Sample REST Service

Sample REST Service Documentation – Version 0.0.1, 2019-02-14

Table of Contents

Introduction

Resources

1. Customer

1.1. Create Customer

1.1.1. Request structure

```
POST /jpa/customers/ HTTP/1.1
Accept: application/json
Host: localhost:0
Content-Length: 159
Content-Type: application/json
Authorization: Bearer oauth-2-token-from-auth-service

{
  "firstName" : "firstName1",
  "lastName" : "lastName1",
  "superSecretData" : "superSecretData1",
  "sampleIgnoreInPublic" : "sampleIgnoreInPublic1"
}
```

1.1.2. Request fields

Path	Type	Description
`firstName`	`String`	Customer’s first name
`lastName`	`String`	Customer’s last name
`superSecretData`	`String`	Some secret data
`sampleIgnoreInPublic`	`String`	Some data that will be ignored while in public view

1.1.3. Response fields

Path	Type	Description

Path	Type	Description
firstName	String	Customer’s first name
lastName	String	Customer’s last name
superSecretData	String	Some secret data
created	String	Created timestamp
updated	String	Updated timestamp
id	Number	ID

1.1.4. Example response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 213

{
  "created" : "2019-02-14T00:46:39.331+0000",
  "updated" : "2019-02-14T00:46:39.331+0000",
  "id" : 5,
  "firstName" : "firstName1",
  "lastName" : "lastName1",
  "superSecretData" : "superSecretData1"
}
```

1.1.5. CURL request

```
$ curl 'http://localhost:0/jpa/customers/' -i -X POST \
    -H 'Accept: application/json' \
    -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer oauth-2-token-from-auth-service' \
    -d '{
  "firstName" : "firstName1",
  "lastName" : "lastName1",
  "superSecretData" : "superSecretData1",
  "sampleIgnoreInPublic" : "sampleIgnoreInPublic1"
}'
```

1.2. Get Customer

1.2.1. Request structure

```
GET /jpa/customers/4 HTTP/1.1
Accept: application/json
Host: localhost:0
Authorization: Bearer oauth-2-token-from-auth-service
```

1.2.2. Path parameters

Table 1. /jpa/customers/{id}

Parameter	Description
id	Customer ID to get

1.2.3. Response fields

Path	Type	Description
firstName	String	Customer’s first name
lastName	String	Customer’s last name
superSecretData	String	Some secret data
created	String	Created timestamp
updated	String	Updated timestamp
id	Number	ID

1.2.4. Example response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 213

{
  "created" : "2019-02-14T00:46:39.230+0000",
  "updated" : "2019-02-14T00:46:39.230+0000",
  "id" : 4,
  "firstName" : "firstName1",
  "lastName" : "lastName1",
  "superSecretData" : "superSecretData1"
}
```

1.2.5. CURL request

```
$ curl 'http://localhost:0/jpa/customers/4' -i -X GET \
    -H 'Accept: application/json' \
    -H 'Authorization: Bearer oauth-2-token-from-auth-service'
```

1.3. Get all Customers

1.3.1. Request structure

```
GET /jpa/customers/?page=0&size=2147483647 HTTP/1.1
Accept: application/json
Host: localhost:0
Authorization: Bearer oauth-2-token-from-auth-service
```

1.3.2. Request fields

Parameter	Description
page	Page number to get
size	Number of results per page

1.3.3. Response fields

Path	Type	Description
list[]	Array	List of customers
list[].firstName	String	Customer’s first name

Path	Type	Description
`list[].lastName`	`String`	Customer’s last name
`list[].superSecretData`	`String`	Some secret data
`list[].created`	`String`	Created timestamp
`list[].updated`	`String`	Updated timestamp
`list[].id`	`Number`	ID
`currentPage`	`Number`	Current page number
`totalPages`	`Number`	Total number of pages
`totalSize`	`Number`	Number of results
`first`	`Boolean`	Is it first page
`last`	`Boolean`	Is it last page

1.3.4. Example response

```
HTTP/1.1 200 OK
Content-Length: 806
Content-Type: application/json;charset=UTF-8

{
  "currentPage" : 0,
  "totalPages" : 1,
  "totalSize" : 3,
  "first" : true,
  "last" : true,
  "list" : [ {
    "created" : "2019-02-14T00:46:38.703+0000",
    "updated" : "2019-02-14T00:46:38.703+0000",
    "id" : 1,
    "firstName" : "firstName0",
    "lastName" : "lastName0",
    "superSecretData" : "superSecretData0"
  }, {
    "created" : "2019-02-14T00:46:38.788+0000",
    "updated" : "2019-02-14T00:46:38.788+0000",
    "id" : 2,
    "firstName" : "firstName1",
    "lastName" : "lastName1",
    "superSecretData" : "superSecretData1"
  }, {
    "created" : "2019-02-14T00:46:38.796+0000",
    "updated" : "2019-02-14T00:46:38.796+0000",
    "id" : 3,
    "firstName" : "firstName2",
    "lastName" : "lastName2",
    "superSecretData" : "superSecretData2"
  } ]
}
```

1.3.5. CURL request

```
$ curl 'http://localhost:0/jpa/customers/?page=0&size=2147483647' -i -X GET \
    -H 'Accept: application/json' \
    -H 'Authorization: Bearer oauth-2-token-from-auth-service'
```

1.4. Get Customers by Last Name

1.4.1. Request structure

```
GET /jpa/customers/search/findAllByLastName/lastName1 HTTP/1.1
Accept: application/json
Host: localhost:0
Authorization: Bearer oauth-2-token-from-auth-service
```

1.4.2. Path parameters

Table 2. /jpa/customers/search/findAllByLastName/{lastName}

Parameter	Description
`lastName`	Last name to search by

1.4.3. Response fields

Path	Type	Description
`[].firstName`	`String`	Customer’s first name
`[].lastName`	`String`	Customer’s last name
`[].superSecretData`	`String`	Some secret data
`[].created`	`String`	Created timestamp
`[].updated`	`String`	Updated timestamp
`[].id`	`Number`	ID

1.4.4. Example response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 217

[ {
  "created" : "2019-02-14T00:46:39.436+0000",
  "updated" : "2019-02-14T00:46:39.436+0000",
  "id" : 6,
  "firstName" : "firstName1",
  "lastName" : "lastName1",
  "superSecretData" : "superSecretData1"
} ]
```

1.4.5. CURL request

```
$ curl 'http://localhost:0/jpa/customers/search/findAllByLastName/lastName1' -i -X GET \
    -H 'Accept: application/json' \
    -H 'Authorization: Bearer oauth-2-token-from-auth-service'
```

Version 0.0.1
Last updated 2019-01-30 10:27:14 PST