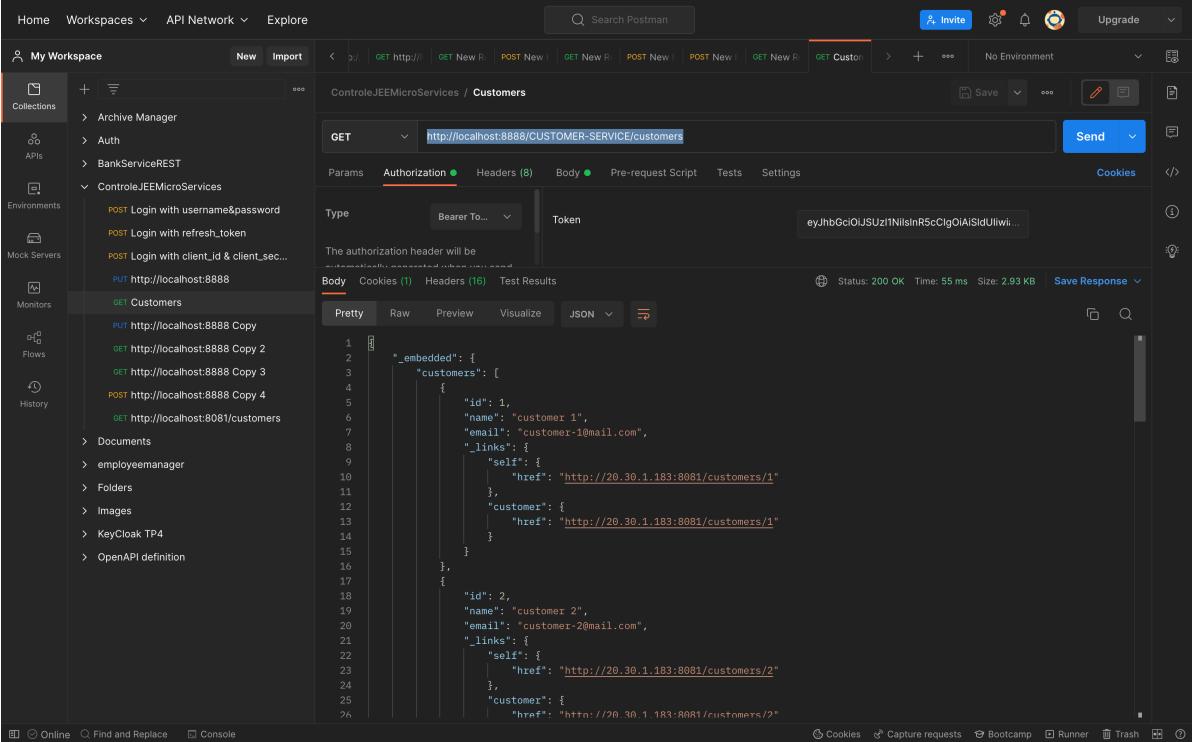


# Controle JEE Rapport

## 1. Customer Service



The screenshot shows the Postman application interface. On the left, there's a sidebar with sections like 'My Workspace', 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. In the main area, a collection named 'ControleJEEMicroServices' is selected. A specific API endpoint, 'Customers', is highlighted. The request method is 'GET' and the URL is 'http://localhost:8888/CUSTOMER-SERVICE/customers'. The 'Authorization' tab is active, showing a token type with a placeholder 'eyJhbGciOiJSUzI1Ni...'. The 'Body' tab shows the JSON response:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
{
  "_embedded": {
    "customers": [
      {
        "id": 1,
        "name": "customer 1",
        "email": "customer-1@mail.com",
        "_links": {
          "self": {
            "href": "http://20.30.1.183:8081/customers/1"
          },
          "customer": {
            "href": "http://20.30.1.183:8081/customers/1"
          }
        }
      },
      {
        "id": 2,
        "name": "customer 2",
        "email": "customer-2@mail.com",
        "_links": {
          "self": {
            "href": "http://20.30.1.183:8081/customers/2"
          },
          "customer": {
            "href": "http://20.30.1.183:8081/customers/2"
          }
        }
      }
    ]
  }
}
```

The screenshot shows the Postman interface with the following details:

- Collection:** ControleJEEMicroServices
- Request Type:** POST
- URL:** http://localhost:8888/CUSTOMER-SERVICE/customers
- Body (JSON):**

```

1
2   "name": "abdelalim",
3   "email": "abdelalim@mail.com"
4

```

- Response Status:** 201 Created
- Response Body (Pretty JSON):**

```

1
2   {
3     "id": 8,
4     "name": "abdelalim",
5     "email": "abdelalim@mail.com",
6     "_links": [
7       "self": {
8         "href": "http://20.30.1.183:8081/customers/8"
9       },
10      "customer": {
11        "href": "http://20.30.1.183:8081/customers/8"
12      }
13    }
14

```

The screenshot shows the Postman interface with the following details:

- Collection:** ControleJEEMicroServices
- Request Type:** PUT
- URL:** http://localhost:8888/CUSTOMER-SERVICE/customers/8
- Body (JSON):**

```

1
2   "name": "abdelalim edited",
3   "email": "abdelalim@mail.com"
4

```

- Response Status:** 200 OK
- Response Body (Pretty JSON):**

```

1
2   {
3     "id": 8,
4     "name": "abdelalim edited",
5     "email": "abdelalim@mail.com",
6     "_links": [
7       "self": {
8         "href": "http://20.30.1.183:8081/customers/8"
9       },
10      "customer": {
11        "href": "http://20.30.1.183:8081/customers/8"
12      }
13    }
14

```

The screenshot shows the Postman application interface. On the left, there's a sidebar with various sections like Home, Workspaces, API Network, and Explore. Under 'My Workspace', there are collections for 'Archive Manager', 'Auth', 'BankServiceREST', and 'ControleJEEMicroServices'. The 'ControleJEEMicroServices' collection contains several requests: 'POST Login with username&password', 'POST Login with refresh\_token', 'POST Login with client\_id & client\_sec...', 'GET POST Test', 'GET Customers', 'GET Bills', 'GET Products', and several copy requests ('PUT http://localhost:8888 Copy', 'GET http://localhost:8888 Copy 2', etc.). The main workspace shows a 'POST Test' request for 'ControleJEEMicroServices / POST Test'. The request method is 'GET' with the URL 'http://localhost:8888/CUSTOMER-SERVICE/customers/8'. The 'Body' tab is selected, showing the response: 'This request does not have a body'. Below the response, there are tabs for Body, Cookies (1), Headers (14), and Test Results. At the bottom, there are buttons for Pretty, Raw, Preview, Visualize, Text, and a 'Save Response' button. The status bar at the bottom indicates 'Status: 404 Not Found Time: 25 ms Size: 442 B'.

## 2. Inventory Service

ControleJEEMicroServices / Products

GET http://localhost:8888/INVENTORY-SERVICE/products

Key	Value	Description
id	1	Description
name	"Product 1"	
price	1500.0	
quantity	200	
links	[{"self": {"href": "http://20.30.1.183:8082/products/1"}, "product": {"href": "http://20.30.1.183:8082/products/1"}]	
id	2	
name	"Product 2"	

ControleJEEMicroServices / POST Test

POST http://localhost:8888/INVENTORY-SERVICE/products

Body
{"name": "Test product", "price": 2000, "quantity": 50}

Body	Cookies	Headers	Test Results
Pretty	Raw	Preview	Visualize
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
45	46	47	48
49	50	51	52
53	54	55	56
57	58	59	60
61	62	63	64
65	66	67	68
69	70	71	72
73	74	75	76
77	78	79	80
81	82	83	84
85	86	87	88
89	90	91	92
93	94	95	96
97	98	99	100
101	102	103	104
105	106	107	108
109	110	111	112
113	114	115	116
117	118	119	120
121	122	123	124
125	126	127	128
129	130	131	132
133	134	135	136
137	138	139	140
141	142	143	144
145	146	147	148
149	150	151	152
153	154	155	156
157	158	159	160
161	162	163	164
165	166	167	168
169	170	171	172
173	174	175	176
177	178	179	180
181	182	183	184
185	186	187	188
189	190	191	192
193	194	195	196
197	198	199	200
201	202	203	204
205	206	207	208
209	210	211	212
213	214	215	216
217	218	219	220
221	222	223	224
225	226	227	228
229	230	231	232
233	234	235	236
237	238	239	240
241	242	243	244
245	246	247	248
249	250	251	252
253	254	255	256
257	258	259	260
261	262	263	264
265	266	267	268
269	270	271	272
273	274	275	276
277	278	279	280
281	282	283	284
285	286	287	288
289	290	291	292
293	294	295	296
297	298	299	300
301	302	303	304
305	306	307	308
309	310	311	312
313	314	315	316
317	318	319	320
321	322	323	324
325	326	327	328
329	330	331	332
333	334	335	336
337	338	339	340
341	342	343	344
345	346	347	348
349	350	351	352
353	354	355	356
357	358	359	360
361	362	363	364
365	366	367	368
369	370	371	372
373	374	375	376
377	378	379	380
381	382	383	384
385	386	387	388
389	390	391	392
393	394	395	396
397	398	399	400
401	402	403	404
405	406	407	408
409	410	411	412
413	414	415	416
417	418	419	420
421	422	423	424
425	426	427	428
429	430	431	432
433	434	435	436
437	438	439	440
441	442	443	444
445	446	447	448
449	450	451	452
453	454	455	456
457	458	459	460
461	462	463	464
465	466	467	468
469	470	471	472
473	474	475	476
477	478	479	480
481	482	483	484
485	486	487	488
489	490	491	492
493	494	495	496
497	498	499	500
501	502	503	504
505	506	507	508
509	510	511	512
513	514	515	516
517	518	519	520
521	522	523	524
525	526	527	528
529	530	531	532
533	534	535	536
537	538	539	540
541	542	543	544
545	546	547	548
549	550	551	552
553	554	555	556
557	558	559	560
561	562	563	564
565	566	567	568
569	570	571	572
573	574	575	576
577	578	579	580
581	582	583	584
585	586	587	588
589	590	591	592
593	594	595	596
597	598	599	600
601	602	603	604
605	606	607	608
609	610	611	612
613	614	615	616
617	618	619	620
621	622	623	624
625	626	627	628
629	630	631	632
633	634	635	636
637	638	639	640
641	642	643	644
645	646	647	648
649	650	651	652
653	654	655	656
657	658	659	660
661	662	663	664
665	666	667	668
669	670	671	672
673	674	675	676
677	678	679	680
681	682	683	684
685	686	687	688
689	690	691	692
693	694	695	696
697	698	699	700
701	702	703	704
705	706	707	708
709	710	711	712
713	714	715	716
717	718	719	720
721	722	723	724
725	726	727	728
729	730	731	732
733	734	735	736
737	738	739	740
741	742	743	744
745	746	747	748
749	750	751	752
753	754	755	756
757	758	759	760
761	762	763	764
765	766	767	768
769	770	771	772
773	774	775	776
777	778	779	780
781	782	783	784
785	786	787	788
789	790	791	792
793	794	795	796
797	798	799	800
801	802	803	804
805	806	807	808
809	810	811	812
813	814	815	816
817	818	819	820
821	822	823	824
825	826	827	828
829	830	831	832
833	834	835	836
837	838	839	840
841	842	843	844
845	846	847	848
849	850	851	852
853	854	855	856
857	858	859	860
861	862	863	864
865	866	867	868
869	870	871	872
873	874	875	876
877	878	879	880
881	882	883	884
885	886	887	888
889	890	891	892
893	894	895	896
897	898	899	900
901	902	903	904
905	906	907	908
909	910	911	912
913	914	915	916
917	918	919	920
921	922	923	924
925	926	927	928
929	930	931	932
933	934	935	936
937	938	939	940
941	942	943	944
945	946	947	948
949	950	951	952
953	954	955	956
957	958	959	960
961	962	963	964
965	966	967	968
969	970	971	972
973	974	975	976
977	978	979	980
981	982	983	984
985	986	987	988
989	990	991	992
993	994	995	996
997	998	999	1000

The screenshot shows the Postman interface with a dark theme. On the left, the sidebar lists collections, environments, mock servers, monitors, flows, and history. The main workspace is titled "My Workspace". A test collection named "ControleJEEMicroServices" is selected, containing several requests like "POST Login with username&password" and "PUT POST Test". The "PUT POST Test" request is currently active, showing a PUT method to the URL `http://localhost:8888/INVENTORY-SERVICE/products/5`. The "Body" tab is selected, displaying a JSON payload:

```

1
2   "id": 5,
3   "name": "Test product edited",
4   "price": 2000,
5   "quantity": 50
6

```

The response status is 200 OK, with a response body showing the updated product details:

```

1
2   "id": 5,
3   "name": "Test product edited",
4   "price": 2000.0,
5   "quantity": 50,
6   "_links": [
7     "self": {
8       "href": "http://20.30.1.183:8082/products/5"
9     },
10    "product": [
11      "href": "http://20.30.1.183:8082/products/5"
12    }
13 ]
14

```

This screenshot shows the same Postman interface as the first one, but the "PUT POST Test" request has been replaced by a new test collection. The new collection is also titled "ControleJEEMicroServices" and contains a single "DELETE POST Test" request. This request uses the DELETE method on the same URL `http://localhost:8888/INVENTORY-SERVICE/products/5`. The "Body" tab is selected, showing the same JSON payload as before:

```

1
2   "id": 5,
3   "name": "Test product edited",
4   "price": 2000,
5   "quantity": 50
6

```

The response status is 204 No Content, indicating the product was successfully deleted.

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections like 'My Workspace', 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. The main area displays a 'POST Test' collection under 'ControleJEEMicroServices'. A specific test case 'GET Customers' is selected, showing a 'GET' request to 'http://localhost:8888/INVENTORY-SERVICE/products/5'. The 'Body' tab is active, containing the following JSON payload:

```
1
2   "name": "Test product edited",
3   "price": 2000,
4   "quantity": 50
5
6
```

Below the request, the response status is shown as 'Status: 404 Not Found Time: 17 ms Size: 442 B Save Response'. At the bottom, there are tabs for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'Text'. The bottom right corner features a small preview window of another Postman session.

### 3. Gateway Service with dynamic routes

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Home, Workspaces, API Network, and Explore. The main area displays a collection named "ControleJEMicroServices" under the "Collections" section. This collection contains several API endpoints:

- GET Bills
- POST Login with username&password
- POST Login with refresh\_token
- POST Login with client\_id & client\_sec...
- PUT http://localhost:8888
- GET Customers
- GET Bills
- GET Products
- PUT http://localhost:8888 Copy
- GET http://localhost:8888 Copy 2
- GET http://localhost:8888 Copy 3
- POST http://localhost:8888 Copy 4
- GET http://localhost:8081/customers

On the right, a specific endpoint "GET Bills" is selected. The request URL is "http://localhost:8888/BILLING-SERVICE/bills". The "Params" tab is active, showing a single parameter "Key" with the value "Value". The "Body" tab shows the response body in JSON format:

```
1  "embedded": {  
2      "bills": []  
3  },  
4  "  
5  "_links": {  
6      "self": {  
7          "href": "http://20.30.1.183:8083/bills"  
8      },  
9      "profile": {  
10         "href": "http://20.30.1.183:8083/profile/bills"  
11     },  
12     "search": {  
13         "href": "http://20.30.1.183:8083/bills/search"  
14     }  
15 },  
16  "  
17  "page": {  
18      "size": 20,  
19      "totalElements": 0,  
20      "totalPages": 0,  
21      "number": 0  
22  }
```

The status bar at the bottom indicates "Status: 200 OK", "Time: 42 ms", and "Size: 943 B".

## 4. Discovery Service

The screenshot shows the Spring Eureka dashboard running at `localhost:8761`. The top navigation bar includes links for "HOME" and "LAST 1000 SINCE STARTUP". The main content area is divided into sections:

- System Status**: A table showing system configuration.

Environment	test	Current time	2022-12-22T09:23:54 +0100
Data center	default	Uptime	00:20
		Lease expiration enabled	true
		Renews threshold	10
		Renews (last min)	20
- DS Replicas**: A table showing registered instances.

Application	AMIs	Availability Zones	Status
BILLING-SERVICE	n/a (1)	(1)	UP (1) - <a href="#">192.168.100.28:billing-service:8083</a>
CONFIG-SERVER	n/a (1)	(1)	UP (1) - <a href="#">192.168.100.28:config-server:8088</a>
CUSTOMER-SERVICE	n/a (1)	(1)	UP (1) - <a href="#">192.168.100.28:customer-service:8081</a>
GATEWAY-SERVICE	n/a (1)	(1)	UP (1) - <a href="#">192.168.100.28:gateway-service:8888</a>
INVENTORY-SERVICE	n/a (1)	(1)	UP (1) - <a href="#">192.168.100.28:inventory-service:8082</a>
- General Info**: A section containing the text "localhost".

## 5. Billing Service

Home Workspaces API Network Explore

My Workspace New Import < /api GET New R POST New | GET New R POST New | POST New | GET New R GET Cust... GET Prod... > + ... No Environment

Collections + ... ControleJEEMicroServices / Bills

APIs Environments Mock Servers Monitors Flows History

ControleJEEMicroServices

- > Archive Manager
- > Auth
- > BankServiceREST
- > ControleJEEMicroServices
  - POST Login with username&password
  - POST Login with refresh\_token
  - POST Login with client\_id & client\_sec...
  - PUT http://localhost:8888
  - GET Customers
  - GET Bills
  - GET Products
  - PUT http://localhost:8888 Copy
  - GET http://localhost:8888 Copy 2
  - GET http://localhost:8888 Copy 3
  - POST http://localhost:8888 Copy 4
  - GET http://localhost:8081/customers
  - > Documents
  - > employeemanager
  - > Folders
  - > Images
  - > Keycloak TP4
  - > OpenAPI definition

Search Postman

GET http://localhost:8888/BILLING-SERVICE/bills

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description	...	

Body Cookies (1) Headers (16) Test Results Status: 200 OK Time: 42 ms Size: 943 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "_embedded": {
3     "bills": []
4   },
5   "_links": [
6     "self": {
7       "href": "http://20.30.1.183:8083/bills"
8     },
9     "profile": {
10       "href": "http://20.30.1.183:8083/profile/bills"
11     },
12     "search": {
13       "href": "http://20.30.1.183:8083/bills/search"
14     }
15   },
16   "page": {
17     "size": 20,
18     "totalElements": 0,
19     "totalPages": 0,
20     "number": 0
21   }
22 }

```

Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

Home Workspaces API Network Explore

My Workspace New Import GET POST Test ● GET Customers POST Login with username&password POST Bills ● + ... No Environment

Collections + ... ControleJEEMicroServices / Bills

APIs Environments Mock Servers Monitors Flows History

ControleJEEMicroServices

- > Archive Manager
- > Auth
- > BankServiceREST
- > ControleJEEMicroServices
  - POST Login with username&password
  - POST Login with refresh\_token
  - POST Login with client\_id & client\_sec...
  - PUT POST Test
  - GET Customers
  - POST Bills
  - GET Products
  - PUT http://localhost:8888 Copy
  - GET http://localhost:8888 Copy 2
  - GET http://localhost:8888 Copy 3
  - POST http://localhost:8888 Copy 4
  - GET http://localhost:8081/customers
  - > Documents
  - > employeemanager
  - > Folders
  - > Images
  - > Keycloak TP4
  - > OpenAPI definition

Search Postman

POST http://localhost:8888/BILLING-SERVICE/fullBill

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

Body none form-data x-www-form-urlencoded raw binary GraphQL JSON

Pretty Raw Preview Visualize JSON

```

1 {
2   "billingDate": "2022-12-24T00:00:00.000+00:00",
3   "productItems": [
4     {
5       "quantity": 6,
6       "productID": 2,
7       "price": 3000,
8       "product": {
9         "id": 2,
10        "name": "Product 2",
11        "price": 3000,
12        "quantity": 494
13      }
14    }
15  ],
16  "customerId": 7
17 }

```

```

1 {
2   "id": 2,
3   "billingDate": "2022-12-24T00:00:00.000+00:00",
4   "productItems": [
5     {
6       "id": 5,
7       "quantity": 6,
8       "productID": 2,
9       "price": 3000.0,
10      "product": {
11        "id": 2,
12        "name": "Product 2",
13        "price": 3000.0,
14        "quantity": 494
15      }
16    }
17  ]
18 }

```

Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

The screenshot shows the Postman interface with a dark theme. On the left, the sidebar lists collections, environments, mock servers, and monitors. The main workspace is titled "My Workspace". A search bar at the top right says "Search Postman". Below it, there are tabs for "Collections", "APIs", "Environments", "Mock Servers", and "Monitors". The "Collections" tab is active, showing a list of items under "ControleJEEMicroServices": "POST Login with username&password", "POST Login with refresh\_token", "POST Login with client\_id & client\_sec...", "PUT POST Test", "GET Customers", "POST Bills", "GET Products", "PUT http://localhost:8888 Copy", "GET http://localhost:8888 Copy 2", "GET http://localhost:8888 Copy 3", "POST http://localhost:8888 Copy 4", and "GET http://localhost:8081/customers".

The main panel displays a request for "ControleJEEMicroServices / Bills". The method is "GET" and the URL is "http://localhost:8888/BILLING-SERVICE/fullBill/2". The "Body" tab is selected, showing a JSON response:

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
{
  "id": 2,
  "billingDate": "2022-12-24T00:00:00.000+00:00",
  "productItems": [
    {
      "id": 5,
      "quantity": 6,
      "productId": 2,
      "price": 3000.0,
      "product": {
        "id": 2,
        "name": "Product 2",
        "price": 3000.0,
        "quantity": 488
      }
    ],
    "customerID": 7,
    "customer": {
      "id": 7,
      "name": "wahbi",
      "email": "wahbi@mail.com"
    }
  }
}

```

The status bar at the bottom indicates "Status: 200 OK Time: 42 ms Size: 652 B Save Response".

This screenshot is similar to the one above, showing the same workspace and collection structure. The main difference is the request type: "DELETE" instead of "GET". The URL is "http://localhost:8888/BILLING-SERVICE/fullBill/2". The "Body" tab is selected, showing a blank response:

```

1

```

The status bar at the bottom indicates "Status: 200 OK Time: 30 ms Size: 346 B Save Response".

## 6. Angular Client

## Customers Management

A screenshot of a web browser displaying the 'Customers List' page at `localhost:4200/user/customers`. The page title is 'JEE-Contrôle'. The main content area shows a table with 5 rows of customer data:

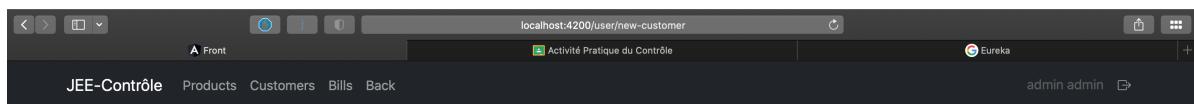
ID	Customer name	Customer email	Bills
1	customer 1	customer-1@mail.com	
2	customer 2	customer-2@mail.com	
3	customer 3	customer-3@mail.com	
4	achraf	achraf@mail.com	

Below the table, there is a pagination section showing '1 / 2' and '5 per page' with buttons for '1' and '2'.

A screenshot of a web browser displaying the 'Customers List' page at `localhost:4200/user/customers`. The page title is 'JEE-Contrôle'. The main content area shows a table with 1 row of customer data:

ID	Customer name	Customer email	Bills
6	khalid	khalid@mail.com	

Below the table, there is a pagination section showing '2 / 2' and '5 per page' with buttons for '1' and '2'.

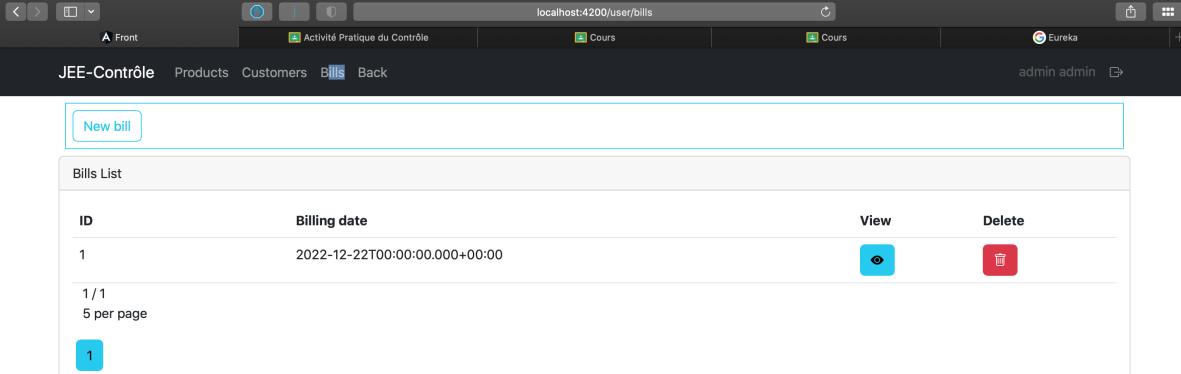


ID	Customer name	Customer email	Bills
6	khalid	khalid@mail.com	
7	wahbi	wahbi@mail.com	

2 / 2  
5 per page

[1](#) [2](#)

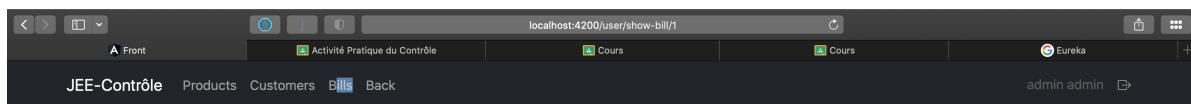
## Bills Management



The screenshot shows a web application interface for managing bills. At the top, there is a navigation bar with links for 'Front', 'Activité Pratique du Contrôle', 'Cours', 'Eureka', and user information 'admin admin'. Below the navigation bar, the main title 'JEE-Contrôle' is followed by 'Products', 'Customers', 'Bills', and 'Back' buttons. A 'New bill' button is located at the top left of the main content area, which is highlighted with a blue border. The main content area is titled 'Bills List' and contains a table with one row. The table has columns for 'ID', 'Billing date', 'View', and 'Delete'. The first row shows ID '1' and Billing date '2022-12-22T00:00:00.000+00:00'. To the right of the table are 'View' and 'Delete' buttons. Below the table, it says '1 / 1' and '5 per page'. A small number '1' is also visible at the bottom left of the table area.

ID	Billing date	View	Delete
1	2022-12-22T00:00:00.000+00:00		

1 / 1  
5 per page  
1



### Bill 1

Ordered at 22 - 12 - 2022

#### Customer

wahbi

wahbi@mail.com

#### Product items

Product id	Product name	Product unit price	Product quantity	Total price
2	Product 2	3000 Dhs	6	18000 Dhs
3	Product 3	900000 Dhs	5	4500000 Dhs
4	Test product	2000 Dhs	5	10000 Dhs
1	Test product	2000 Dhs	39	78000 Dhs



customer-1-new@mail.com	customer-1-new@mail.com	khalid@mail.com
<button>Select</button>	<button>Select</button>	<button>Select</button>
wahbi wahbi@mail.com		
<b>Selected</b>		

[Previous page](#)
1 / 2
100 per page
[Next page](#)

Products

<b>Test product</b> 2000 Dhs      11 Left <input type="text" value="8"/> <a href="#">Remove</a>	<b>Product 2</b> 3000 Dhs      488 Left <input type="text" value="7"/> <a href="#">Remove</a>	<b>Product 3</b> 900000 Dhs      45 Left <a href="#">Add</a>
<b>Test product</b> 2000 Dhs      45 Left <a href="#">Add</a>		

[Previous page](#)
1 / 2
100 per page
[Next page](#)

[Add bill](#)

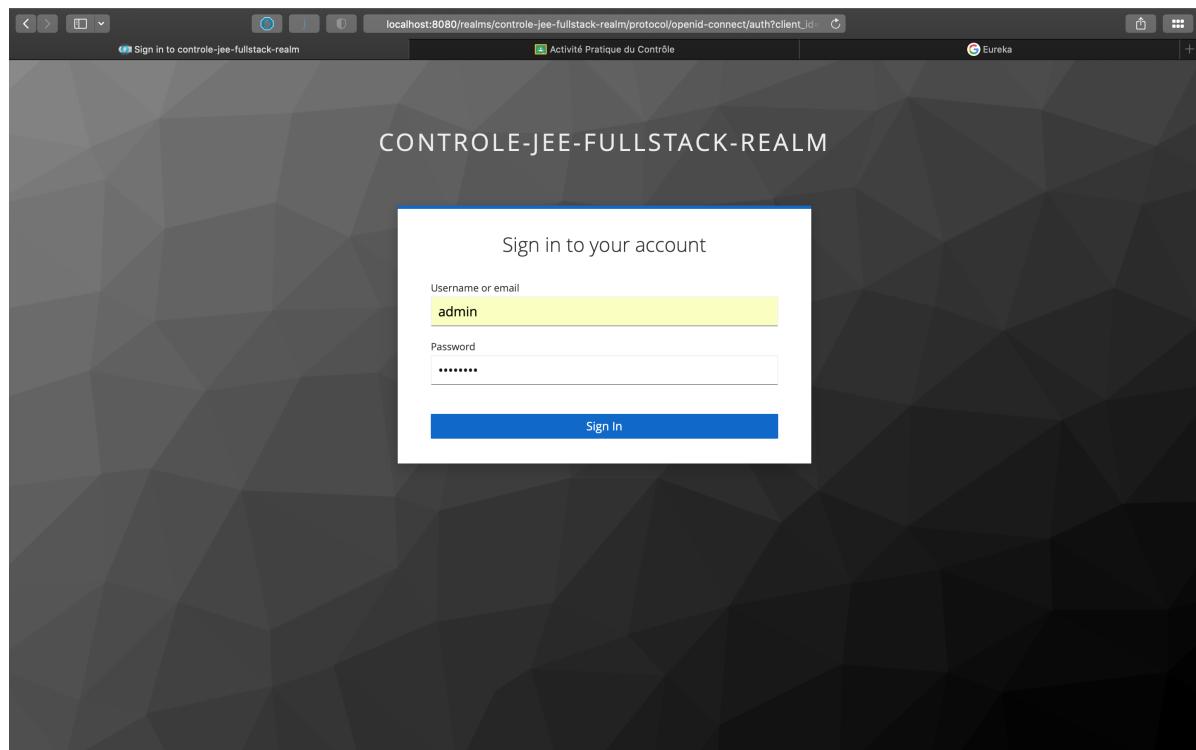
A screenshot of a web browser window titled "localhost:4200/user/bills". The header includes navigation icons and tabs for "Front", "Activité Pratique du Contrôle", "Cours", "Cours", "Eureka", and user "admin admin". The main content area is titled "Bills List" and contains a table with two rows. The columns are "ID", "Billing date", "View", and "Delete". Row 1 has ID 1 and Billing date 2022-12-22T00:00:00.000+00:00, with View and Delete buttons. Row 3 has ID 3 and Billing date 2022-12-20T00:00:00.000+00:00, also with View and Delete buttons. Below the table, it says "1 / 1" and "5 per page". A blue button labeled "1" is at the bottom left.

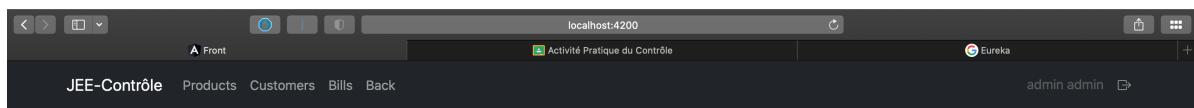


A screenshot of a web browser window titled "localhost:4200/user/show-bill/3". The header is identical to the previous screenshot. The main content shows "Bill 3" ordered at "20 - 12 - 2022". It details a customer "wahbi" with email "wahbi@mail.com". Below is a table for "Product items" with columns: Product id, Product name, Product unit price, Product quantity, and Total price. Two items are listed: item 1 (Test product) at 2000 Dhs each with 8 units, totaling 16000 Dhs; and item 2 (Product 2) at 3000 Dhs each with 7 units, totaling 21000 Dhs.



## 7. Keycloak Integration





## 8. Configuration Server

Ce service permet de centraliser les configurations de tous les autres micro services

```

server.port=8888
spring.cloud.config.server.bootstrap=true
spring.cloud.config.server.git.uri=file://${user.home}/Documents/GitHub/git-config-local-repo
spring.cloud.config.server.git.default-label=master
spring.cloud.discovery.enabled=true
spring.application.name=config-server

```

