

Test App Random Beer Alejandro Flores :

URL = <http://portalpega.cl/alexRandomBeer/api/beer/>

APP = PostMan

First Insert :

POST ▼ <http://portalpega.cl/alexRandomBeer/api/beer/> Params Send ▼

Authorization Headers (1) **Body** Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "name" : "guinness",  
3   "description" : "very cool beer",  
4   "abv" : "4.2",  
5   "brewery" : "Temple Bar"  
6 }
```

Body Cookies Headers (7) Test Results Status: 200 OK

Pretty Raw Preview

```
{ "id": "5af379e4d30b7fc8e7d68f15", "name": "guinness", "description": "very cool beer", "abv": "4.2", "brewery": "Temple Bar" }
```

Insert nº 2 =

POST ▼ <http://portalpega.cl/alexRandomBeer/api/beer/> Params Send ▼

Authorization Headers (1) **Body** Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "name" : "Leffe",  
3   "description" : "the best beer ever",  
4   "abv" : "5",  
5   "brewery" : "coffee Bruxelles"  
6 }
```

Body Cookies Headers (7) Test Results Status: 200 OK

Pretty Raw Preview

```
{ "id": "5af37b18d30b7fc8e7d68f16", "name": "Leffe", "description": "the best beer ever", "abv": "5", "brewery": "coffee Bruxelles" }
```

Insert Nº 3 =

POST

http://portalpega.cl/alexRandomBeer/api/beer/

Params

Send

Authorization

Headers (1)

Body

Pre-request Script

Tests

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1

{

2

"name" : "chouffe",

3

"description" : " strong blond 'gnome beer', first brewed in 1982",

4

"abv" : "5",

5

"brewery" : "coffee chouffe"

6

}

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Pretty

Raw

Preview

{

"id":

"5af37bc5d30b7fc8e7d68f17",

"name":

"chouffe",

"description":

" strong blond 'gnome beer', first brewed in

1982",

"abv":

"5",

"brewery":

"coffee chouffe"}

}

GET All =

GET

http://portalpega.cl/alexRandomBeer/api/beer/

Params

Send

Authorization

Headers (1)

Body

Pre-request Script

Tests

Type

No Auth

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Pretty

Raw

Preview

[

{

"id":

"5af379e4d30b7fc8e7d68f15",

"name":

"guinness",

"description":

"very cool beer",

"abv":

"4.2",

"brewery":

"Temple Bar"},

{

"id":

"5af37b18d30b7fc8e7d68f16",

"name":

"Leffe",

"description":

"the best beer ever",

"abv":

"5",

"brewery":

"coffee Bruxelles"},

{

"id":

"5af37bc5d30b7fc8e7d68f17",

"name":

"chouffe",

"description":

" strong blond 'gnome beer', first brewed in

1982",

"abv":

"5",

"brewery":

"coffee chouffe"}

}

]

UPDATE =

The screenshot shows a REST client interface with a PUT request to `http://portalpega.cl/alexRandomBeer/api/beer/5af379e4d30b7fc8e7d68f15/`. The request body is a JSON object representing a beer record. The interface includes tabs for Authorization, Headers (1), Body, Pre-request Script, and Tests. The Body tab is active, showing the JSON payload. Below the request, the response headers are displayed, including `connection: keep-alive`, `content-encoding: gzip`, `date: Wed, 09 May 2018 22:55:16 GMT`, `server: nginx`, `transfer-encoding: chunked`, and `vary: Accept-Encoding,User-Agent`. The status is 200 OK.

```
PUT http://portalpega.cl/alexRandomBeer/api/beer/5af379e4d30b7fc8e7d68f15/ Params Send
```

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "name": "GUINNESS STOUT",
3   "description": "very cool beer",
4   "abv": "4.2",
5   "brewery": "Temple Bar "
6 }
7 }
```

Body Cookies Headers (6) Test Results Status: 200 OK

connection → keep-alive

content-encoding → gzip

date → Wed, 09 May 2018 22:55:16 GMT

server → nginx

transfer-encoding → chunked

vary → Accept-Encoding,User-Agent

Get ALL =

The screenshot shows a REST client interface with a GET request to `http://portalpega.cl/alexRandomBeer/api/beer/`. The response body is a JSON array containing three beer records. The interface includes tabs for Authorization, Headers (1), Body, Pre-request Script, and Tests. The Body tab is active, showing the JSON array. Below the request, the response headers are displayed, including `connection: keep-alive`, `content-encoding: gzip`, `date: Wed, 09 May 2018 22:55:16 GMT`, `server: nginx`, `transfer-encoding: chunked`, and `vary: Accept-Encoding,User-Agent`. The status is 200 OK.

```
GET http://portalpega.cl/alexRandomBeer/api/beer/ Params Send
```

Authorization Headers (1) Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (7) Test Results Status: 200 OK

Pretty Raw Preview JSON

```
1 [
2   {
3     "id": "5af379e4d30b7fc8e7d68f15",
4     "name": "GUINNESS STOUT",
5     "description": "very cool beer",
6     "abv": "4.2",
7     "brewery": "Temple Bar "
8   },
9   {
10    "id": "5af37b18d30b7fc8e7d68f16",
11    "name": "Leffe",
12    "description": "the best beer ever",
13    "abv": "5",
14    "brewery": "coffee Bruxelles"
15  },
16  {
17    "id": "5af37bc5d30b7fc8e7d68f17",
18    "name": "chouffe",
19    "description": "strong blond 'gnome beer', first brewed in 1982",
20    "abv": "5",
21    "brewery": "coffee chouffe"
22  }
23 ]
```

connection → keep-alive

content-encoding → gzip

date → Wed, 09 May 2018 22:55:16 GMT

server → nginx

transfer-encoding → chunked

vary → Accept-Encoding,User-Agent

Delete =

DELETE

http://portalpega.cl/alexRandomBeer/api/beer/5af37b18d30b7fc8e7d68f16/

Params

Send

AuthorizationHeaders (1)BodyPre-request ScriptTests

TypeNo Auth

BodyCookiesHeaders (7)Test ResultsStatus: 200 OK

PrettyRawPreviewText

1deleted

Get All to check =

GET

http://portalpega.cl/alexRandomBeer/api/beer/

Params

Send

AuthorizationHeaders (1)BodyPre-request ScriptTests

TypeNo Auth

BodyCookiesHeaders (7)Test ResultsStatus: 200 OK

PrettyRawPreviewJSON

1[  
2 {  
3 "id": "5af379e4d30b7fc8e7d68f15",  
4 "name": "GUINNESS STOUT",  
5 "description": "very cool beer",  
6 "abv": "4.2",  
7 "brewery": "Temple Bar "  
8 },  
9 {  
10 "id": "5af37bc5d30b7fc8e7d68f17",  
11 "name": "chouffe",  
12 "description": "strong blond 'gnome beer', first brewed in 1982",  
13 "abv": "5",  
14 "brewery": "coffee chouffe"  
15 }  
16 ]

Get Random =

GET

http://portalpega.cl/alexRandomBeer/api/randomBeer/

Params

Send

AuthorizationHeaders (1)BodyPre-request ScriptTests

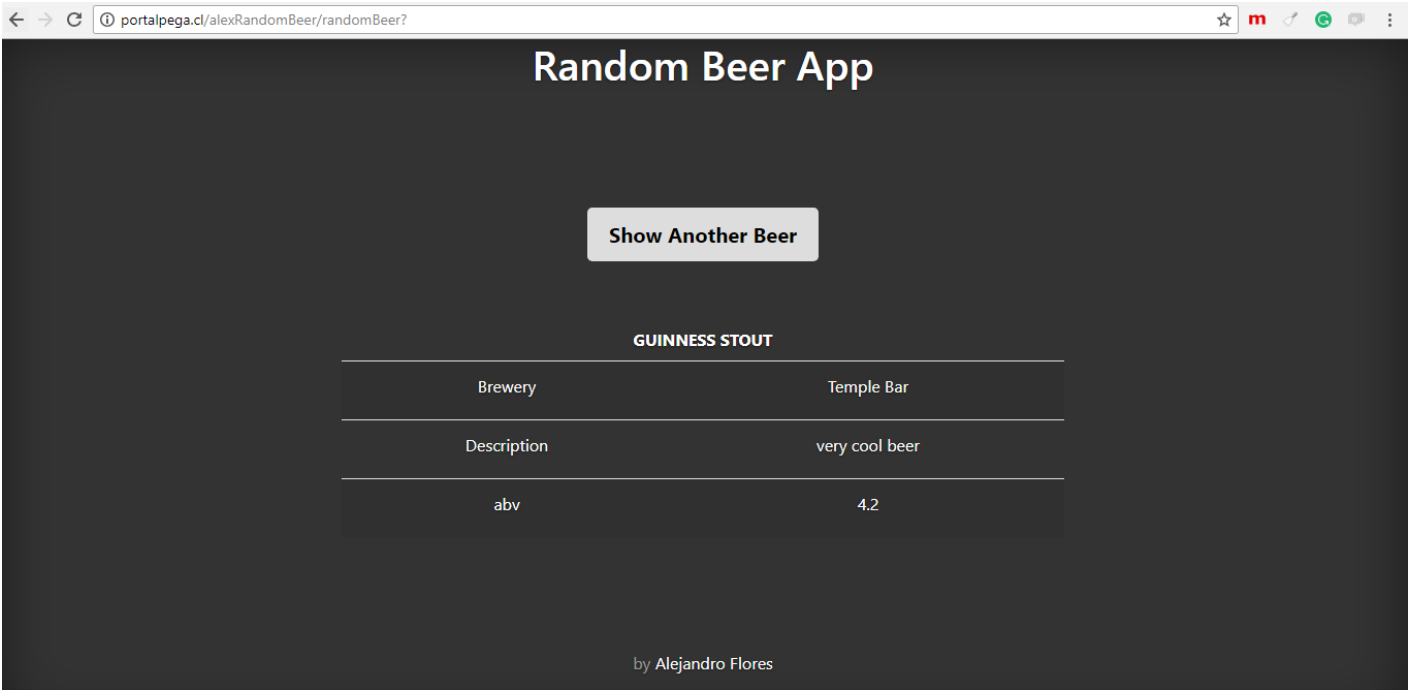
TypeNo Auth

BodyCookiesHeaders (7)Test ResultsStatus: 200

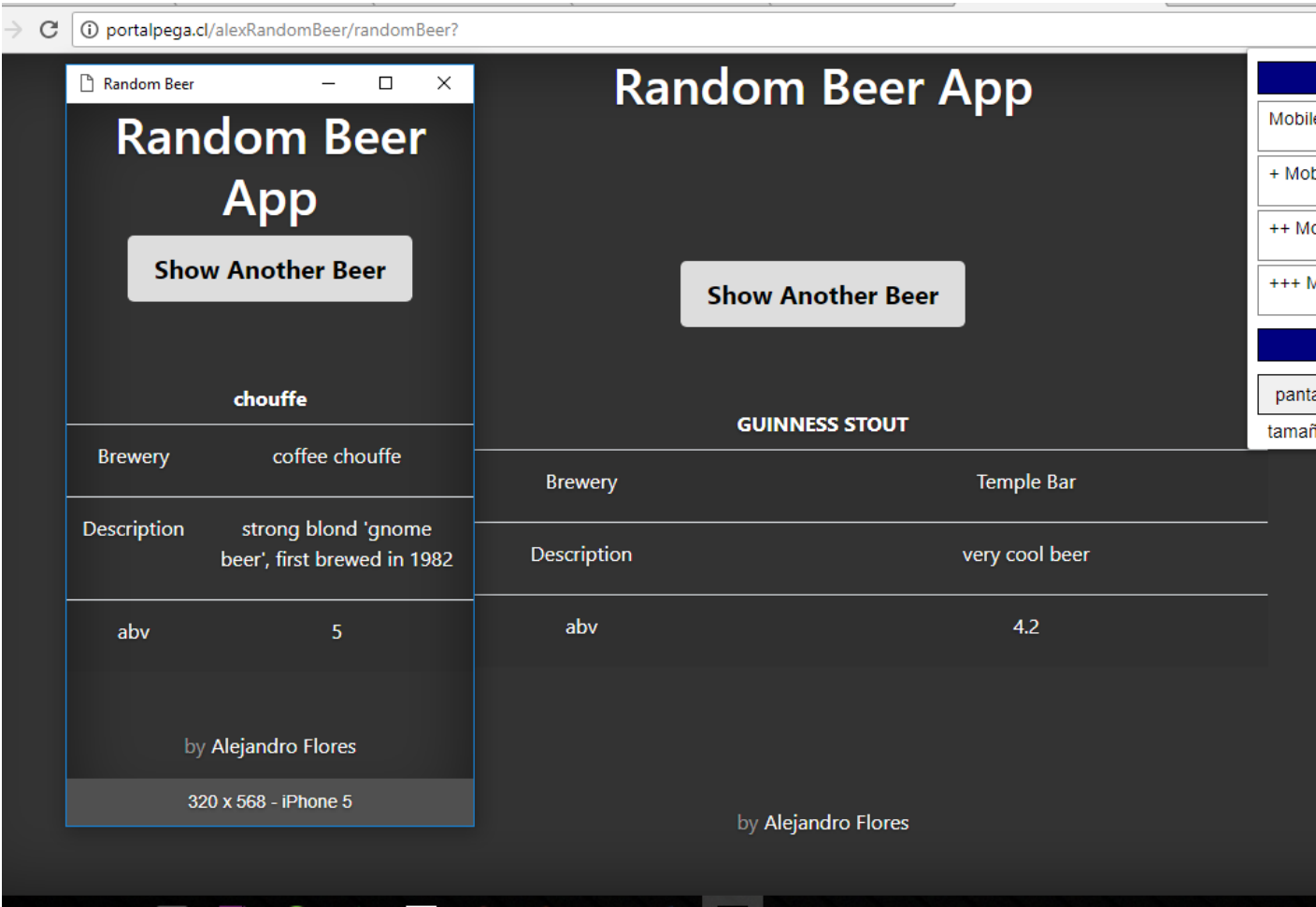
PrettyRawPreviewJSON

1 {  
2 "id": "5af379e4d30b7fc8e7d68f15",  
3 "name": "GUINNESS STOUT",  
4 "description": "very cool beer",  
5 "abv": "4.2",  
6 "brewery": "Temple Bar "  
7 }

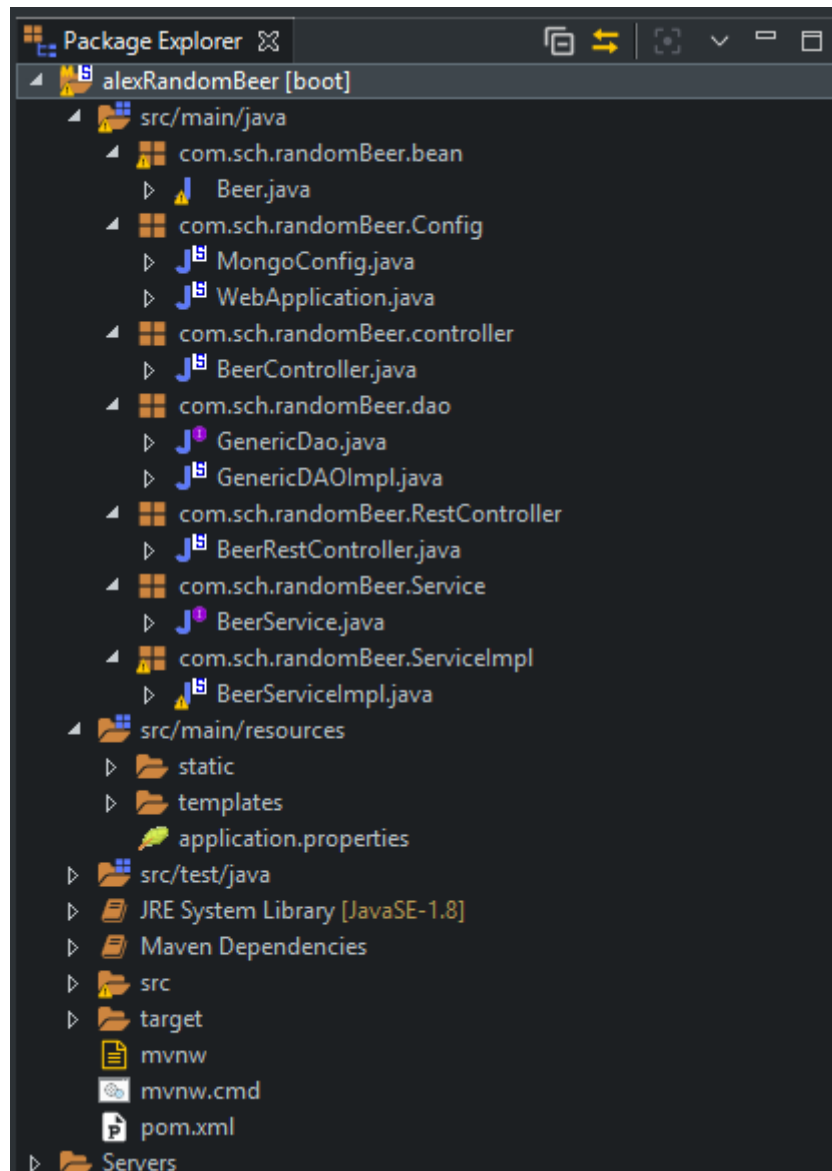
Finally website =



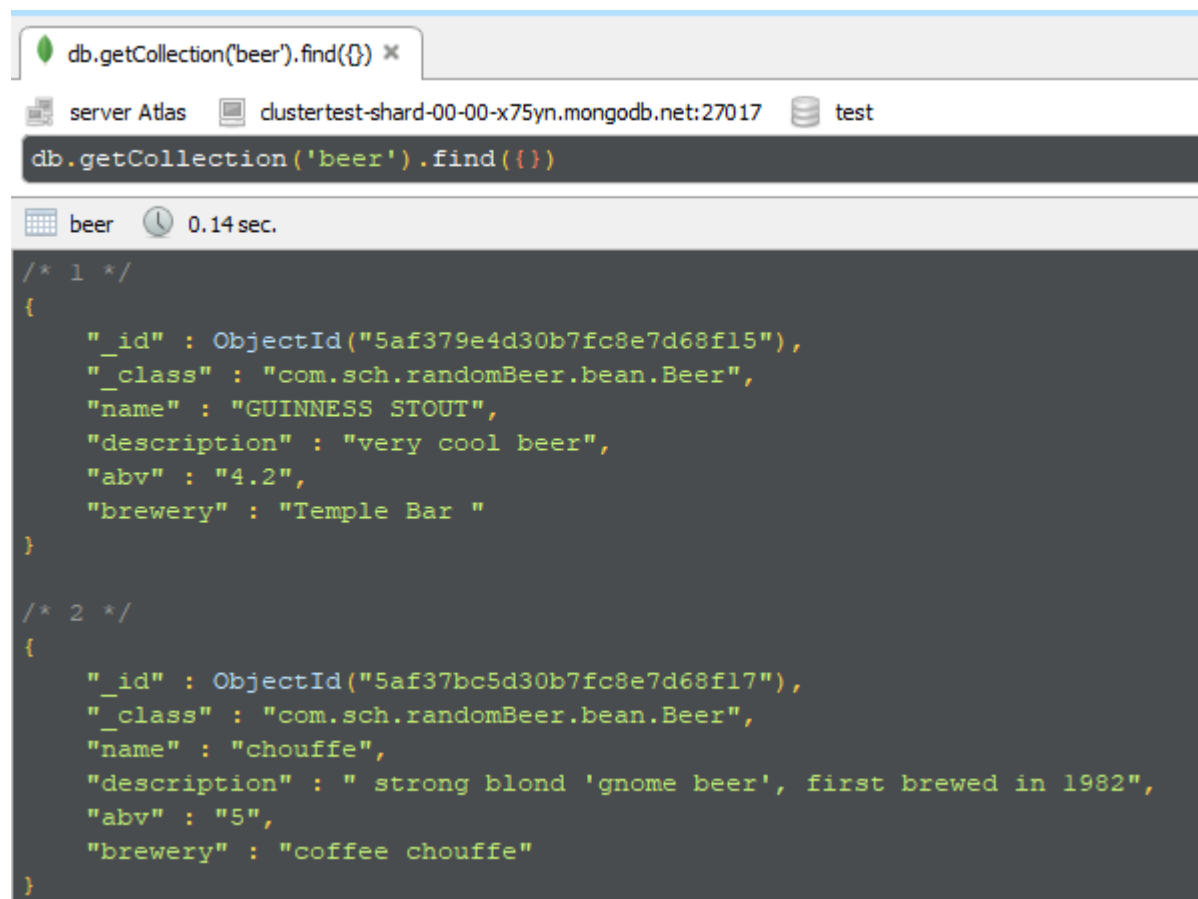
Bootstrap=



## Project Eclipse Architecture



Schema MongoDB =



The screenshot shows the MongoDB Atlas web interface. At the top, there's a browser tab with the URL `db.getCollection('beer').find({})`. Below the tab, the server information is displayed: `server Atlas`, `clustertest-shard-00-00-x75yn.mongodb.net:27017`, and the database name `test`. The main query bar contains `db.getCollection('beer').find({})`. Below the query bar, the results are shown for the `beer` collection, with a execution time of `0.14 sec.`. The results are displayed in a dark-themed code editor, showing two JSON documents. The first document is for a beer named "GUINNESS STOUT" with an alcohol content of 4.2% and is brewed at "Temple Bar". The second document is for a beer named "chouffe" with an alcohol content of 5% and is brewed at "coffee chouffe".

```
/* 1 */
{
  "_id" : ObjectId("5af379e4d30b7fc8e7d68f15"),
  "_class" : "com.sch.randomBeer.bean.Beer",
  "name" : "GUINNESS STOUT",
  "description" : "very cool beer",
  "abv" : "4.2",
  "brewery" : "Temple Bar "
}

/* 2 */
{
  "_id" : ObjectId("5af37bc5d30b7fc8e7d68f17"),
  "_class" : "com.sch.randomBeer.bean.Beer",
  "name" : "chouffe",
  "description" : " strong blond 'gnome beer', first brewed in 1982",
  "abv" : "5",
  "brewery" : "coffee chouffe"
}
```