

Api Documentation

Overview

Api Documentation

Version information

Version : 1.0

License information

License : Apache 2.0

License URL : <http://www.apache.org/licenses/LICENSE-2.0>

Terms of service : urn:tos

URI scheme

Host : localhost:8080

BasePath : /

Tags

- customer-controller : Customer Controller
- role-controller : Role Controller
- user-controller : User Controller
- warehouse-controller : Warehouse Controller

Paths

Creates a new Customer.

POST /customer

Parameters

Type	Name	Description	Schema
Body	newCustomer <i>required</i>	newCustomer	Customer

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- customer-controller

Returns list of all Customers in the system.

GET /customers

Responses

HTTP Code	Description	Schema
200	OK	< Customer > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- customer-controller

Returns a specific Customer by their identifier. 404 if does not exist.

```
GET /customers/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- customer-controller

Deletes a Customer from the system. 404 if the customer's identifier is not found.

```
DELETE /deleteCustomer/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `application/json`

Tags

- customer-controller

Deletes a Role from the system. 404 if the role's identifier is not found.

```
DELETE /deleteRole/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `application/json`

Tags

- role-controller

Deletes a User from the system. 404 if the user's identifier is not found.

```
DELETE /deleteUser/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity

HTTP Code	Description	Schema
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `application/json`

Tags

- user-controller

Deletes a Warehouse from the system. 404 if the warehouse's identifier is not found.

```
DELETE /deleteWarehouse/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `application/json`

Tags

- `warehouse-controller`

Creates a new Role.

POST /role

Parameters

Type	Name	Description	Schema
Body	newRole <i>required</i>	newRole	Role

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- `role-controller`

Returns list of all Roles in the system.

```
GET /roles
```

Responses

HTTP Code	Description	Schema
200	OK	< Role > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- [application/json](#)

Tags

- role-controller

Returns a specific Role by their identifier. 404 if does not exist.

```
GET /roles/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity

HTTP Code	Description	Schema
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- role-controller

Updates Customer.

```
PUT /updateCustomer/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)
Body	customer <i>required</i>	customer	Customer

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- `customer-controller`

Updates Role.

PUT /updateRole/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)
Body	role <i>required</i>	role	Role

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- `role-controller`

Updates User.

PUT /updateUser/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)
Body	user <i>required</i>	user	User

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- user-controller

Updates Warehouse.

PUT /updateWarehouse/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)
Body	warehouse <i>required</i>	warehouse	Warehouse

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- [application/json](#)

Produces

- [application/json](#)

Tags

- warehouse-controller

Creates a new User.

POST /user

Parameters

Type	Name	Description	Schema
Body	newUser <i>required</i>	newUser	User

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- [application/json](#)

Produces

- [application/json](#)

Tags

- user-controller

Returns User of a specific Customer by their identifier. 404 if does not exist.

GET /user/customer/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- customer-controller

Returns list of all Users in the system.

GET /users

Responses

HTTP Code	Description	Schema
200	OK	< User > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- user-controller

Returns Role of a specific User by their username. 404 if does not exist.

```
GET /users/role/username/{username}
```

Parameters

Type	Name	Description	Schema
Path	username <i>required</i>	username	string

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- user-controller

Returns list of Users who have the Role by Role identifier. 404 if does not exist.

```
GET /users/role/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< User > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- user-controller

Returns a specific User by their username. 404 if does not exist.

```
GET /users/username/{username}
```

Parameters

Type	Name	Description	Schema
Path	username <i>required</i>	username	string

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- user-controller

Returns Warehouse of a specific User by their identifier. 404 if does not exist.

```
GET /users/warehouse/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity

HTTP Code	Description	Schema
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- user-controller

Returns a specific User by their identifier. 404 if does not exist.

```
GET /users/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- user-controller

Creates a new Warehouse.

POST /warehouse

Parameters

Type	Name	Description	Schema
Body	newWarehouse <i>required</i>	newWarehouse	Warehouse

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- warehouse-controller

Returns list of all Warehouses in the system.

GET /warehouses

Responses

HTTP Code	Description	Schema
200	OK	< Warehouse > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- [application/json](#)

Tags

- warehouse-controller

Returns list of Warehouses where User belongs, specified by User identifier. 404 if does not exist.

GET /warehouses/user/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- warehouse-controller

Returns a specific Warehouse by their identifier. 404 if does not exist.

```
GET /warehouses/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	ResponseEntity
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`

Tags

- warehouse-controller

Definitions

Customer

Name	Schema
address <i>optional</i>	string
email <i>optional</i>	string
first_name <i>optional</i>	string
id <i>optional</i>	integer (int64)
last_name <i>optional</i>	string
phone <i>optional</i>	string
userID <i>optional</i>	User

ResponseEntity

Name	Schema
body <i>optional</i>	object

Name	Schema
statusCode <i>optional</i>	enum (100 CONTINUE, 101 SWITCHING_PROTOCOLS, 102 PROCESSING, 103 CHECKPOINT, 200 OK, 201 CREATED, 202 ACCEPTED, 203 NON_AUTHORITATIVE_INFORMATION, 204 NO_CONTENT, 205 RESET_CONTENT, 206 PARTIAL_CONTENT, 207 MULTI_STATUS, 208 ALREADY_REPORTED, 226 IM_USED, 300 MULTIPLE_CHOICES, 301 MOVED_PERMANENTLY, 302 FOUND, 302 MOVED_TEMPORARILY, 303 SEE_OTHER, 304 NOT_MODIFIED, 305 USE_PROXY, 307 TEMPORARY_REDIRECT, 308 PERMANENT_REDIRECT, 400 BAD_REQUEST, 401 UNAUTHORIZED, 402 PAYMENT_REQUIRED, 403 FORBIDDEN, 404 NOT_FOUND, 405 METHOD_NOT_ALLOWED, 406 NOT_ACCEPTABLE, 407 PROXY_AUTHENTICATION_REQUIRED, 408 REQUEST_TIMEOUT, 409 CONFLICT, 410 GONE, 411 LENGTH_REQUIRED, 412 PRECONDITION_FAILED, 413 PAYLOAD_TOO_LARGE, 413 REQUEST_ENTITY_TOO_LARGE, 414 URI_TOO_LONG, 414 REQUEST_URI_TOO_LONG, 415 UNSUPPORTED_MEDIA_TYPE, 416 REQUESTED_RANGE_NOT_SATISFIABLE, 417 EXPECTATION_FAILED, 418 I_AM_A_TEAPOT, 419 INSUFFICIENT_SPACE_ON_RESOURCE, 420 METHOD_FAILURE, 421 DESTINATION_LOCKED, 422 UNPROCESSABLE_ENTITY, 423 LOCKED, 424 FAILED_DEPENDENCY, 425 TOO_EARLY, 426 UPGRADE_REQUIRED, 428 PRECONDITION_REQUIRED, 429 TOO_MANY_REQUESTS, 431 REQUEST_HEADER_FIELDS_TOO_LARGE, 451 UNAVAILABLE_FOR_LEGAL_REASONS, 500 INTERNAL_SERVER_ERROR, 501 NOT_IMPLEMENTED, 502 BAD_GATEWAY, 503 SERVICE_UNAVAILABLE, 504 GATEWAY_TIMEOUT, 505 HTTP_VERSION_NOT_SUPPORTED, 506 VARIANT_ALSO_NEGOTIATES, 507 INSUFFICIENT_STORAGE, 508 LOOP_DETECTED, 509 BANDWIDTH_LIMIT_EXCEEDED, 510 NOT_EXTENDED, 511 NETWORK_AUTHENTICATION_REQUIRED)
statusCodeValue <i>optional</i>	integer (int32)

Role

Name	Schema
description <i>optional</i>	string
id <i>optional</i>	integer (int64)
roleName <i>optional</i>	enum (ADMIN, USER)
users <i>optional</i>	< User > array

User

Name	Schema
address <i>optional</i>	string
customers <i>optional</i>	< Customer > array
email <i>optional</i>	string
first_name <i>optional</i>	string
id <i>optional</i>	integer (int64)
last_name <i>optional</i>	string
password <i>optional</i>	string
phone <i>optional</i>	string

Name	Schema
roleID <i>optional</i>	Role
username <i>optional</i>	string
warehouseID <i>optional</i>	Warehouse

Warehouse

Name	Schema
company_name <i>optional</i>	string
id <i>optional</i>	integer (int64)
inventory_start_date <i>optional</i>	string (date-time)
location <i>optional</i>	string
users <i>optional</i>	< User > array