

FINAL PROJECT - SPRING BOOT  
Raina Gulbrandson

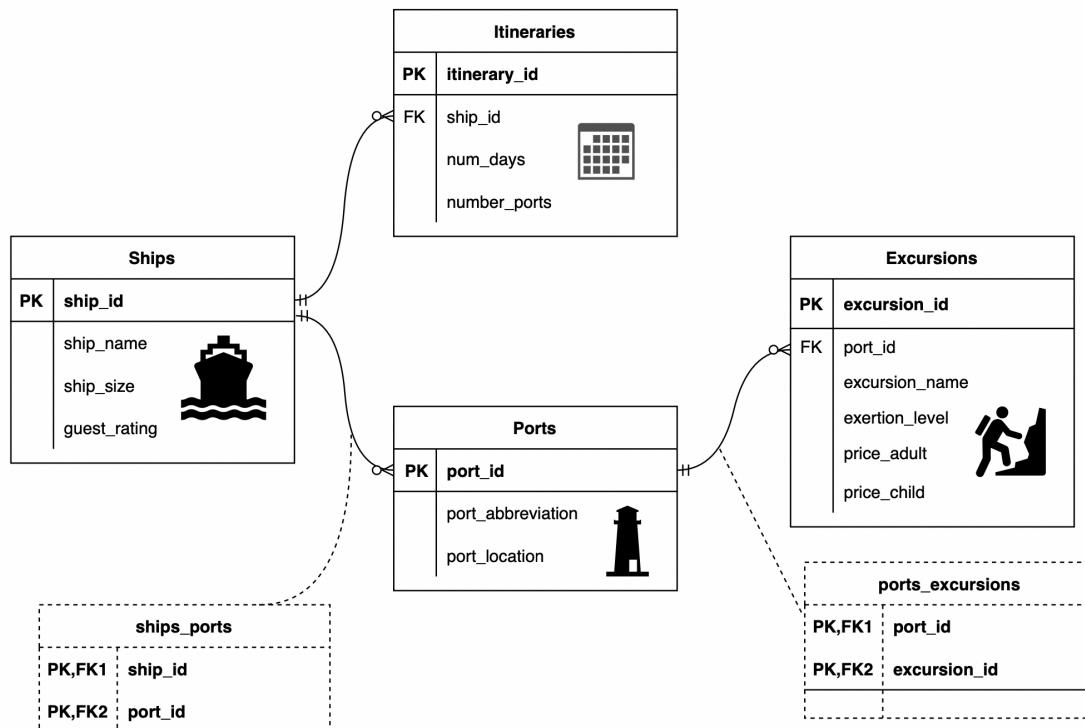
GitHub Repo URL:  
<https://github.com/rqlbrandson27/Spring-Boot-Cruise-Trips-Project.git>

Description:

This Web API allows a cruise line travel company to keep track of information about the specific ships in operation. Each cruise ship has its own attributes, such as name, size and overall guest rating. A particular ship may have more than one route it takes at different times of the year. These routes are called itineraries and each unique itinerary has a specific trip length and number of ports that are visited.

This API is also something a traveler can use to choose their perfect travel plans based on the ship features, trip length, and ports visited. Each port has its own options of activities that cruise goers can choose among to get the most out of their vacation whether that be challenging adventures, immersive experiences, hours of relaxation, or a mix of everything.

**LET'S GO ON A CRUISE TRIP!**



## ships-controller-impl

**GET** /Ships Returns a list of ships

**PUT** /Ships Updates ship details

**POST** /Ships Create a new Ship

**DELETE** /Ships Deletes a Ship

**GET** /Ships/{id} Gets a specific ship by Id

## ports-controller-impl

**GET** /Ports Returns a list of ports

**PUT** /Ports Updates Port details

**POST** /Ports Create a new Port

**DELETE** /Ports Deletes a Port

**GET** /Ports/{id} Gets a specific port by Id

## itineraries-controller-impl

**GET** /Itineraries Returns a list of itineraries

**PUT** /Itineraries Updates Itinerary details

**POST** /Itineraries Create a new Itinerary

**DELETE** /Itineraries Deletes an Itinerary

**GET** /Itineraries/{id} Gets a specific Itinerary by Id

## excursions-controller-impl

**GET** /Excursions Returns a list of Excursions

**PUT** /Excursions Updates Excursion details

**POST** /Excursions Create a new Excursion

**DELETE** /Excursions Deletes an Excursion

**GET** /Excursions/{id} Gets a specific Excursion by Id

## ships-controller-impl

**GET** /Ships Returns a list of ships

Returns a list of all ships

### Parameters

[Cancel](#)

No parameters

[Execute](#)

[Clear](#)

### Responses

Curl

```
curl -X 'GET' \
'http://localhost:8080/Ships' \
-H 'accept: application/json'
```



Request URL

```
http://localhost:8080/ships
```

Server response

[Code](#) [Details](#)

200

Response body

```
[  
  {  
    "shipId": 1,  
    "shipName": "Intchantment of the Seas",  
    "shipSize": "2500",  
    "guestRating": 4.9  
  },  
  {  
    "shipId": 2,  
    "shipName": "Double Paradise",  
    "shipSize": "1800",  
    "guestRating": 4.3  
  },  
]
```

Code	Details
Response body	
200	<pre>[{"shipId": 3, "shipName": "Long Voyager", "shipSize": "2000", "guestRating": 4.2}, {"shipId": 4, "shipName": "Costa Corta", "shipSize": "800", "guestRating": 4.2}, {"shipId": 5, "shipName": "Floating Oasis", "shipSize": "3000", "guestRating": 4.8}, {"shipId": 6, "shipName": "ByteTonic", "shipSize": "4000", "guestRating": 4.7}, {"shipId": 7}</pre>
Response headers	
200	<pre>connection: keep-alive content-type: application/json date: Thu, 15 Dec 2022 06:27:41 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>
Responses	
Code	Description
200	A list of Ships is returned
Media type	
application/json	
Controls Accept header.	

**GET** /ships/{id} Gets a specific ship by id

Gets the ship details using the ship id

**Parameters**

Name	Description
shipId <small>* required</small>	The shipId (i.e., '6')
(query)	<input type="text" value="6"/>

**Execute** **Clear**

**Responses**

**Curl**

```
curl -X 'GET' \
  'http://localhost:8080/Ships/{id}?shipId=6' \
  -H 'accept: application/json'
```

**Request URL**

```
http://localhost:8080/Ships/{id}?shipId=6
```

**Server response**

Code	Details
200	Response body
200	<pre>[{"shipId": 6, "shipName": "ByteTonic", "shipSize": "4000", "guestRating": 4.7}]</pre>

**POST /Ships** Create a new Ship

Enter the name, size and guest rating of the ship

**Parameters**

Name	Description
<b>shipName</b> * required string (query)	The name of the new Ship (i.e., 'Lady of the Sea')  The StringRay
<b>shipSize</b> * required string (query)	The size/occupancy of the Ship (i.e. '2,500')  2,000
<b>guestRating</b> * required number (query)	The star guest rating of the ship (i.e. '4.7')  2.7
<input type="checkbox"/> Send empty value	

**Execute**

**Responses**

**Curl**

```
curl -X 'POST' \
'http://localhost:8080/Ships?shipName=The%20StringRay&shipSize=2%2C000&guestRating=2.7' \
-H 'accept: application/json' \
-d ''
```

**Request URL**

<http://localhost:8080/Ships?shipName=The%20StringRay&shipSize=2%2C000&guestRating=2.7>

**Server response**

Code	Details
201	<b>Response body</b> <pre>{   "shipId": null,   "shipName": "The StringRay",   "shipSize": "2,000",   "guestRating": 2.7 }</pre> <p><a href="#">Copy</a> <a href="#">Download</a></p> <b>Response headers</b> <pre>connection: keep-alive content-type: application/json date: Thu,15 Dec 2022 06:33:57 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

**DELETE /Ships** Deletes a Ship

Deletes a Ship and all of its attributes

**Parameters**

Name	Description
<b>shipId</b> * required integer(\$int64) (query)	The ID number of the Shipname of the new Ship (i.e., '8') <input type="text" value="7"/>
<b>shipName</b> * required string (query)	The name of the Ship (i.e. 'Lady of the Seas') <input type="text" value="Charming Sea Princess"/>

**Execute**

**Curl**

```
curl -X 'DELETE' \
'http://localhost:8080/Ships?shipId=7&shipName=Charming%20Sea%20Princess' \
-H 'accept: application/json'
```

**Request URL**

```
http://localhost:8080/ships?shipId=7&shipName=Charming%20Sea%20Princess
```

**Server response**

Code	Details
200	<b>Response body</b> <input type="text" value="null"/> <b>Response headers</b> <pre>connection: keep-alive content-type: application/json date: Thu, 15 Dec 2022 06:45:31 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

**Responses**

Code	Description	Links
200	A Ship has been deleted	No links

**Media type**

Controls Accept header.

Example Value Schema

```
{ "shipId": 0 }
```

**PUT /Ships** Updates ship details

Updates the details of the name, size, and/or guest rating of a ship

**Parameters**

Name	Description
<b>shipId</b> * required integer(\$int64) (query)	The number ID of the Ship (i.e. '8') <input type="text" value="8"/>
<b>ShipName</b> * required string (query)	The name of the Ship to be updated (i.e., 'Lady of the Sea') <input type="text" value="True Beauty"/>
<b>shipSize</b> * required string (query)	The size/occupancy of the Ship (i.e. '2,500') <input type="text" value="2,400"/>
<b>guestRating</b> * required number (query)	The star guest rating of the ship (i.e. '4.7') <input type="text" value="0.0"/>

Send empty value

Send empty value

<b>newShipName</b> * required string (query)	The name of the new Ship (i.e., 'Lady of the Ocean') <input type="text" value="True Beauty"/>
<b>newShipSize</b> * required string (query)	The size/occupancy of the Ship (i.e. '3,000') <input type="text" value="2,400"/>
<b>newGuestRating</b> * required number (query)	The star guest rating of the ship (i.e. '4.8') <input type="text" value="4.9"/>

Send empty value

**Execute** **Clear**

**Responses**

---

Code	Details	Links
200	<p>Response body</p> <pre>null</pre> <p><a href="#">Copy</a> <a href="#">Download</a></p> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Thu,15 Dec 2022 06:49:34 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>	
<b>Responses</b>		
Code	Description	Links
200	The ship details have been updated	No links
<p>Media type</p> <p><b>application/json</b> <a href="#">▼</a></p> <p>Controls Accept header.</p> <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>{   "shipId": 0,   "ShipName": "string",   "shipSize": "string",   "guestRating": 0 }</pre>		

## Response Status Code List (for all entities)

Responses		Links
Code	Description	
200	A list of Excursions with details is returned	No links
	<p>Media type</p> <div><p>application/json</p><p>Controls Accept header.</p></div> <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>{ "excursionId": 0, "excursionName": "string", "exertionLevel": "string", "adultPrice": 0, "childPrice": 0 }</pre>	
400	The request parameters are invalid	No links
	<p>Media type</p> <div><p>application/json</p></div>	
404	No Excursions were found with the input criteria	No links
	<p>Media type</p> <div><p>application/json</p></div>	
500	An unplanned error occurred.	No links
	<p>Media type</p> <div><p>application/json</p></div>	