

Freezer Rest API

Implementation

I have implemented the solution using Maven, Java 8, Hibernate and Restlet with an in-memory H2 database (and JUnit for tests).

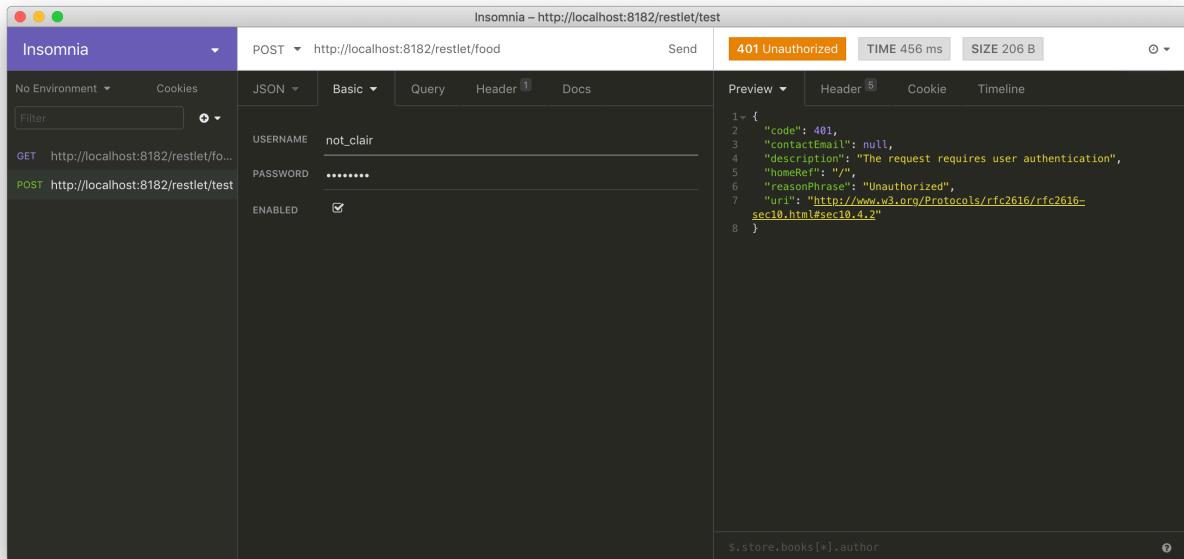
Restlet was chosen as it is a lightweight framework that can be easily added to a small project.

I'm quite satisfied with my implementation but I can see various possible improvements, starting with using Spring to simplify dependency injection as well as handling Hibernate transactions in a more sensible manner with `@Transactional` annotation (currently I get around the issue by using Java 8 Functional interface, but it isn't the prettiest code).

Ideally the authentication process would also be based on OAuth rather than simple username + secret combination which is much less secure especially over HTTP. Authentication details should also not be hardcoded, especially in a non-hashed, unsalted manner.

Another thing I would improve is separating the save and update operations so it's more clear to the user whether they are adding or simply updating items.

Test cases



The screenshot shows the Insomnia REST client interface. The top bar displays "Insomnia - http://localhost:8182/restlet/test". The status bar indicates a 401 Unauthorized response, a time of 456 ms, and a size of 206 B. The main area shows a POST request to "http://localhost:8182/restlet/test". The request body is set to JSON and contains the following data:

```
USERNAME: not_clair  
PASSWORD: .....  
ENABLED: 
```

The "Header" tab is selected, showing the response header:

```
HTTP/1.1 401 Unauthorized  
Content-Type: application/json; charset=UTF-8  
Date: Mon, 10 Jul 2017 11:44:16 GMT  
Server: Apache-Coyote/1.1  
WWW-Authenticate: Basic realm="RESTLET TEST"  
X-Content-Type-Options: nosniff  
X-Frame-Options: SAMEORIGIN  
X-XSS-Protection: 1; mode=block
```

The "Preview" tab shows the JSON response body:

```
{  
  "code": 401,  
  "contactEmail": null,  
  "description": "The request requires user authentication",  
  "homeRef": "/",  
  "reasonPhrase": "Unauthorized",  
  "uri": "http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.2"  
}
```

Failed authentication (test username is “clair” and password is “password”)

Insomnia

POST http://localhost:8182/restlet/food

Send 200 OK TIME 331 ms SIZE 1 B

No Environment Cookies

JSON Basic Query Header Docs

```
1 {  
2   "name": "onion",  
3   "type": "VEGETABLES",  
4   "amount": 4  
5 }
```

Preview Header 5 Cookie Timeline

1 1

GET http://localhost:8182/restlet/fo...
POST http://localhost:8182/restlet/test

Beautify JSON

\$.store.books[*].author

This screenshot shows the Insomnia REST client interface. A POST request is made to the URL `http://localhost:8182/restlet/food`. The response status is 200 OK, with a response time of 331 ms and a size of 1 B. The response body is a JSON object with one item: `{ "name": "onion", "type": "VEGETABLES", "amount": 4 }`. The 'Preview' tab shows the raw response, while the 'Header' tab shows the response headers.

Save an item

Insomnia

POST http://localhost:8182/restlet/food

Send 200 OK TIME 82.8 ms SIZE 1 B

No Environment Cookies

JSON Basic Query Header Docs

```
1 {  
2   "name": "beef",  
3   "type": "MEAT",  
4   "amount": 1  
5 }
```

Preview Header 5 Cookie Timeline

1 2

GET http://localhost:8182/restlet/fo...
POST http://localhost:8182/restlet/test

Beautify JSON

\$.store.books[*].author

This screenshot shows the Insomnia REST client interface. A POST request is made to the URL `http://localhost:8182/restlet/food`. The response status is 200 OK, with a response time of 82.8 ms and a size of 1 B. The response body is a JSON object with one item: `{ "name": "beef", "type": "MEAT", "amount": 1 }`. The 'Preview' tab shows the raw response, while the 'Header' tab shows the response headers.

Insomnia

POST http://localhost:8182/restlet/food

Send 200 OK TIME 12.4 ms SIZE 1 B

No Environment Cookies

JSON Basic Query Header Docs

```
1 {  
2   "name": "cod",  
3   "type": "FISH",  
4   "amount": 2  
5 }
```

Preview Header 5 Cookie Timeline

1 3

GET http://localhost:8182/restlet/fo...
POST http://localhost:8182/restlet/test

Beautify JSON

\$.store.books[*].author

This screenshot shows the Insomnia REST client interface. A POST request is made to the URL `http://localhost:8182/restlet/food`. The response status is 200 OK, with a response time of 12.4 ms and a size of 1 B. The response body is a JSON object with one item: `{ "name": "cod", "type": "FISH", "amount": 2 }`. The 'Preview' tab shows the raw response, while the 'Header' tab shows the response headers.

Save other items

Insomnia – http://localhost:8182/restlet/food

Insomnia	GET ▾ http://localhost:8182/restlet/food/search	Send	200 OK	TIME 12.9 ms	SIZE 226 B	⋮
No Environment ▾ Cookies	JSON ▾ Basic ▾ Query Header 1 Docs	Preview ▾	Header 5	Cookie	Timeline	?
Filter ⌂	1 = { 2 "dateAdded": "2019-02-03" 3 }	1 = [2 { 3 "id": 1, 4 "name": "onion", 5 "type": "VEGETABLES", 6 "dateAdded": "2019-02-03", 7 "amount": 4 8 }, 9 { 10 "id": 2, 11 "name": "beef", 12 "type": "MEAT", 13 "dateAdded": "2019-02-03", 14 "amount": 1 15 }, 16 { 17 "id": 3, 18 "name": "cod", 19 "type": "FISH", 20 "dateAdded": "2019-02-03", 21 "amount": 2 22 } 23]	Header 5	Cookie	Timeline	?
POST http://localhost:8182/restlet/test	Beautify JSON	\$.store.books[*].author	?	?	?	?

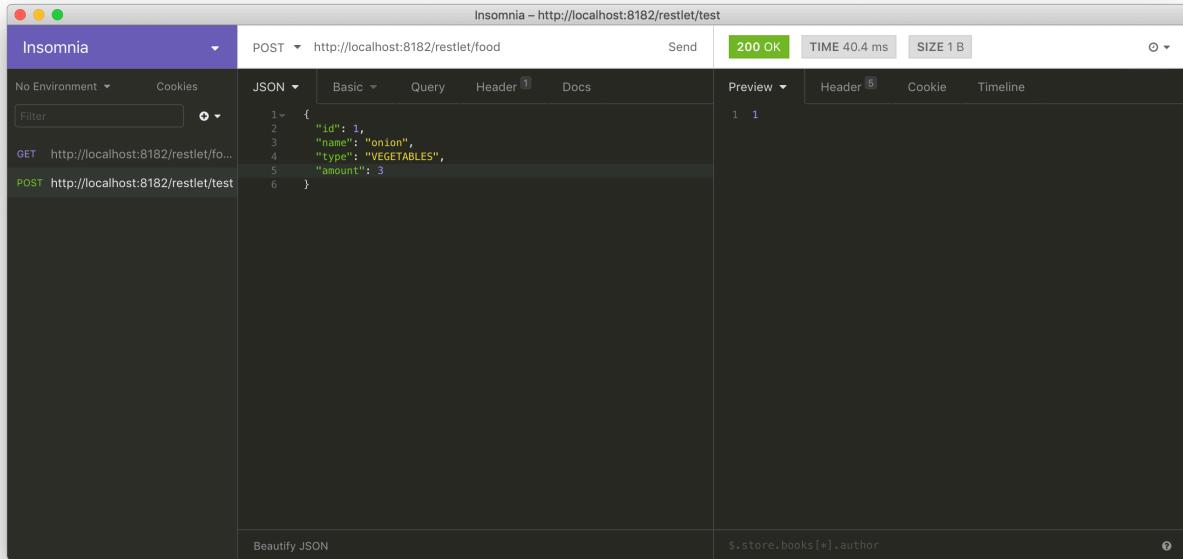
Insomnia – http://localhost:8182/restlet/food

Insomnia	GET ▾ http://localhost:8182/restlet/food/search	Send	200 OK	TIME 27.2 ms	SIZE 81 B	⋮
No Environment ▾ Cookies	JSON ▾ Basic ▾ Query Header 1 Docs	Preview ▾	Header 5	Cookie	Timeline	?
Filter ⌂	1 = { 2 "name": "onion" 3 }	1 = [2 { 3 "id": 1, 4 "name": "onion", 5 "type": "VEGETABLES", 6 "dateAdded": "2019-02-03", 7 "amount": 4 8 } 9]	Header 5	Cookie	Timeline	?
POST http://localhost:8182/restlet/test	Beautify JSON	\$.store.books[*].author	?	?	?	?

Insomnia – http://localhost:8182/restlet/food

Insomnia	GET ▾ http://localhost:8182/restlet/food/search	Send	200 OK	TIME 119 ms	SIZE 74 B	⋮
No Environment ▾ Cookies	JSON ▾ Basic ▾ Query Header 1 Docs	Preview ▾	Header 5	Cookie	Timeline	?
Filter ⌂	1 = { 2 "dateAdded": "2019-02-03", 3 "type": "MEAT" 4 }	1 = [2 { 3 "id": 2, 4 "name": "beef", 5 "type": "MEAT", 6 "dateAdded": "2019-02-03", 7 "amount": 1 8 } 9]	Header 5	Cookie	Timeline	?
POST http://localhost:8182/restlet/test	Beautify JSON	\$.store.books[*].author	?	?	?	?

Retrieve items based on one or multiple parameters



Insomnia – http://localhost:8182/restlet/test

POST http://localhost:8182/restlet/food

Send 200 OK TIME 40.4 ms SIZE 1 B

No Environment Cookies

Filter

Header 1 Docs

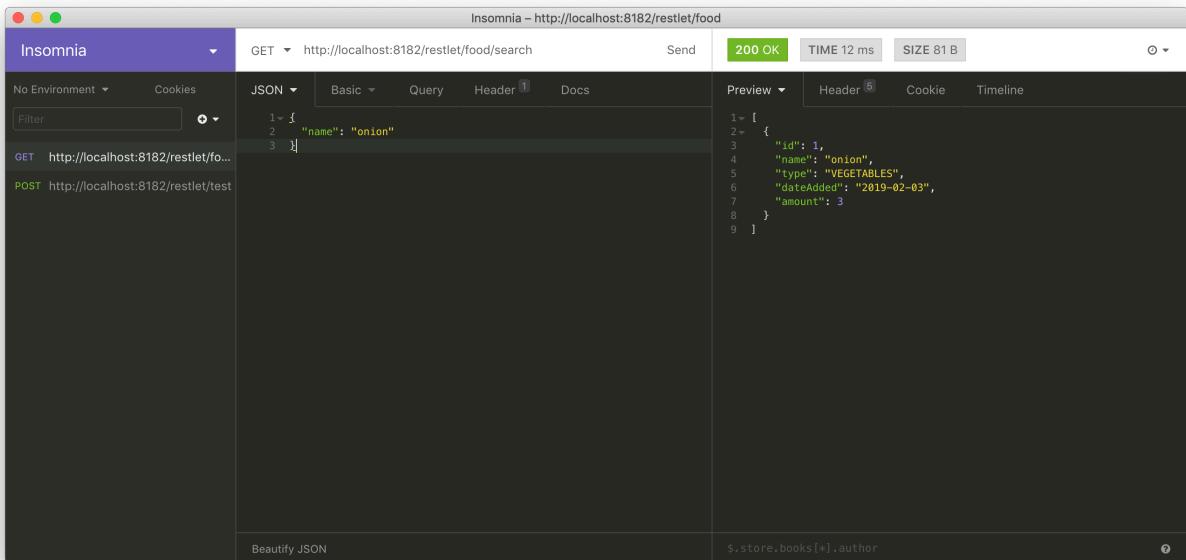
1 {
2 "id": 1,
3 "name": "onion",
4 "type": "VEGETABLES",
5 "amount": 3
6 }

Preview Header 5 Cookie Timeline

1 1

Beautify JSON \$,store.books[*].author

Update an item (changing amount from 4 to 3).
This uses the same interface as saving an item.



Insomnia – http://localhost:8182/restlet/food

GET http://localhost:8182/restlet/food/search

Send 200 OK TIME 12 ms SIZE 81 B

No Environment Cookies

Filter

Header 5 Docs

1 {
2 "name": "onion"
3 }

Preview Header 5 Cookie Timeline

1 [
2 {
3 "id": 1,
4 "name": "onion",
5 "type": "VEGETABLES",
6 "dateAdded": "2019-02-03",
7 "amount": 3
8 }
9]

Beautify JSON \$,store.books[*].author

Retrieve the updated item

Insomnia – http://localhost:8182/restlet/test

GET ▾ http://localhost:8182/restlet/food

Send **200 OK** TIME 65.2 ms SIZE 79 B

No Environment ▾ Cookies

JSON ▾ Basic ▾ Query Header 1 Docs

Filter 1 1

1 GET http://localhost:8182/restlet/fo...
2 GET http://localhost:8182/restlet/test

Preview ▾ Header 5 Cookie Timeline

```
1 = {  
2   "id": 1,  
3   "name": "onion",  
4   "type": "VEGETABLES",  
5   "dateAdded": "2019-02-03",  
6   "amount": 3  
7 }
```

Beautify JSON

s.store.books[*].author

Insomnia – http://localhost:8182/restlet/test

GET ▾ http://localhost:8182/restlet/food

Send **200 OK** TIME 23.9 ms SIZE 72 B

No Environment ▾ Cookies

JSON ▾ Basic ▾ Query Header 1 Docs

Filter 1 z|

1 GET http://localhost:8182/restlet/fo...
2 GET http://localhost:8182/restlet/test

Preview ▾ Header 5 Cookie Timeline

```
1 = {  
2   "id": 2,  
3   "name": "beet",  
4   "type": "MEAT",  
5   "dateAdded": "2019-02-03",  
6   "amount": 1  
7 }
```

Beautify JSON

s.store.books[*].author

Retrieve an item by id