

Api Documentation

Table of Contents

1. Overview	1
1.1. Version information	1
1.2. License information	1
1.3. URI scheme	1
1.4. Tags	1
2. Paths	2
2.1. links	2
2.1.1. Responses	2
2.1.2. Produces	2
2.1.3. Tags	2
2.1.4. Security	2
2.2. handle	2
2.2.1. Parameters	2
2.2.2. Responses	3
2.2.3. Produces	3
2.2.4. Tags	3
2.2.5. Security	3
2.3. handle	3
2.3.1. Parameters	3
2.3.2. Responses	3
2.3.3. Produces	3
2.3.4. Tags	4
2.3.5. Security	4
2.4. handle	4
2.4.1. Parameters	4
2.4.2. Responses	4
2.4.3. Produces	4
2.4.4. Tags	4
2.4.5. Security	4
2.5. handle	5
2.5.1. Parameters	5
2.5.2. Responses	5
2.5.3. Produces	5
2.5.4. Tags	5
2.5.5. Security	5
2.6. getAllCountries	5
2.6.1. Responses	5
2.6.2. Produces	6

2.6.3. Tags	6
2.6.4. Security	6
2.7. getByCountry	6
2.7.1. Parameters	6
2.7.2. Responses	6
2.7.3. Produces	6
2.7.4. Tags	7
2.7.5. Security	7
2.8. getCitiesByDepartment	7
2.8.1. Parameters	7
2.8.2. Responses	7
2.8.3. Produces	7
2.8.4. Tags	7
2.8.5. Security	7
2.9. getCitiesByProvince	8
2.9.1. Parameters	8
2.9.2. Responses	8
2.9.3. Produces	8
2.9.4. Tags	8
2.9.5. Security	8
2.10. getDepartmentsByProvince	8
2.10.1. Parameters	8
2.10.2. Responses	9
2.10.3. Produces	9
2.10.4. Tags	9
2.10.5. Security	9
2.11. login	9
2.11.1. Parameters	9
2.11.2. Responses	9
2.11.3. Consumes	10
2.11.4. Produces	10
2.11.5. Tags	10
2.11.6. Security	10
2.12. create	10
2.12.1. Parameters	10
2.12.2. Responses	10
2.12.3. Consumes	10
2.12.4. Produces	11
2.12.5. Tags	11
2.12.6. Security	11
2.13. getMany	11

2.13.1. Parameters	11
2.13.2. Responses	11
2.13.3. Produces	11
2.13.4. Tags	11
2.13.5. Security	11
2.14. getOne	12
2.14.1. Parameters	12
2.14.2. Responses	12
2.14.3. Produces	12
2.14.4. Tags	12
2.14.5. Security	12
2.15. update	12
2.15.1. Parameters	12
2.15.2. Responses	13
2.15.3. Consumes	13
2.15.4. Produces	13
2.15.5. Tags	13
2.15.6. Security	13
2.16. delete	13
2.16.1. Parameters	13
2.16.2. Responses	13
2.16.3. Produces	14
2.16.4. Tags	14
2.16.5. Security	14
2.17. create	14
2.17.1. Parameters	14
2.17.2. Responses	14
2.17.3. Consumes	14
2.17.4. Produces	14
2.17.5. Tags	15
2.17.6. Security	15
2.18. getList	15
2.18.1. Parameters	15
2.18.2. Responses	15
2.18.3. Produces	16
2.18.4. Tags	16
2.18.5. Security	16
2.19. getOne	16
2.19.1. Parameters	16
2.19.2. Responses	16
2.19.3. Produces	16

2.19.4. Tags	16
2.19.5. Security	16
2.20. update	17
2.20.1. Parameters	17
2.20.2. Responses	17
2.20.3. Consumes	17
2.20.4. Produces	17
2.20.5. Tags	17
2.20.6. Security	17
2.21. delete	17
2.21.1. Parameters	18
2.21.2. Responses	18
2.21.3. Produces	18
2.21.4. Tags	18
2.21.5. Security	18
2.22. error	18
2.22.1. Responses	18
2.22.2. Consumes	19
2.22.3. Produces	19
2.22.4. Tags	19
2.22.5. Security	19
2.23. error	19
2.23.1. Responses	19
2.23.2. Produces	19
2.23.3. Tags	19
2.23.4. Security	19
2.24. error	20
2.24.1. Responses	20
2.24.2. Consumes	20
2.24.3. Produces	20
2.24.4. Tags	20
2.24.5. Security	20
2.25. error	20
2.25.1. Responses	20
2.25.2. Produces	21
2.25.3. Tags	21
2.25.4. Security	21
2.26. error	21
2.26.1. Responses	21
2.26.2. Consumes	21
2.26.3. Produces	21

2.26.4. Tags	21
2.26.5. Security	21
2.27. error	22
2.27.1. Responses	22
2.27.2. Consumes	22
2.27.3. Produces	22
2.27.4. Tags	22
2.27.5. Security	22
2.28. error	22
2.28.1. Responses	22
2.28.2. Consumes	23
2.28.3. Produces	23
2.28.4. Tags	23
2.28.5. Security	23
2.29. getLanguages	23
2.29.1. Responses	23
2.29.2. Produces	23
2.29.3. Tags	23
2.29.4. Security	23
2.30. updatePassword	24
2.30.1. Parameters	24
2.30.2. Responses	24
2.30.3. Consumes	24
2.30.4. Produces	24
2.30.5. Tags	24
2.30.6. Security	24
2.31. setRoleUser	24
2.31.1. Parameters	25
2.31.2. Responses	25
2.31.3. Consumes	25
2.31.4. Produces	25
2.31.5. Tags	25
2.31.6. Security	25
2.32. addUser	25
2.32.1. Parameters	25
2.32.2. Responses	26
2.32.3. Consumes	26
2.32.4. Produces	26
2.32.5. Tags	26
2.32.6. Security	26
2.33. getUsers	26

2.33.1. Parameters	26
2.33.2. Responses	26
2.33.3. Produces	27
2.33.4. Tags	27
2.33.5. Security	27
2.34. changeStatusAccount	27
2.34.1. Parameters	27
2.34.2. Responses	27
2.34.3. Consumes	27
2.34.4. Produces	28
2.34.5. Tags	28
2.34.6. Security	28
3. Security	29
3.1. JWT	29
4. Definitions	30
4.1. AddUserDto	30
4.2. AddressProjection	30
4.3. BackofficeDeleteResponseOfCity	30
4.4. BackofficeDeleteResponseOfDepartment	30
4.5. Character	30
4.6. City	31
4.7. Department	31
4.8. IterableOfCity	31
4.9. IterableOfDepartment	31
4.10. JWTTokenDto	31
4.11. Link	32
4.12. LoginDto	32
4.13. MapOfstringAndLink	32
4.14. ModelAndView	32
4.15. PageOfCity	35
4.16. PageOfDepartment	36
4.17. PageOfPageableUsers	36
4.18. Pageable	37
4.19. PageableUsers	38
4.20. RequestUserRoleDto	38
4.21. Sort	38
4.22. UserDto	39
4.23. View	39

Chapter 1. Overview

Api Documentation

1.1. Version information

Version : 1.0

1.2. License information

License : Apache 2.0

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

Terms of service : urn:tos

1.3. URI scheme

Host : localhost

BasePath : /

1.4. Tags

- Address : Address Controller
- Authorization : Authentication Controller
- I18n : I 18 N Controller
- Password : User Password Controller
- Role : Role Controller
- User : User Controller
- backoffice-city-controller : Backoffice City Controller
- backoffice-department-controller : Backoffice Department Controller
- basic-error-controller : Basic Error Controller
- operation-handler : Operation Handler
- web-mvc-links-handler : Web Mvc Links Handler

Chapter 2. Paths

2.1. links

GET /actuator

2.1.1. Responses

HTTP Code	Description	Schema
200	OK	< string, < string, Link > map > map

2.1.2. Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

2.1.3. Tags

- `web-mvc-links-handler`

2.1.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.2. handle

GET /actuator/health

2.2.1. Parameters

Type	Name	Description	Schema
Body	body <i>optional</i>	body	< string, string > map

2.2.2. Responses

HTTP Code	Description	Schema
200	OK	object

2.2.3. Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

2.2.4. Tags

- operation-handler

2.2.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.3. handle

```
GET /actuator/health/{component}
```

2.3.1. Parameters

Type	Name	Description	Schema
Body	body <i>optional</i>	body	< string, string > map

2.3.2. Responses

HTTP Code	Description	Schema
200	OK	object

2.3.3. Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

2.3.4. Tags

- operation-handler

2.3.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.4. handle

```
GET /actuator/health/{component}/{instance}
```

2.4.1. Parameters

Type	Name	Description	Schema
Body	body <i>optional</i>	body	< string, string > map

2.4.2. Responses

HTTP Code	Description	Schema
200	OK	object

2.4.3. Produces

- application/json
- application/vnd.spring-boot.actuator.v2+json

2.4.4. Tags

- operation-handler

2.4.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.5. handle

GET /actuator/info

2.5.1. Parameters

Type	Name	Description	Schema
Body	body <i>optional</i>	body	< string, string > map

2.5.2. Responses

HTTP Code	Description	Schema
200	OK	object

2.5.3. Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

2.5.4. Tags

- operation-handler

2.5.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.6. getAllCountries

GET /address/countries

2.6.1. Responses

HTTP Code	Description	Schema
200	OK	< AddressProjection > array

2.6.2. Produces

- */*

2.6.3. Tags

- Address

2.6.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.7. getByCountry

GET /address/country/{countryId}/provinces

2.7.1. Parameters

Type	Name	Description	Schema
Path	countryId <i>required</i>	countryId	integer (int32)

2.7.2. Responses

HTTP Code	Description	Schema
200	OK	< AddressProjection > array

2.7.3. Produces

- */*

2.7.4. Tags

- Address

2.7.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.8. getCitiesByDepartment

GET /address/department/{department_id}/cities

2.8.1. Parameters

Type	Name	Description	Schema
Path	department_id <i>required</i>	department_id	integer (int32)

2.8.2. Responses

HTTP Code	Description	Schema
200	OK	< AddressProjection > array

2.8.3. Produces

- **/**

2.8.4. Tags

- Address

2.8.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.9. getCitiesByProvince

GET /address/province/{provinceId}/cities

2.9.1. Parameters

Type	Name	Description	Schema
Path	provinceId <i>required</i>	provinceId	integer (int32)

2.9.2. Responses

HTTP Code	Description	Schema
200	OK	< AddressProjection > array

2.9.3. Produces

- */*

2.9.4. Tags

- Address

2.9.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.10. getDepartmentsByProvince

GET /address/province/{provinceId}/departments

2.10.1. Parameters

Type	Name	Description	Schema
Path	provinceId <i>required</i>	provinceId	integer (int32)

2.10.2. Responses

HTTP Code	Description	Schema
200	OK	< AddressProjection > array

2.10.3. Produces

- **/**

2.10.4. Tags

- Address

2.10.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.11. login

POST /auth

2.11.1. Parameters

Type	Name	Description	Schema
Body	loginDto <i>required</i>	loginDto	LoginDto

2.11.2. Responses

HTTP Code	Description	Schema
200	OK	JWTTokenDto

2.11.3. Consumes

- [application/json](#)

2.11.4. Produces

- [*/*](#)

2.11.5. Tags

- Authorization

2.11.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.12. create

POST /backoffice/cities

2.12.1. Parameters

Type	Name	Description	Schema
Body	entity <i>required</i>	entity	City

2.12.2. Responses

HTTP Code	Description	Schema
200	OK	City

2.12.3. Consumes

- [application/json](#)

2.12.4. Produces

- **/**

2.12.5. Tags

- backoffice-city-controller

2.12.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.13. getMany

GET /backoffice/cities

2.13.1. Parameters

Type	Name	Description	Schema
Query	ids <i>required</i>	ids	string

2.13.2. Responses

HTTP Code	Description	Schema
200	OK	IterableOfCity

2.13.3. Produces

- **/**

2.13.4. Tags

- backoffice-city-controller

2.13.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.14. getOne

GET /backoffice/cities/{id}

2.14.1. Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int32)

2.14.2. Responses

HTTP Code	Description	Schema
200	OK	City

2.14.3. Produces

- **/**

2.14.4. Tags

- backoffice-city-controller

2.14.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.15. update

PUT /backoffice/cities/{id}

2.15.1. Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int32)

Type	Name	Description	Schema
Body	body <i>required</i>	body	City

2.15.2. Responses

HTTP Code	Description	Schema
200	OK	City

2.15.3. Consumes

- `application/json`

2.15.4. Produces

- `*/*`

2.15.5. Tags

- backoffice-city-controller

2.15.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.16. delete

```
DELETE /backoffice/cities/{id}
```

2.16.1. Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int32)

2.16.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfCity

2.16.3. Produces

- `*/*`

2.16.4. Tags

- backoffice-city-controller

2.16.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.17. create

POST /backoffice/departments

2.17.1. Parameters

Type	Name	Description	Schema
Body	entity <i>required</i>	entity	Department

2.17.2. Responses

HTTP Code	Description	Schema
200	OK	Department

2.17.3. Consumes

- `application/json`

2.17.4. Produces

- `*/*`

2.17.5. Tags

- backoffice-department-controller

2.17.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.18. getList

GET /backoffice/departments

2.18.1. Parameters

Type	Name	Schema
Query	description <i>optional</i>	string
Query	id <i>optional</i>	integer (int32)
Query	page <i>optional</i>	integer (int32)
Query	provinceId <i>optional</i>	integer (int32)
Query	size <i>optional</i>	integer (int32)
Query	sort <i>optional</i>	string

2.18.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfDepartment

2.18.3. Produces

- **/**

2.18.4. Tags

- backoffice-department-controller

2.18.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.19. getOne

GET /backoffice/departments/{id}

2.19.1. Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int32)

2.19.2. Responses

HTTP Code	Description	Schema
200	OK	Department

2.19.3. Produces

- **/**

2.19.4. Tags

- backoffice-department-controller

2.19.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.20. update

PUT /backoffice/departments/{id}

2.20.1. Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int32)
Body	body <i>required</i>	body	Department

2.20.2. Responses

HTTP Code	Description	Schema
200	OK	Department

2.20.3. Consumes

- [application/json](#)

2.20.4. Produces

- [*/*](#)

2.20.5. Tags

- backoffice-department-controller

2.20.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.21. delete

DELETE /backoffice/departments/{id}

2.21.1. Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int32)

2.21.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfDepartment

2.21.3. Produces

- **/**

2.21.4. Tags

- backoffice-department-controller

2.21.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.22. error

POST /error

2.22.1. Responses

HTTP Code	Description	Schema
200	OK	< string, object > map

2.22.2. Consumes

- `application/json`

2.22.3. Produces

- `*/*`

2.22.4. Tags

- `basic-error-controller`

2.22.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.23. error

GET /error

2.23.1. Responses

HTTP Code	Description	Schema
200	OK	< string, object > map

2.23.2. Produces

- `*/*`

2.23.3. Tags

- `basic-error-controller`

2.23.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.24. error

PUT /error

2.24.1. Responses

HTTP Code	Description	Schema
200	OK	< string, object > map

2.24.2. Consumes

- `application/json`

2.24.3. Produces

- `*/*`

2.24.4. Tags

- `basic-error-controller`

2.24.5. Security

Type	Name	Scopes
apiKey	<code>JWT</code>	global

2.25. error

DELETE /error

2.25.1. Responses

HTTP Code	Description	Schema
200	OK	< string, object > map

2.25.2. Produces

- */*

2.25.3. Tags

- basic-error-controller

2.25.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.26. error

PATCH /error

2.26.1. Responses

HTTP Code	Description	Schema
200	OK	< string, object > map

2.26.2. Consumes

- application/json

2.26.3. Produces

- */*

2.26.4. Tags

- basic-error-controller

2.26.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.27. error

HEAD /error

2.27.1. Responses

HTTP Code	Description	Schema
200	OK	< string, object > map

2.27.2. Consumes

- `application/json`

2.27.3. Produces

- `*/*`

2.27.4. Tags

- `basic-error-controller`

2.27.5. Security

Type	Name	Scopes
apiKey	<code>JWT</code>	global

2.28. error

OPTIONS /error

2.28.1. Responses

HTTP Code	Description	Schema
200	OK	< string, object > map

2.28.2. Consumes

- `application/json`

2.28.3. Produces

- `*/*`

2.28.4. Tags

- `basic-error-controller`

2.28.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.29. getLanguages

GET /i18n/support-languages

2.29.1. Responses

HTTP Code	Description	Schema
200	OK	< string > array

2.29.2. Produces

- `*/*`

2.29.3. Tags

- `I18n`

2.29.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.30. updatePassword

PATCH /passwords/{userId}

2.30.1. Parameters

Type	Name	Description	Schema
Path	userId <i>required</i>	userId	integer (int32)
Body	password <i>required</i>	password	string

2.30.2. Responses

HTTP Code	Description	Schema
200	OK	boolean

2.30.3. Consumes

- `application/json`

2.30.4. Produces

- `*/*`

2.30.5. Tags

- Password

2.30.6. Security

Type	Name	Scopes
apiKey	<code>JWT</code>	global

2.31. setRoleUser

POST /roles

2.31.1. Parameters

Type	Name	Description	Schema
Body	userRoleDto <i>required</i>	userRoleDto	RequestUserRoleDto

2.31.2. Responses

HTTP Code	Description	Schema
200	OK	boolean

2.31.3. Consumes

- [application/json](#)

2.31.4. Produces

- [*/*](#)

2.31.5. Tags

- Role

2.31.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.32. addUser

POST /users

2.32.1. Parameters

Type	Name	Description	Schema
Body	userDto <i>required</i>	userDto	AddUserDto

2.32.2. Responses

HTTP Code	Description	Schema
200	OK	UserDto

2.32.3. Consumes

- [application/json](#)

2.32.4. Produces

- [*/*](#)

2.32.5. Tags

- User

2.32.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.33. getUsers

GET /users/page

2.33.1. Parameters

Type	Name	Schema
Query	page <i>optional</i>	integer (int32)
Query	size <i>optional</i>	integer (int32)
Query	sort <i>optional</i>	string

2.33.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfPageableUsers

2.33.3. Produces

- `*/*`

2.33.4. Tags

- User

2.33.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.34. changeStatusAccount

PATCH /users/{id}

2.34.1. Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer (int32)
Query	status <i>required</i>	status	boolean

2.34.2. Responses

HTTP Code	Description	Schema
200	OK	boolean

2.34.3. Consumes

- `application/json`

2.34.4. Produces

- *****/*****

2.34.5. Tags

- User

2.34.6. Security

Type	Name	Scopes
apiKey	JWT	global

Chapter 3. Security

3.1. JWT

Type : apiKey

Name : X-Auth-Token

In : HEADER

Chapter 4. Definitions

4.1. AddUserDto

Name	Schema
email <i>required</i>	string
password <i>required</i>	string

4.2. AddressProjection

Name	Schema
description <i>optional</i>	string
id <i>optional</i>	integer (int32)

4.3. BackofficeDeleteResponseOfCity

Name	Schema
id <i>optional</i>	integer (int32)

4.4. BackofficeDeleteResponseOfDepartment

Name	Schema
id <i>optional</i>	integer (int32)

4.5. Character

Type : object

4.6. City

Name	Schema
active <i>optional</i>	boolean
city_type <i>optional</i>	Character
departmentId <i>optional</i>	integer (int32)
description <i>optional</i>	string
id <i>optional</i>	integer (int32)

4.7. Department

Name	Schema
description <i>optional</i>	string
id <i>optional</i>	integer (int32)
provinceId <i>optional</i>	integer (int32)

4.8. IterableOfCity

Type : object

4.9. IterableOfDepartment

Type : object

4.10. JWTTokenDto

Name	Schema
refreshToken <i>optional</i>	string
token <i>optional</i>	string

4.11. Link

Name	Schema
href <i>optional</i>	string
templated <i>optional</i>	boolean

4.12. LoginDto

Name	Schema
password <i>required</i>	string
username <i>required</i>	string

4.13. MapOfStringAndLink

Type : < string, [Link](#) > map

4.14. ModelAndView

Name	Schema
empty <i>optional</i>	boolean
model <i>optional</i>	object

Name	Schema
modelMap <i>optional</i>	< string, object > map
reference <i>optional</i>	boolean

Name	Schema
status <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, 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)
view <i>optional</i>	View

Name	Schema
viewName <i>optional</i>	string

4.15. PageOfCity

Name	Schema
content <i>optional</i>	< City > array
empty <i>optional</i>	boolean
first <i>optional</i>	boolean
last <i>optional</i>	boolean
number <i>optional</i>	integer (int32)
numberOfElements <i>optional</i>	integer (int32)
pageable <i>optional</i>	Pageable
size <i>optional</i>	integer (int32)
sort <i>optional</i>	Sort
totalElements <i>optional</i>	integer (int64)
totalPages <i>optional</i>	integer (int32)

4.16. PageOfDepartment

Name	Schema
content <i>optional</i>	< Department > array
empty <i>optional</i>	boolean
first <i>optional</i>	boolean
last <i>optional</i>	boolean
number <i>optional</i>	integer (int32)
numberOfElements <i>optional</i>	integer (int32)
pageable <i>optional</i>	Pageable
size <i>optional</i>	integer (int32)
sort <i>optional</i>	Sort
totalElements <i>optional</i>	integer (int64)
totalPages <i>optional</i>	integer (int32)

4.17. PageOfPageableUsers

Name	Schema
content <i>optional</i>	< PageableUsers > array

Name	Schema
empty <i>optional</i>	boolean
first <i>optional</i>	boolean
last <i>optional</i>	boolean
number <i>optional</i>	integer (int32)
numberOfElements <i>optional</i>	integer (int32)
pageable <i>optional</i>	Pageable
size <i>optional</i>	integer (int32)
sort <i>optional</i>	Sort
totalElements <i>optional</i>	integer (int64)
totalPages <i>optional</i>	integer (int32)

4.18. Pageable

Name	Schema
page <i>optional</i>	integer (int32)
size <i>optional</i>	integer (int32)
sort <i>optional</i>	string

4.19. PageableUsers

Name	Schema
enable <i>optional</i>	boolean
id <i>optional</i>	integer (int32)
lastLogin <i>optional</i>	string (date-time)
role <i>optional</i>	string
username <i>optional</i>	string

4.20. RequestUserRoleDto

Name	Schema
roleId <i>optional</i>	integer (int32)
userId <i>required</i>	integer (int32)

4.21. Sort

Name	Schema
empty <i>optional</i>	boolean
sorted <i>optional</i>	boolean
unsorted <i>optional</i>	boolean

4.22. UserDto

Name	Schema
email <i>required</i>	string
id <i>optional</i>	integer (int32)

4.23. View

Name	Schema
contentType <i>optional</i>	string