "pear",
  "item_type": "fruit",
  "species_name": "Pyrus"
}
```



Response headers

```
connection: keep-alive
content-type: application/json
date: Wed,07 Sep 2022 22:10:46 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Code Description Links

200 A item was updated No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "item_ID": 0,
  "item_name": "string",
  "item_type": "string",
  "species_name": "string"
}
```

400 The input parameters are invalid No links

Media type

application/json

404 Cannot update item with input criteria No links

Media type

application/json

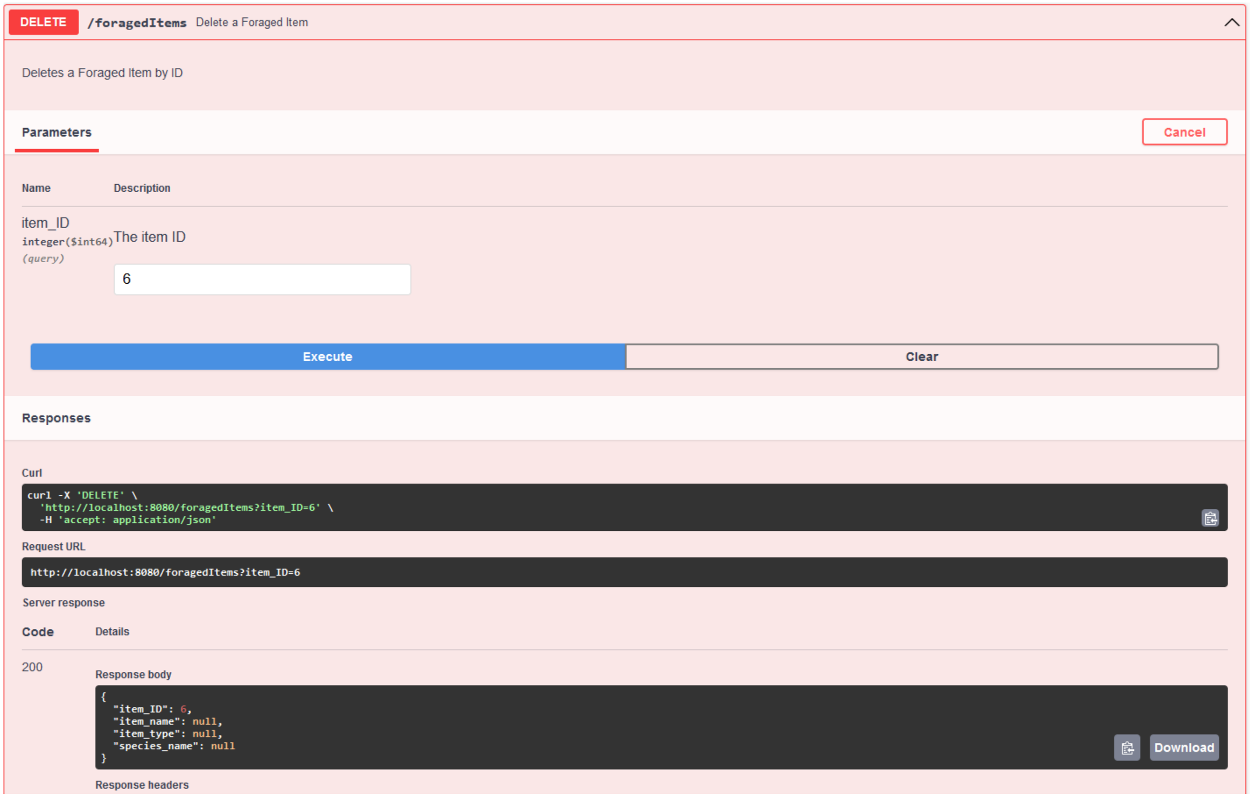
500 An unplanned error occured. No links

Media type

application/json

DELETE:

Final project submission



DELETE **/foragedItems** Delete a Foraged Item

Deletes a Foraged Item by ID

Parameters Cancel

Name	Description
item_ID integer($int64) *(query)*	The item ID 6

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:8080/foragedItems?item_ID=6' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/foragedItems?item_ID=6
```

Server response

Code Details

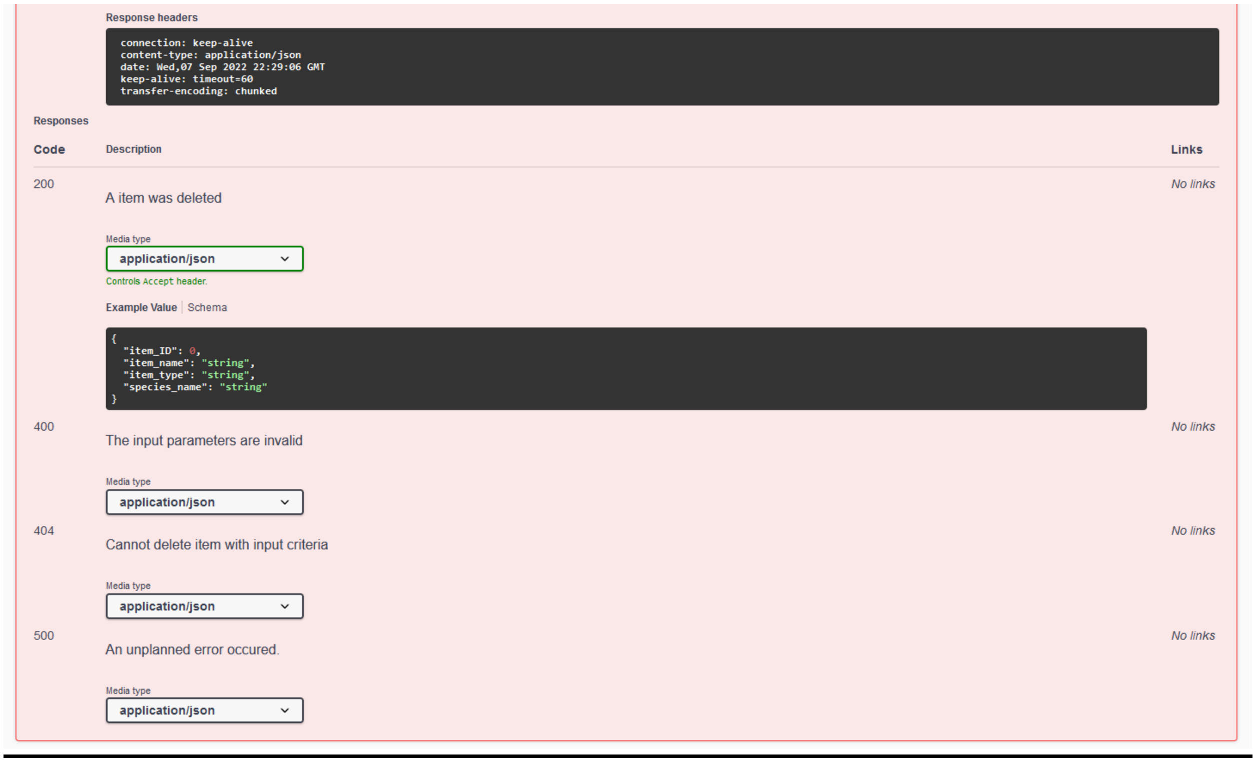
200

Response body

```
{
  "item_ID": 6,
  "item_name": null,
  "item_type": null,
  "species_name": null
}
```

Download

Response headers



Response headers

```
connection: keep-alive
content-type: application/json
date: Wed,07 Sep 2022 22:29:06 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Code Description **Links**

200

A item was deleted

No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "item_ID": 0,
  "item_name": "string",
  "item_type": "string",
  "species_name": "string"
}
```

400

The input parameters are invalid

No links

Media type

application/json

404

Cannot delete item with input criteria

No links

Media type

application/json

500

An unplanned error occured.

No links

Media type

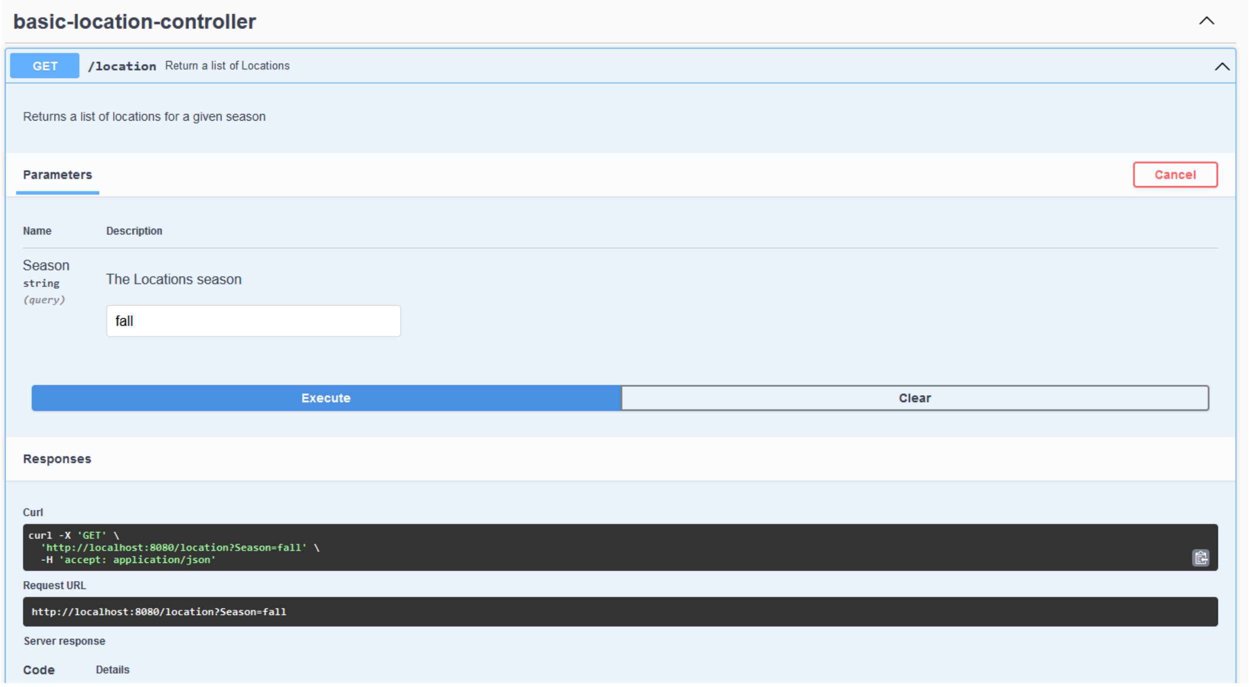
application/json

Final project submission

Basic-location-controller:

GET:

Final project submission



basic-location-controller

GET /location Return a list of Locations

Returns a list of locations for a given season

Parameters

Cancel

Name | Description

Season string (query) | The Locations season

fall

Execute | Clear

Responses

Curl

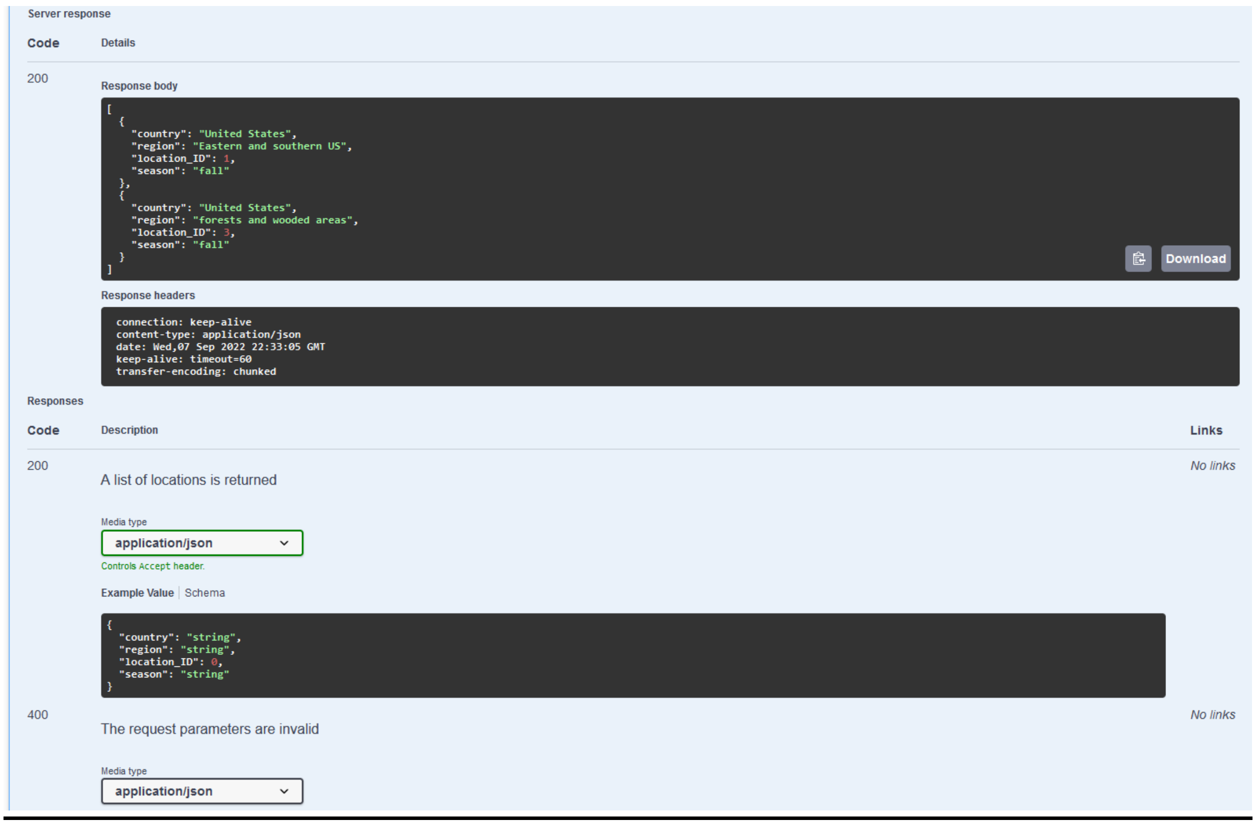
```
curl -X 'GET' \
  'http://localhost:8080/location?Season=fall' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/location?Season=fall
```

Server response

Code | Details



Server response

Code | Details

200

Response body

```
[
  {
    "country": "United States",
    "region": "Eastern and southern US",
    "location_ID": 1,
    "season": "fall"
  },
  {
    "country": "United States",
    "region": "forests and wooded areas",
    "location_ID": 3,
    "season": "fall"
  }
]
```

Download

Response headers

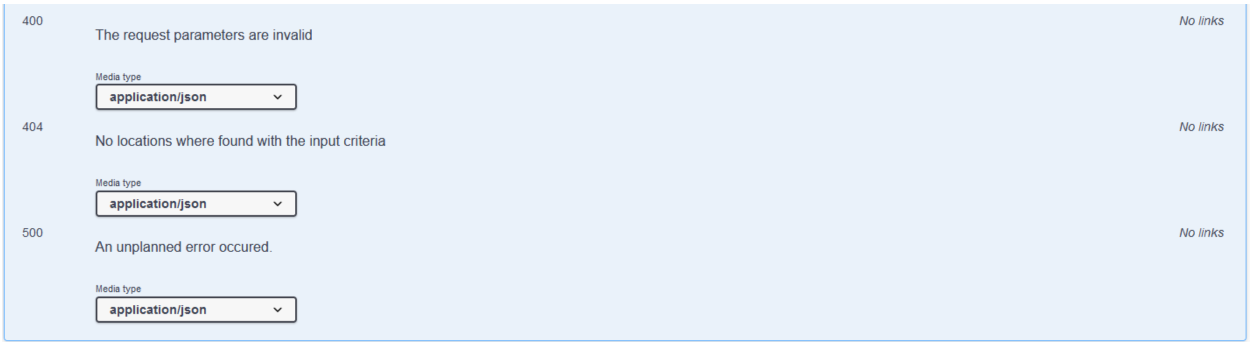
```
 connection: keep-alive
 content-type: application/json
 date: Wed,07 Sep 2022 22:33:05 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```

Responses

Code	Description	Links
200	A list of locations is returned Media type: application/json Controls Accept header. Example Value | Schema	*No links*
400	The request parameters are invalid Media type: application/json	*No links*

Example Value:

```
{
  "country": "string",
  "region": "string",
  "location_ID": 0,
  "season": "string"
}
```



400 The request parameters are invalid

No links

Media type: application/json

404 No locations where found with the input criteria

No links

Media type: application/json

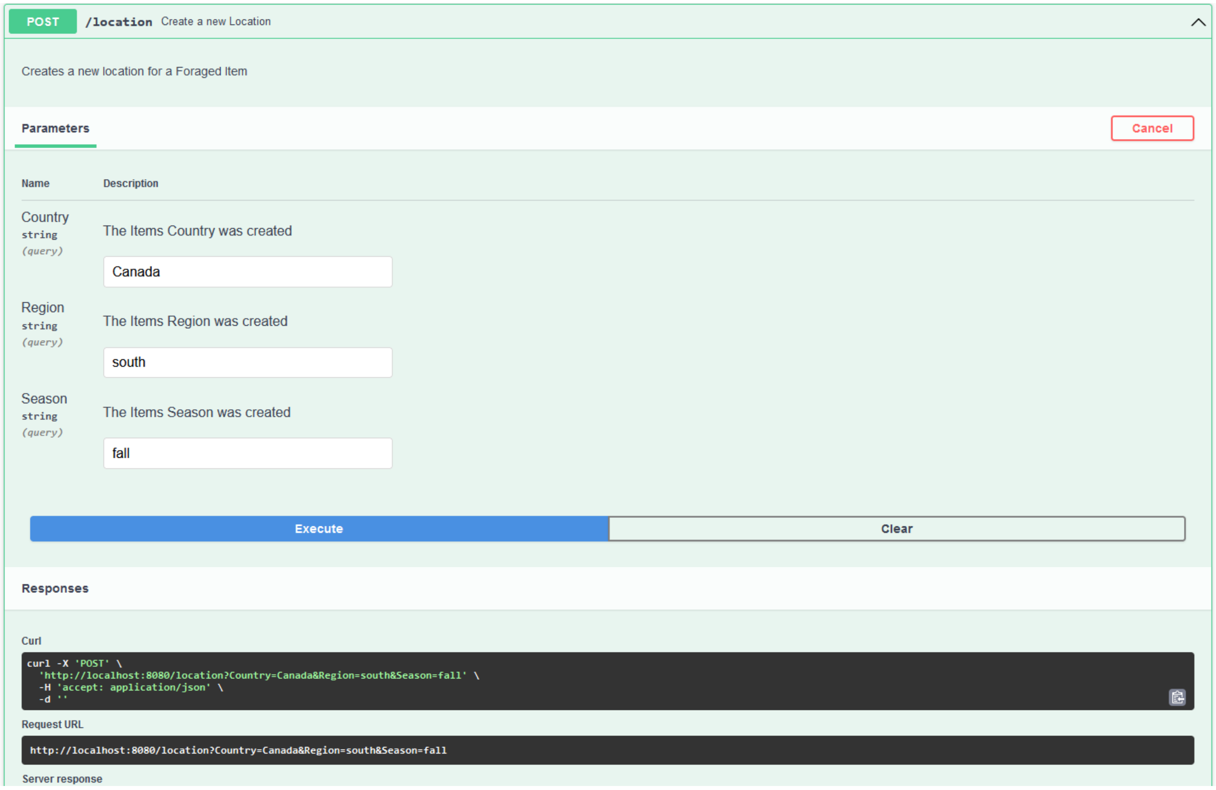
500 An unplanned error occured.

No links

Media type: application/json

POST:

Final project submission



POST /location Create a new Location

Creates a new location for a Foraged Item

Parameters

Cancel

Name	Description
Country string *(query)*	The Items Country was created Canada
Region string *(query)*	The Items Region was created south
Season string *(query)*	The Items Season was created fall

Execute | Clear

Responses

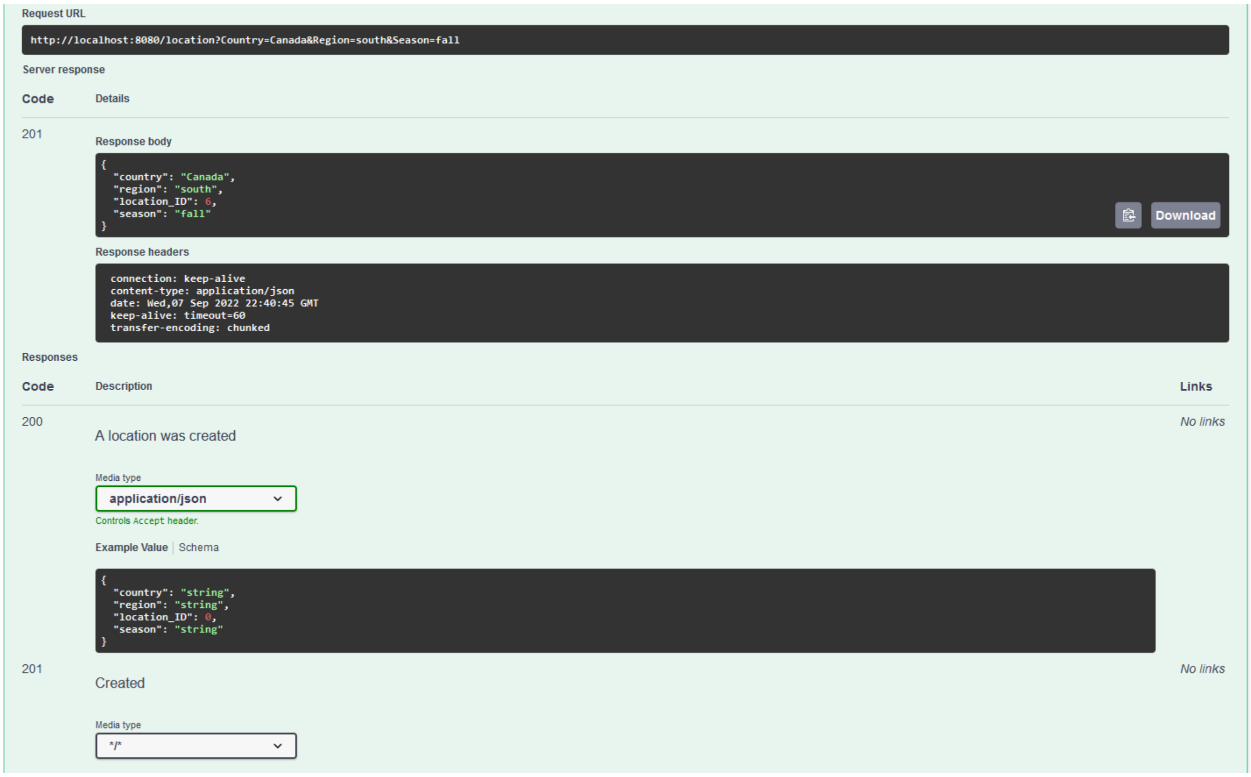
Curl

```
curl -X 'POST' \
  'http://localhost:8080/location?Country=Canada&Region=south&Season=fall' \
  -H 'accept: application/json' \
  -d ''
```

Request URL

```
http://localhost:8080/location?Country=Canada&Region=south&Season=fall
```

Server response



Request URL

```
http://localhost:8080/location?Country=Canada&Region=south&Season=fall
```

Server response

Code	Details
201	Response body

```
{
  "country": "Canada",
  "region": "south",
  "location_ID": 6,
  "season": "fall"
}
```

Download

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,07 Sep 2022 22:40:45 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```

Responses

Code	Description	Links
200	A location was created	*No links*

Media type: application/json

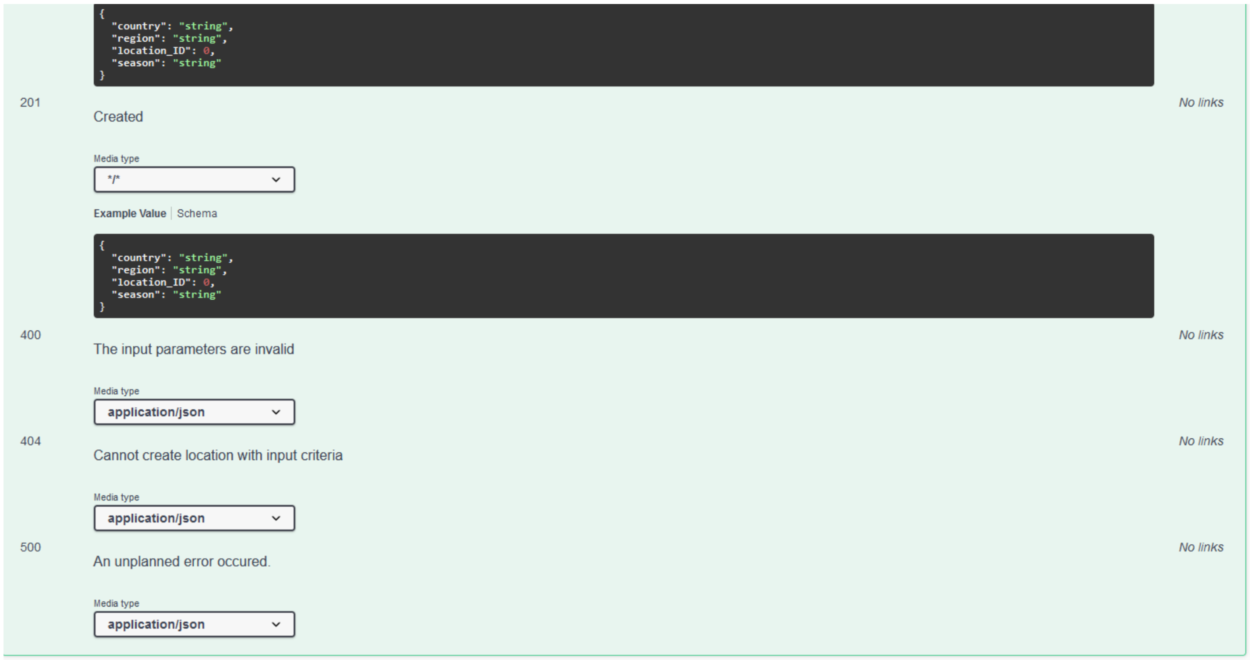
Controls Accept header.

Example Value | Schema

```
{
  "country": "string",
  "region": "string",
  "location_ID": 0,
  "season": "string"
}
```

Code	Description	Links
201	Created	*No links*

Media type: */*



```
{
  "country": "string",
  "region": "string",
  "location_ID": 0,
  "season": "string"
}
```

201

Created

Media type

/

Example Value | Schema

```
{
  "country": "string",
  "region": "string",
  "location_ID": 0,
  "season": "string"
}
```

400

The input parameters are invalid

Media type

application/json

404

Cannot create location with input criteria

Media type

application/json

500

An unplanned error occured.

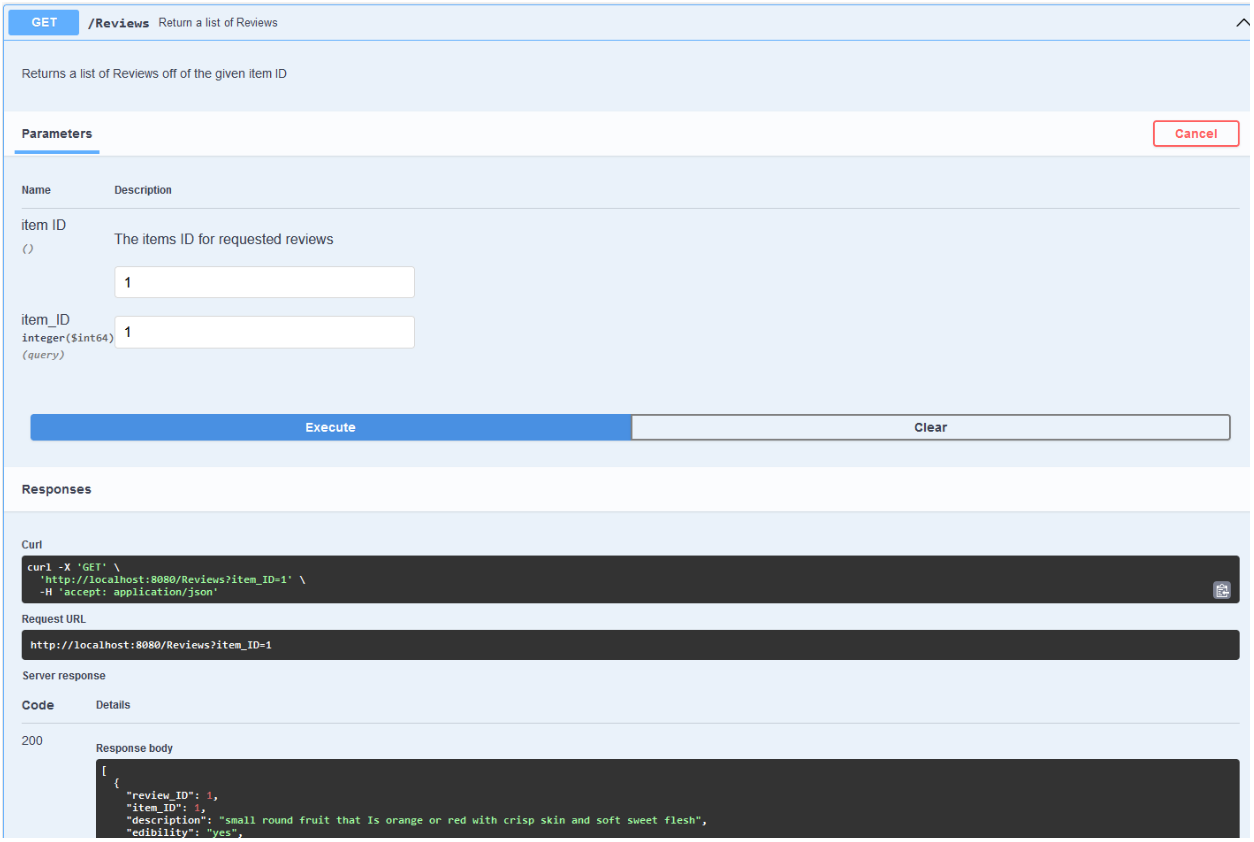
Media type

application/json

Basic-reviews-controller

GET:

Final project submission



Return a list of Reviews

Returns a list of Reviews off of the given item ID

Parameters

Name	Description
item ID ()	The items ID for requested reviews 1
item_ID integer($int64) (query)	1

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/Reviews?item_ID=1' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/Reviews?item_ID=1
```

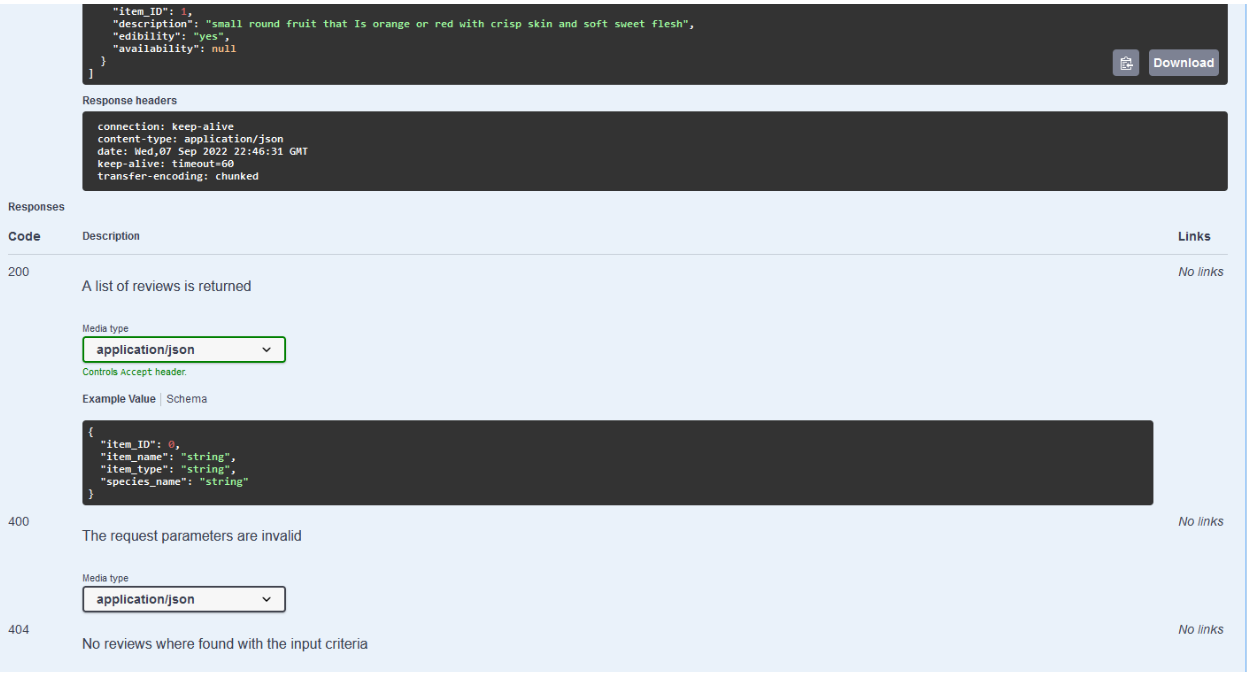
Server response

Code Details

200

Response body

```
[
  {
    "review_ID": 1,
    "item_ID": 1,
    "description": "small round fruit that Is orange or red with crisp skin and soft sweet flesh",
    "edibility": "yes",
```



```
    "item_ID": 1,
    "description": "small round fruit that Is orange or red with crisp skin and soft sweet flesh",
    "edibility": "yes",
    "availability": null
  }
]
```

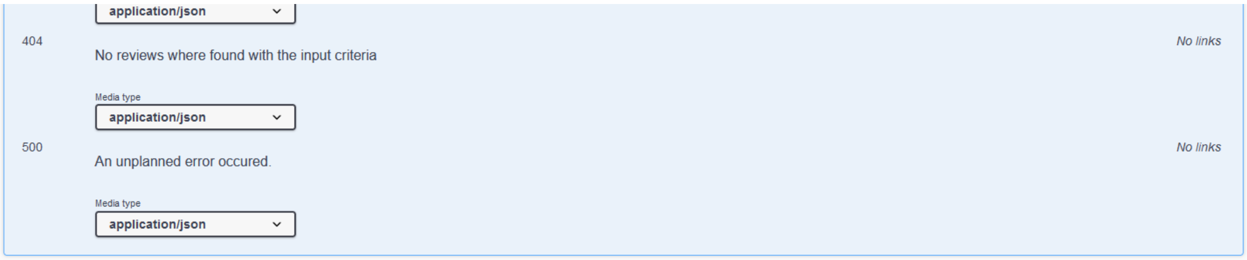
Response headers

```
connection: keep-alive
content-type: application/json
date: Wed,07 Sep 2022 22:46:31 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Code	Description	Links
200	A list of reviews is returned Media type: application/json Controls Accept header.	No links
400	The request parameters are invalid Media type: application/json	No links
404	No reviews where found with the input criteria	No links

```
{
  "item_ID": 0,
  "item_name": "string",
  "item_type": "string",
  "species_name": "string"
}
```



application/json

404

No reviews where found with the input criteria

Media type

application/json

No links

500

An unplanned error occured.

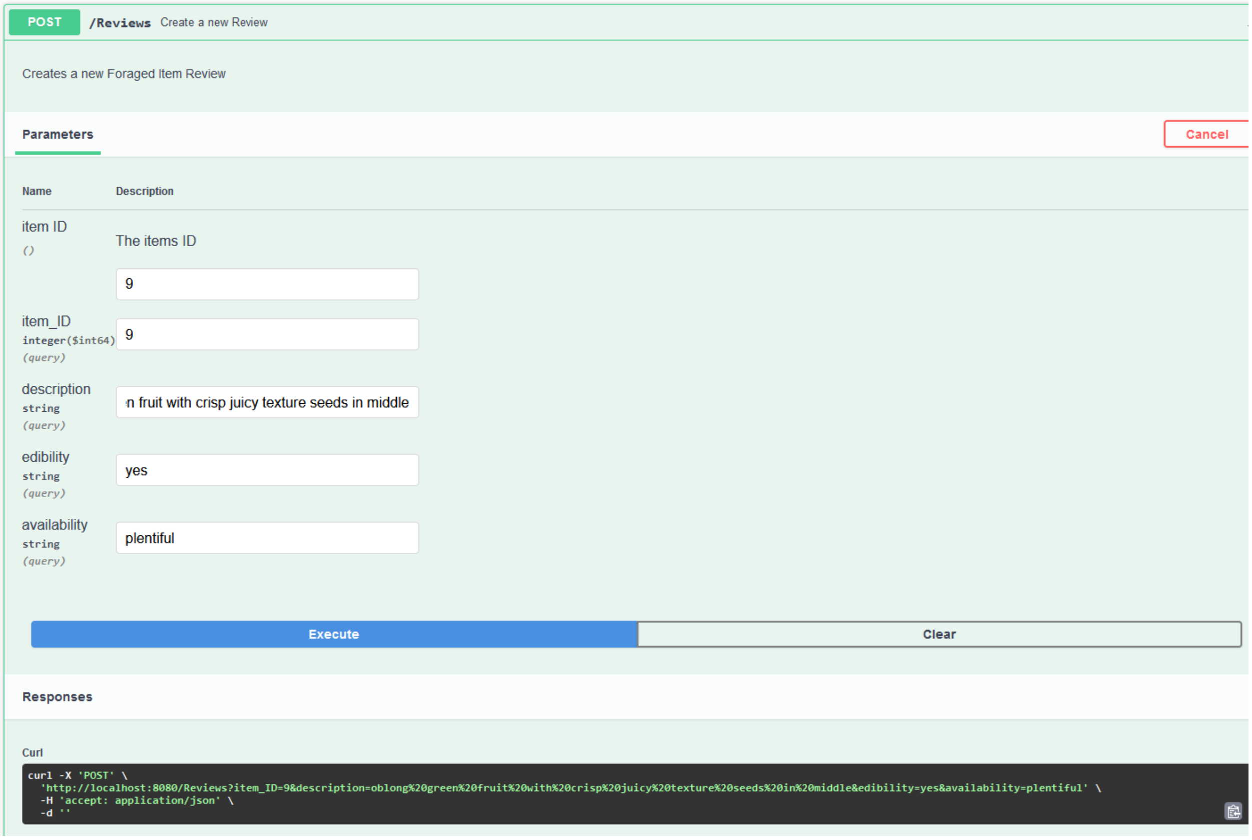
Media type

application/json

No links

POST:

Final project submission



POST /Reviews Create a new Review

Creates a new Foraged Item Review

Parameters

Cancel

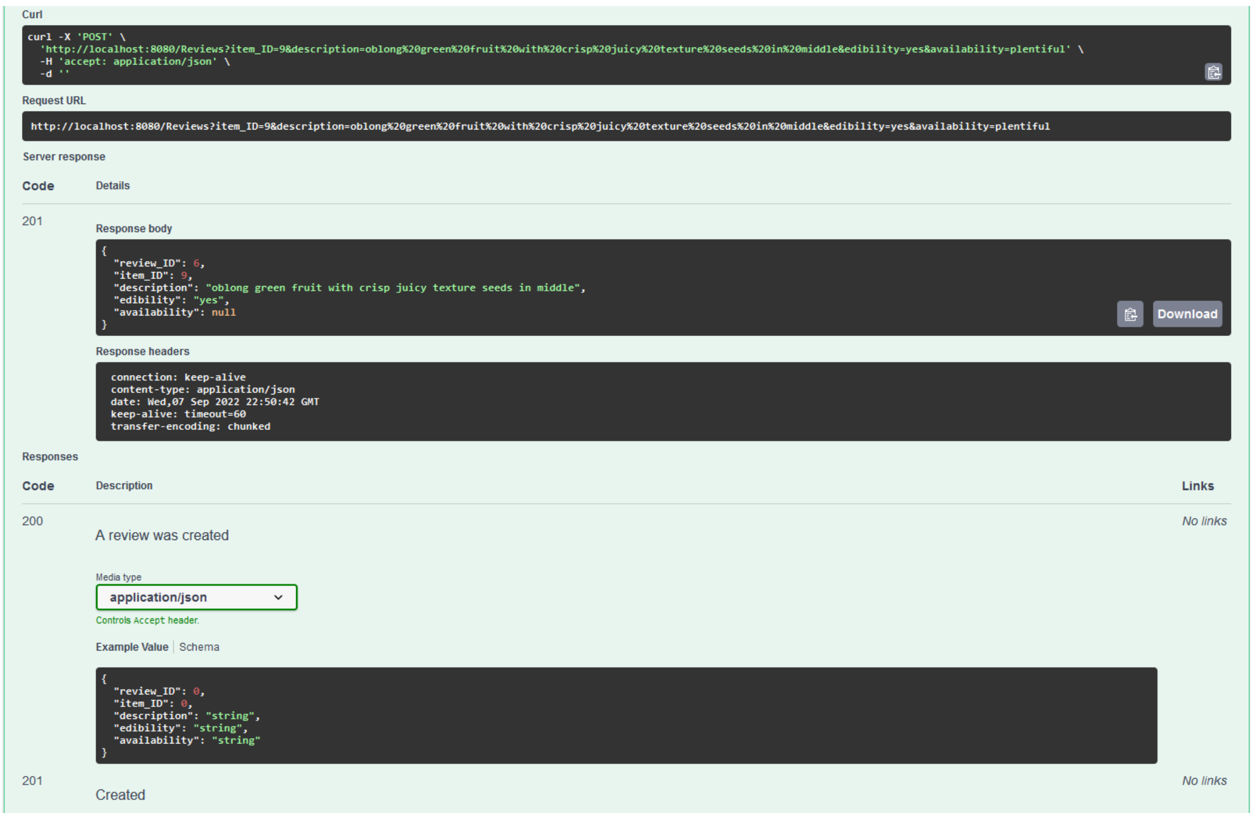
Name	Description
item ID ()	The items ID 9
item_ID integer($int64) (query)	9
description string (query)	n fruit with crisp juicy texture seeds in middle
edibility string (query)	yes
availability string (query)	plentiful

Execute | Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/Reviews?item_ID=9&description=oblong%20green%20fruit%20with%20crisp%20juicy%20texture%20seeds%20in%20middle&edibility=yes&availability=plentiful' \
  -H 'accept: application/json' \
  -d ''
```



Curl

```
curl -X 'POST' \
  'http://localhost:8080/Reviews?item_ID=9&description=oblong%20green%20fruit%20with%20crisp%20juicy%20texture%20seeds%20in%20middle&edibility=yes&availability=plentiful' \
  -H 'accept: application/json' \
  -d ''
```

Request URL

```
http://localhost:8080/Reviews?item_ID=9&description=oblong%20green%20fruit%20with%20crisp%20juicy%20texture%20seeds%20in%20middle&edibility=yes&availability=plentiful
```

Server response

Code	Details
201	Response body

```
{
  "review_ID": 6,
  "item_ID": 9,
  "description": "oblong green fruit with crisp juicy texture seeds in middle",
  "edibility": "yes",
  "availability": null
}
```

Download

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,07 Sep 2022 22:50:42 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```

Responses

Code	Description	Links
200	A review was created	*No links*

Media type

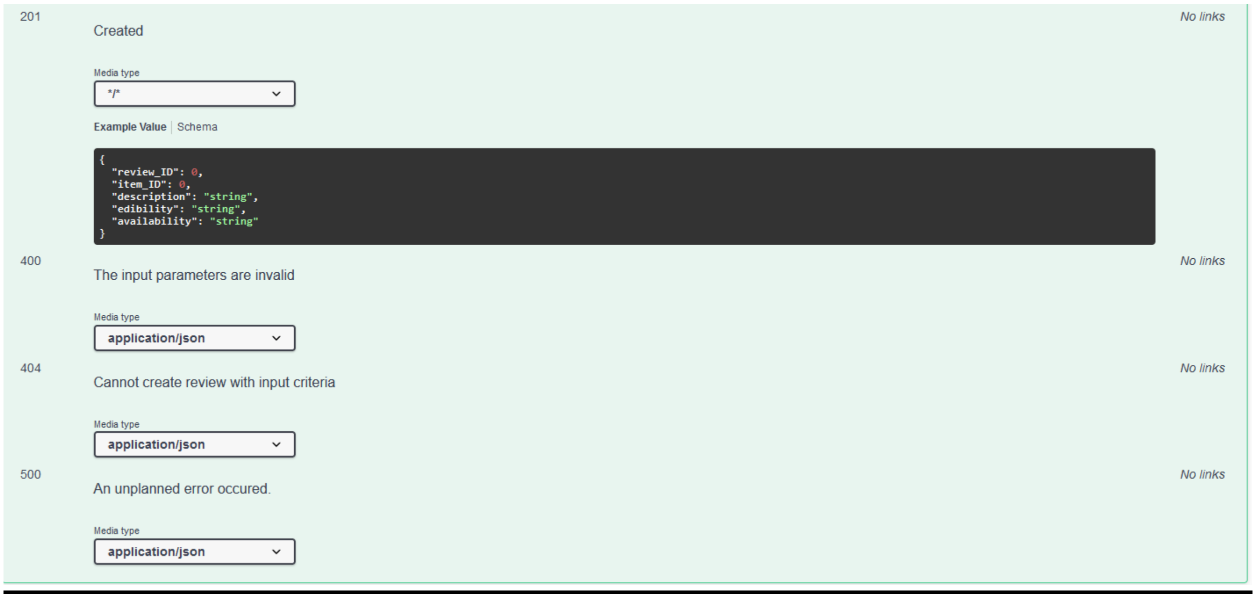
application/json

Controls Accept header.

Example Value | Schema

```
{
  "review_ID": 0,
  "item_ID": 0,
  "description": "string",
  "edibility": "string",
  "availability": "string"
}
```

Code	Description	Links
201	Created	*No links*



201

Created

No links

Media type

/

Example Value | Schema

```
{
  "review_ID": 0,
  "item_ID": 0,
  "description": "string",
  "edibility": "string",
  "availability": "string"
}
```

400

The input parameters are invalid

No links

Media type

application/json

404

Cannot create review with input criteria

No links

Media type

application/json

500

An unplanned error occured.

No links

Media type

application/json

API documentation: (will also be a copy attached in git)

The foraging-API I built helps ease the process of foraging for food. With the basic features rolled out in foraging-1.0 you can:

Current features for the foraging-API

- Search for foraged items.
- If item does not exist, you can upload it into the system.
- Can update past entries if information is not accurate.
- Deleting an entry is also a possibility with the items name.

- Can return a list of user uploaded reviews for a specified foraged item.
- If the review you were looking for doesn't exist you can upload your own.

- Currently gives you the ability to return a list or locations off of a requested season.
- Can upload your own location for foraging.

Final project submission

Future features for the Foraging-API

- In future updates I hope to implement the ability to attach images to foraged items.
- & have users be able to upload images to reviews.
- Have a more in-depth connection between foraged items and locations.
- The ability to update previous locations.
- Alter or delete previous reviews left.

Links to my github repo:

https://github.com/lwick95/PromineoTech-Final