

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	3
2.1. links	3
2.1.1. Responses	3
2.1.2. Produces	3
2.1.3. Tags	3
2.1.4. Security	3
2.2. handle	3
2.2.1. Parameters	3
2.2.2. Responses	4
2.2.3. Produces	4
2.2.4. Tags	4
2.2.5. Security	4
2.3. handle	4
2.3.1. Parameters	4
2.3.2. Responses	4
2.3.3. Produces	4
2.3.4. Tags	5
2.3.5. Security	5
2.4. handle	5
2.4.1. Parameters	5
2.4.2. Responses	5
2.4.3. Produces	5
2.4.4. Tags	5
2.4.5. Security	5
2.5. getAllCountries	6
2.5.1. Responses	6
2.5.2. Produces	6
2.5.3. Tags	6
2.5.4. Security	6
2.6. getByCountry	6
2.6.1. Parameters	6
2.6.2. Responses	7
2.6.3. Produces	7

2.6.4. Tags	7
2.6.5. Security	7
2.7. getCitiesByDepartment	7
2.7.1. Parameters	7
2.7.2. Responses	7
2.7.3. Produces	8
2.7.4. Tags	8
2.7.5. Security	8
2.8. getCitiesByProvince	8
2.8.1. Parameters	8
2.8.2. Responses	8
2.8.3. Produces	8
2.8.4. Tags	8
2.8.5. Security	8
2.9. getDepartmentsByProvince	9
2.9.1. Parameters	9
2.9.2. Responses	9
2.9.3. Produces	9
2.9.4. Tags	9
2.9.5. Security	9
2.10. login	9
2.10.1. Parameters	10
2.10.2. Responses	10
2.10.3. Consumes	10
2.10.4. Produces	10
2.10.5. Tags	10
2.10.6. Security	10
2.11. create	10
2.11.1. Parameters	10
2.11.2. Responses	11
2.11.3. Consumes	11
2.11.4. Produces	11
2.11.5. Tags	11
2.11.6. Security	11
2.12. getMany	11
2.12.1. Parameters	11
2.12.2. Responses	11
2.12.3. Produces	12
2.12.4. Tags	12
2.12.5. Security	12
2.13. getOne	12

2.13.1. Parameters	12
2.13.2. Responses	12
2.13.3. Produces	12
2.13.4. Tags	12
2.13.5. Security	12
2.14. update	13
2.14.1. Parameters	13
2.14.2. Responses	13
2.14.3. Consumes	13
2.14.4. Produces	13
2.14.5. Tags	13
2.14.6. Security	13
2.15. delete	13
2.15.1. Parameters	14
2.15.2. Responses	14
2.15.3. Produces	14
2.15.4. Tags	14
2.15.5. Security	14
2.16. create	14
2.16.1. Parameters	14
2.16.2. Responses	14
2.16.3. Consumes	15
2.16.4. Produces	15
2.16.5. Tags	15
2.16.6. Security	15
2.17. getList	15
2.17.1. Parameters	15
2.17.2. Responses	16
2.17.3. Produces	16
2.17.4. Tags	16
2.17.5. Security	16
2.18. getOne	16
2.18.1. Parameters	16
2.18.2. Responses	16
2.18.3. Produces	17
2.18.4. Tags	17
2.18.5. Security	17
2.19. update	17
2.19.1. Parameters	17
2.19.2. Responses	17
2.19.3. Consumes	17

2.19.4. Produces	17
2.19.5. Tags	18
2.19.6. Security	18
2.20. delete	18
2.20.1. Parameters	18
2.20.2. Responses	18
2.20.3. Produces	18
2.20.4. Tags	18
2.20.5. Security	18
2.21. create	19
2.21.1. Parameters	19
2.21.2. Responses	19
2.21.3. Consumes	19
2.21.4. Produces	19
2.21.5. Tags	19
2.21.6. Security	19
2.22. getList	19
2.22.1. Parameters	19
2.22.2. Responses	20
2.22.3. Produces	20
2.22.4. Tags	20
2.22.5. Security	20
2.23. getOne	21
2.23.1. Parameters	21
2.23.2. Responses	21
2.23.3. Produces	21
2.23.4. Tags	21
2.23.5. Security	21
2.24. update	21
2.24.1. Parameters	21
2.24.2. Responses	22
2.24.3. Consumes	22
2.24.4. Produces	22
2.24.5. Tags	22
2.24.6. Security	22
2.25. delete	22
2.25.1. Parameters	22
2.25.2. Responses	23
2.25.3. Produces	23
2.25.4. Tags	23
2.25.5. Security	23

2.26. create	23
2.26.1. Parameters	23
2.26.2. Responses	23
2.26.3. Consumes	23
2.26.4. Produces	24
2.26.5. Tags	24
2.26.6. Security	24
2.27. getList	24
2.27.1. Parameters	24
2.27.2. Responses	25
2.27.3. Produces	25
2.27.4. Tags	25
2.27.5. Security	25
2.28. getOne	25
2.28.1. Parameters	25
2.28.2. Responses	25
2.28.3. Produces	25
2.28.4. Tags	26
2.28.5. Security	26
2.29. update	26
2.29.1. Parameters	26
2.29.2. Responses	26
2.29.3. Consumes	26
2.29.4. Produces	26
2.29.5. Tags	26
2.29.6. Security	27
2.30. delete	27
2.30.1. Parameters	27
2.30.2. Responses	27
2.30.3. Produces	27
2.30.4. Tags	27
2.30.5. Security	27
2.31. create	27
2.31.1. Parameters	28
2.31.2. Responses	28
2.31.3. Consumes	28
2.31.4. Produces	28
2.31.5. Tags	28
2.31.6. Security	28
2.32. getMany	28
2.32.1. Parameters	28

2.32.2. Responses	29
2.32.3. Produces	29
2.32.4. Tags	29
2.32.5. Security	29
2.33. getOne	29
2.33.1. Parameters	29
2.33.2. Responses	29
2.33.3. Produces	29
2.33.4. Tags	30
2.33.5. Security	30
2.34. update	30
2.34.1. Parameters	30
2.34.2. Responses	30
2.34.3. Consumes	30
2.34.4. Produces	30
2.34.5. Tags	30
2.34.6. Security	31
2.35. delete	31
2.35.1. Parameters	31
2.35.2. Responses	31
2.35.3. Produces	31
2.35.4. Tags	31
2.35.5. Security	31
2.36. create	31
2.36.1. Parameters	32
2.36.2. Responses	32
2.36.3. Consumes	32
2.36.4. Produces	32
2.36.5. Tags	32
2.36.6. Security	32
2.37. getMany	32
2.37.1. Parameters	32
2.37.2. Responses	33
2.37.3. Produces	33
2.37.4. Tags	33
2.37.5. Security	33
2.38. getOne	33
2.38.1. Parameters	33
2.38.2. Responses	33
2.38.3. Produces	33
2.38.4. Tags	34

2.38.5. Security	34
2.39. update	34
2.39.1. Parameters	34
2.39.2. Responses	34
2.39.3. Consumes	34
2.39.4. Produces	34
2.39.5. Tags	34
2.39.6. Security	35
2.40. delete	35
2.40.1. Parameters	35
2.40.2. Responses	35
2.40.3. Produces	35
2.40.4. Tags	35
2.40.5. Security	35
2.41. create	35
2.41.1. Parameters	36
2.41.2. Responses	36
2.41.3. Consumes	36
2.41.4. Produces	36
2.41.5. Tags	36
2.41.6. Security	36
2.42. getMany	36
2.42.1. Parameters	36
2.42.2. Responses	37
2.42.3. Produces	37
2.42.4. Tags	37
2.42.5. Security	37
2.43. getOne	37
2.43.1. Parameters	37
2.43.2. Responses	37
2.43.3. Produces	37
2.43.4. Tags	38
2.43.5. Security	38
2.44. update	38
2.44.1. Parameters	38
2.44.2. Responses	38
2.44.3. Consumes	38
2.44.4. Produces	38
2.44.5. Tags	38
2.44.6. Security	39
2.45. delete	39

2.45.1. Parameters	39
2.45.2. Responses	39
2.45.3. Produces	39
2.45.4. Tags	39
2.45.5. Security	39
2.46. create	39
2.46.1. Parameters	40
2.46.2. Responses	40
2.46.3. Consumes	40
2.46.4. Produces	40
2.46.5. Tags	40
2.46.6. Security	40
2.47. getList	40
2.47.1. Parameters	40
2.47.2. Responses	41
2.47.3. Produces	41
2.47.4. Tags	41
2.47.5. Security	41
2.48. getOne	41
2.48.1. Parameters	41
2.48.2. Responses	42
2.48.3. Produces	42
2.48.4. Tags	42
2.48.5. Security	42
2.49. update	42
2.49.1. Parameters	42
2.49.2. Responses	42
2.49.3. Consumes	43
2.49.4. Produces	43
2.49.5. Tags	43
2.49.6. Security	43
2.50. delete	43
2.50.1. Parameters	43
2.50.2. Responses	43
2.50.3. Produces	44
2.50.4. Tags	44
2.50.5. Security	44
2.51. create	44
2.51.1. Parameters	44
2.51.2. Responses	44
2.51.3. Consumes	44

2.51.4. Produces	44
2.51.5. Tags	44
2.51.6. Security	45
2.52. getList	45
2.52.1. Parameters	45
2.52.2. Responses	45
2.52.3. Produces	46
2.52.4. Tags	46
2.52.5. Security	46
2.53. getOne	46
2.53.1. Parameters	46
2.53.2. Responses	46
2.53.3. Produces	46
2.53.4. Tags	46
2.53.5. Security	47
2.54. update	47
2.54.1. Parameters	47
2.54.2. Responses	47
2.54.3. Consumes	47
2.54.4. Produces	47
2.54.5. Tags	47
2.54.6. Security	47
2.55. delete	48
2.55.1. Parameters	48
2.55.2. Responses	48
2.55.3. Produces	48
2.55.4. Tags	48
2.55.5. Security	48
2.56. create	48
2.56.1. Parameters	48
2.56.2. Responses	49
2.56.3. Consumes	49
2.56.4. Produces	49
2.56.5. Tags	49
2.56.6. Security	49
2.57. getMany	49
2.57.1. Parameters	49
2.57.2. Responses	49
2.57.3. Produces	50
2.57.4. Tags	50
2.57.5. Security	50

2.58. getOne	50
2.58.1. Parameters	50
2.58.2. Responses	50
2.58.3. Produces	50
2.58.4. Tags	50
2.58.5. Security	50
2.59. update	51
2.59.1. Parameters	51
2.59.2. Responses	51
2.59.3. Consumes	51
2.59.4. Produces	51
2.59.5. Tags	51
2.59.6. Security	51
2.60. delete	51
2.60.1. Parameters	52
2.60.2. Responses	52
2.60.3. Produces	52
2.60.4. Tags	52
2.60.5. Security	52
2.61. errorHtml	52
2.61.1. Responses	52
2.61.2. Consumes	52
2.61.3. Produces	53
2.61.4. Tags	53
2.61.5. Security	53
2.62. errorHtml	53
2.62.1. Responses	53
2.62.2. Produces	53
2.62.3. Tags	53
2.62.4. Security	53
2.63. errorHtml	53
2.63.1. Responses	54
2.63.2. Consumes	54
2.63.3. Produces	54
2.63.4. Tags	54
2.63.5. Security	54
2.64. errorHtml	54
2.64.1. Responses	54
2.64.2. Produces	54
2.64.3. Tags	54
2.64.4. Security	55

2.65. errorHtml	55
2.65.1. Responses	55
2.65.2. Consumes	55
2.65.3. Produces	55
2.65.4. Tags	55
2.65.5. Security	55
2.66. errorHtml	55
2.66.1. Responses	55
2.66.2. Consumes	56
2.66.3. Produces	56
2.66.4. Tags	56
2.66.5. Security	56
2.67. errorHtml	56
2.67.1. Responses	56
2.67.2. Consumes	56
2.67.3. Produces	56
2.67.4. Tags	56
2.67.5. Security	57
2.68. getLanguages	57
2.68.1. Responses	57
2.68.2. Produces	57
2.68.3. Tags	57
2.68.4. Security	57
2.69. getAllInternmentPatient	57
2.69.1. Parameters	57
2.69.2. Responses	58
2.69.3. Produces	58
2.69.4. Tags	58
2.69.5. Security	58
2.70. getIdentityVerificationStatus	58
2.70.1. Responses	58
2.70.2. Consumes	58
2.70.3. Produces	58
2.70.4. Tags	59
2.70.5. Security	59
2.71. getIdentityVerificationStatus	59
2.71.1. Responses	59
2.71.2. Produces	59
2.71.3. Tags	59
2.71.4. Security	59
2.72. getIdentityVerificationStatus	59

2.72.1. Responses	60
2.72.2. Consumes	60
2.72.3. Produces	60
2.72.4. Tags	60
2.72.5. Security	60
2.73. getIdentityVerificationStatus	60
2.73.1. Responses	60
2.73.2. Produces	60
2.73.3. Tags	61
2.73.4. Security	61
2.74. getIdentityVerificationStatus	61
2.74.1. Responses	61
2.74.2. Consumes	61
2.74.3. Produces	61
2.74.4. Tags	61
2.74.5. Security	61
2.75. getIdentityVerificationStatus	61
2.75.1. Responses	62
2.75.2. Consumes	62
2.75.3. Produces	62
2.75.4. Tags	62
2.75.5. Security	62
2.76. getIdentityVerificationStatus	62
2.76.1. Responses	62
2.76.2. Consumes	62
2.76.3. Produces	63
2.76.4. Tags	63
2.76.5. Security	63
2.77. updatePassword	63
2.77.1. Parameters	63
2.77.2. Responses	63
2.77.3. Consumes	63
2.77.4. Produces	63
2.77.5. Tags	64
2.77.6. Security	64
2.78. addPatient	64
2.78.1. Parameters	64
2.78.2. Responses	64
2.78.3. Consumes	64
2.78.4. Produces	64
2.78.5. Tags	64

2.78.6. Security	64
2.79. getPatientMinimal	65
2.79.1. Parameters	65
2.79.2. Responses	65
2.79.3. Produces	65
2.79.4. Tags	65
2.79.5. Security	65
2.80. searchPatient	65
2.80.1. Parameters	66
2.80.2. Responses	66
2.80.3. Produces	66
2.80.4. Tags	66
2.80.5. Security	66
2.81. getBasicDataPatient	66
2.81.1. Parameters	66
2.81.2. Responses	67
2.81.3. Produces	67
2.81.4. Tags	67
2.81.5. Security	67
2.82. addPerson	67
2.82.1. Parameters	67
2.82.2. Responses	67
2.82.3. Consumes	67
2.82.4. Produces	68
2.82.5. Tags	68
2.82.6. Security	68
2.83. getPersonMinimal	68
2.83.1. Parameters	68
2.83.2. Responses	68
2.83.3. Produces	68
2.83.4. Tags	69
2.83.5. Security	69
2.84. getPersonalInformation	69
2.84.1. Parameters	69
2.84.2. Responses	69
2.84.3. Produces	69
2.84.4. Tags	69
2.84.5. Security	69
2.85. setRoleUser	70
2.85.1. Parameters	70
2.85.2. Responses	70

2.85.3. Consumes	70
2.85.4. Produces	70
2.85.5. Tags	70
2.85.6. Security	70
2.86. addUser	70
2.86.1. Parameters	70
2.86.2. Responses	71
2.86.3. Consumes	71
2.86.4. Produces	71
2.86.5. Tags	71
2.86.6. Security	71
2.87. getUsers	71
2.87.1. Parameters	71
2.87.2. Responses	72
2.87.3. Produces	72
2.87.4. Tags	72
2.87.5. Security	72
2.88. changeStatusAccount	72
2.88.1. Parameters	72
2.88.2. Responses	72
2.88.3. Consumes	73
2.88.4. Produces	73
2.88.5. Tags	73
2.88.6. Security	73
3. Security	74
3.1. JWT	74
4. Definitions	75
4.1. APatientDto	75
4.2. APersonDto	76
4.3. AddUserDto	78
4.4. Address	79
4.5. AddressDto	79
4.6. AddressProjection	80
4.7. BMPatientDto	80
4.8. BMPersonDto	82
4.9. BackofficeDeleteResponseOfAddress	84
4.10. BackofficeDeleteResponseOfBed	84
4.11. BackofficeDeleteResponseOfCity	84
4.12. BackofficeDeleteResponseOfClinicalSpecialty	85
4.13. BackofficeDeleteResponseOfClinicalSpecialtySector	85
4.14. BackofficeDeleteResponseOfDepartment	85

4.15. BackofficeDeleteResponseOfInstitution	85
4.16. BackofficeDeleteResponseOfProvince	85
4.17. BackofficeDeleteResponseOfRoom	85
4.18. BackofficeDeleteResponseOfSector	86
4.19. BasicDataPersonDto	86
4.20. BasicPatientDto	86
4.21. Bed	87
4.22. BedDto	87
4.23. Character	87
4.24. City	87
4.25. CityDto	88
4.26. ClinicalSpecialty	88
4.27. ClinicalSpecialtyDto	88
4.28. ClinicalSpecialtySector	89
4.29. Department	89
4.30. HealthInsuranceDto	89
4.31. IdentificationTypeDto	90
4.32. IdentityVerificationStatus	90
4.33. Institution	90
4.34. InternmentEpisodeDto	91
4.35. IterableOfAddress	91
4.36. IterableOfBed	91
4.37. IterableOfCity	91
4.38. IterableOfClinicalSpecialty	91
4.39. IterableOfClinicalSpecialtySector	91
4.40. IterableOfDepartment	92
4.41. IterableOfInstitution	92
4.42. IterableOfProvince	92
4.43. IterableOfRoom	92
4.44. IterableOfSector	92
4.45. JWTTokenDto	92
4.46. Link	92
4.47. LoginDto	92
4.48. MapOfstringAndLink	93
4.49. ModelAndView	93
4.50. PageOfAddress	95
4.51. PageOfBed	96
4.52. PageOfCity	96
4.53. PageOfClinicalSpecialty	97
4.54. PageOfClinicalSpecialtySector	98
4.55. PageOfDepartment	99

4.56. PageOfInstitution	100
4.57. PageOfPageableUsers	101
4.58. PageOfProvince	102
4.59. PageOfRoom	102
4.60. PageOfSector	103
4.61. Pageable	104
4.62. PageableUsers	104
4.63. PatientDto	105
4.64. PatientSearchDto	105
4.65. PersonalInformationDto	105
4.66. Province	106
4.67. ProvinceDto	107
4.68. RequestUserRoleDto	107
4.69. ResponsibleDoctorDto	107
4.70. Room	107
4.71. RoomDto	108
4.72. Sector	108
4.73. SectorDto	108
4.74. Sort	109
4.75. UserDto	109
4.76. View	109

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 Master Data : Address Master Data Controller
- Authorization : Authentication Controller
- I18n : I 18 N Controller
- Internment Patient : Internment Patient Controller
- Password : User Password Controller
- Patient : Patient Controller
- PatientMasterData : Patient Master Data Controller
- Person : Person Controller
- Role : Role Controller
- User : User Controller
- backoffice-address-controller : Backoffice Address Controller
- backoffice-bed-controller : Backoffice Bed Controller
- backoffice-city-controller : Backoffice City Controller
- backoffice-clinical-specialty-controller : Backoffice Clinical Specialty Controller
- backoffice-clinical-specialty-sector-controller : Backoffice Clinical Specialty Sector Controller
- backoffice-department-controller : Backoffice Department Controller
- backoffice-institution-controller : Backoffice Institution Controller
- backoffice-province-controller : Backoffice Province Controller

- backoffice-room-controller : Backoffice Room Controller
- backoffice-sector-controller : Backoffice Sector 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/vnd.spring-boot.actuator.v3+json`
- `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/vnd.spring-boot.actuator.v3+json`
- `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/**
```

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/vnd.spring-boot.actuator.v3+json`

- `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/info

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/vnd.spring-boot.actuator.v3+json`
- `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. getAllCountries

GET /address/masterdata/countries

2.5.1. Responses

HTTP Code	Description	Schema
200	OK	< AddressProjection > array

2.5.2. Produces

- */*

2.5.3. Tags

- Address Master Data

2.5.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.6. getByCountry

GET /address/masterdata/country/{countryId}/provinces

2.6.1. Parameters

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

2.6.2. Responses

HTTP Code	Description	Schema
200	OK	< AddressProjection > array

2.6.3. Produces

- **/**

2.6.4. Tags

- Address Master Data

2.6.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.7. getCitiesByDepartment

GET /address/masterdata/department/{department_id}/cities

2.7.1. Parameters

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

2.7.2. Responses

HTTP Code	Description	Schema
200	OK	< AddressProjection > array

2.7.3. Produces

- */*

2.7.4. Tags

- Address Master Data

2.7.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.8. getCitiesByProvince

GET /address/masterdata/province/{provinceId}/cities

2.8.1. Parameters

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

2.8.2. Responses

HTTP Code	Description	Schema
200	OK	< AddressProjection > array

2.8.3. Produces

- */*

2.8.4. Tags

- Address Master Data

2.8.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.9. getDepartmentsByProvince

GET /address/masterdata/province/{provinceId}/departments

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 Master Data

2.9.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.10. login

POST /auth

2.10.1. Parameters

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

2.10.2. Responses

HTTP Code	Description	Schema
200	OK	JWTTokenDto

2.10.3. Consumes

- [application/json](#)

2.10.4. Produces

- [*/*](#)

2.10.5. Tags

- Authorization

2.10.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.11. create

POST /backoffice/addresses

2.11.1. Parameters

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

2.11.2. Responses

HTTP Code	Description	Schema
200	OK	Address

2.11.3. Consumes

- [application/json](#)

2.11.4. Produces

- [*/*](#)

2.11.5. Tags

- backoffice-address-controller

2.11.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.12. getMany

GET /backoffice/addresses

2.12.1. Parameters

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

2.12.2. Responses

HTTP Code	Description	Schema
200	OK	IterableOfAddress

2.12.3. Produces

- **/**

2.12.4. Tags

- backoffice-address-controller

2.12.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.13. getOne

GET /backoffice/addresses/{id}

2.13.1. Parameters

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

2.13.2. Responses

HTTP Code	Description	Schema
200	OK	Address

2.13.3. Produces

- **/**

2.13.4. Tags

- backoffice-address-controller

2.13.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.14. update

PUT /backoffice/addresses/{id}

2.14.1. Parameters

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

2.14.2. Responses

HTTP Code	Description	Schema
200	OK	Address

2.14.3. Consumes

- [application/json](#)

2.14.4. Produces

- [*/*](#)

2.14.5. Tags

- backoffice-address-controller

2.14.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.15. delete

DELETE /backoffice/addresses/{id}

2.15.1. Parameters

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

2.15.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfAddress

2.15.3. Produces

- **/**

2.15.4. Tags

- backoffice-address-controller

2.15.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.16. create

POST /backoffice/beds

2.16.1. Parameters

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

2.16.2. Responses

HTTP Code	Description	Schema
200	OK	Bed

2.16.3. Consumes

- `application/json`

2.16.4. Produces

- `*/*`

2.16.5. Tags

- backoffice-bed-controller

2.16.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.17. getList

GET /backoffice/beds

2.17.1. Parameters

Type	Name	Schema
Query	bedCategoryId <i>optional</i>	integer (int32)
Query	id <i>optional</i>	integer (int32)
Query	page <i>optional</i>	integer (int32)
Query	roomId <i>optional</i>	integer (int32)
Query	size <i>optional</i>	integer (int32)

Type	Name	Schema
Query	sort <i>optional</i>	string

2.17.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfBed

2.17.3. Produces

- **/**

2.17.4. Tags

- backoffice-bed-controller

2.17.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.18. getOne

```
GET /backoffice/beds/{id}
```

2.18.1. Parameters

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

2.18.2. Responses

HTTP Code	Description	Schema
200	OK	Bed

2.18.3. Produces

- **/**

2.18.4. Tags

- backoffice-bed-controller

2.18.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.19. update

PUT /backoffice/beds/{id}

2.19.1. Parameters

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

2.19.2. Responses

HTTP Code	Description	Schema
200	OK	Bed

2.19.3. Consumes

- *application/json*

2.19.4. Produces

- **/**

2.19.5. Tags

- backoffice-bed-controller

2.19.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.20. delete

```
DELETE /backoffice/beds/{id}
```

2.20.1. Parameters

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

2.20.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfBed

2.20.3. Produces

- **/**

2.20.4. Tags

- backoffice-bed-controller

2.20.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.21. create

POST /backoffice/cities

2.21.1. Parameters

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

2.21.2. Responses

HTTP Code	Description	Schema
200	OK	City

2.21.3. Consumes

- [application/json](#)

2.21.4. Produces

- [*/*](#)

2.21.5. Tags

- backoffice-city-controller

2.21.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.22. getList

GET /backoffice/cities

2.22.1. Parameters

Type	Name	Schema
Query	active <i>optional</i>	boolean
Query	city_type <i>optional</i>	ref
Query	departmentId <i>optional</i>	integer (int32)
Query	description <i>optional</i>	string
Query	id <i>optional</i>	integer (int32)
Query	page <i>optional</i>	integer (int32)
Query	size <i>optional</i>	integer (int32)
Query	sort <i>optional</i>	string

2.22.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfCity

2.22.3. Produces

- **/**

2.22.4. Tags

- backoffice-city-controller

2.22.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.23. getOne

GET /backoffice/cities/{id}

2.23.1. Parameters

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

2.23.2. Responses

HTTP Code	Description	Schema
200	OK	City

2.23.3. Produces

- */*

2.23.4. Tags

- backoffice-city-controller

2.23.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.24. update

PUT /backoffice/cities/{id}

2.24.1. Parameters

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

2.24.2. Responses

HTTP Code	Description	Schema
200	OK	City

2.24.3. Consumes

- [application/json](#)

2.24.4. Produces

- [*/*](#)

2.24.5. Tags

- backoffice-city-controller

2.24.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.25. delete

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

2.25.1. Parameters

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

2.25.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfCity

2.25.3. Produces

- `*/*`

2.25.4. Tags

- backoffice-city-controller

2.25.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.26. create

POST /backoffice/clinicalspecialties

2.26.1. Parameters

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

2.26.2. Responses

HTTP Code	Description	Schema
200	OK	ClinicalSpecialty

2.26.3. Consumes

- `application/json`

2.26.4. Produces

- ***/***

2.26.5. Tags

- backoffice-clinical-specialty-controller

2.26.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.27. getList

GET /backoffice/clinicalspecialties

2.27.1. Parameters

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

2.27.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfClinicalSpecialty

2.27.3. Produces

- **/**

2.27.4. Tags

- backoffice-clinical-specialty-controller

2.27.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.28. getOne

```
GET /backoffice/clinicalspecialties/{id}
```

2.28.1. Parameters

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

2.28.2. Responses

HTTP Code	Description	Schema
200	OK	ClinicalSpecialty

2.28.3. Produces

- **/**

2.28.4. Tags

- backoffice-clinical-specialty-controller

2.28.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.29. update

```
PUT /backoffice/clinicalspecialties/{id}
```

2.29.1. Parameters

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

2.29.2. Responses

HTTP Code	Description	Schema
200	OK	ClinicalSpecialty

2.29.3. Consumes

- [application/json](#)

2.29.4. Produces

- [**](#)

2.29.5. Tags

- backoffice-clinical-specialty-controller

2.29.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.30. delete

DELETE /backoffice/clinicalspecialties/{id}

2.30.1. Parameters

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

2.30.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfClinicalSpecialty

2.30.3. Produces

- [*/*](#)

2.30.4. Tags

- backoffice-clinical-specialty-controller

2.30.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.31. create

POST /backoffice/clinicalspecialtysectors

2.31.1. Parameters

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

2.31.2. Responses

HTTP Code	Description	Schema
200	OK	ClinicalSpecialtySector

2.31.3. Consumes

- [application/json](#)

2.31.4. Produces

- [*/*](#)

2.31.5. Tags

- backoffice-clinical-specialty-sector-controller

2.31.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.32. getMany

GET /backoffice/clinicalspecialtysectors

2.32.1. Parameters

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

2.32.2. Responses

HTTP Code	Description	Schema
200	OK	IterableOfClinicalSpecialtySector

2.32.3. Produces

- [*/*](#)

2.32.4. Tags

- backoffice-clinical-specialty-sector-controller

2.32.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.33. getOne

```
GET /backoffice/clinicalspecialtysectors/{id}
```

2.33.1. Parameters

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

2.33.2. Responses

HTTP Code	Description	Schema
200	OK	ClinicalSpecialtySector

2.33.3. Produces

- [*/*](#)

2.33.4. Tags

- backoffice-clinical-specialty-sector-controller

2.33.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.34. update

```
PUT /backoffice/clinicalspecialtysectors/{id}
```

2.34.1. Parameters

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

2.34.2. Responses

HTTP Code	Description	Schema
200	OK	ClinicalSpecialtySector

2.34.3. Consumes

- [application/json](#)

2.34.4. Produces

- [*/*](#)

2.34.5. Tags

- backoffice-clinical-specialty-sector-controller

2.34.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.35. delete

DELETE /backoffice/clinicalspecialtysectors/{id}

2.35.1. Parameters

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

2.35.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfClinicalSpecialtySector

2.35.3. Produces

- [*/*](#)

2.35.4. Tags

- backoffice-clinical-specialty-sector-controller

2.35.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.36. create

POST /backoffice/departments

2.36.1. Parameters

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

2.36.2. Responses

HTTP Code	Description	Schema
200	OK	Department

2.36.3. Consumes

- [application/json](#)

2.36.4. Produces

- [*/*](#)

2.36.5. Tags

- backoffice-department-controller

2.36.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.37. getMany

GET /backoffice/departments

2.37.1. Parameters

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

2.37.2. Responses

HTTP Code	Description	Schema
200	OK	IterableOfDepartment

2.37.3. Produces

- [*/*](#)

2.37.4. Tags

- backoffice-department-controller

2.37.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.38. getOne

GET /backoffice/departments/{id}

2.38.1. Parameters

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

2.38.2. Responses

HTTP Code	Description	Schema
200	OK	Department

2.38.3. Produces

- [*/*](#)

2.38.4. Tags

- backoffice-department-controller

2.38.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.39. update

```
PUT /backoffice/departments/{id}
```

2.39.1. Parameters

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

2.39.2. Responses

HTTP Code	Description	Schema
200	OK	Department

2.39.3. Consumes

- [application/json](#)

2.39.4. Produces

- [**](#)

2.39.5. Tags

- backoffice-department-controller

2.39.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.40. delete

DELETE /backoffice/departments/{id}

2.40.1. Parameters

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

2.40.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfDepartment

2.40.3. Produces

- [*/*](#)

2.40.4. Tags

- backoffice-department-controller

2.40.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.41. create

POST /backoffice/institutions

2.41.1. Parameters

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

2.41.2. Responses

HTTP Code	Description	Schema
200	OK	Institution

2.41.3. Consumes

- [application/json](#)

2.41.4. Produces

- [*/*](#)

2.41.5. Tags

- backoffice-institution-controller

2.41.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.42. getMany

GET /backoffice/institutions

2.42.1. Parameters

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

2.42.2. Responses

HTTP Code	Description	Schema
200	OK	IterableOfInstitution

2.42.3. Produces

- [*/*](#)

2.42.4. Tags

- backoffice-institution-controller

2.42.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.43. getOne

```
GET /backoffice/institutions/{id}
```

2.43.1. Parameters

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

2.43.2. Responses

HTTP Code	Description	Schema
200	OK	Institution

2.43.3. Produces

- [*/*](#)

2.43.4. Tags

- backoffice-institution-controller

2.43.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.44. update

PUT /backoffice/institutions/{id}

2.44.1. Parameters

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

2.44.2. Responses

HTTP Code	Description	Schema
200	OK	Institution

2.44.3. Consumes

- [application/json](#)

2.44.4. Produces

- [**](#)

2.44.5. Tags

- backoffice-institution-controller

2.44.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.45. delete

DELETE /backoffice/institutions/{id}

2.45.1. Parameters

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

2.45.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfInstitution

2.45.3. Produces

- **/**

2.45.4. Tags

- backoffice-institution-controller

2.45.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.46. create

POST /backoffice/provinces

2.46.1. Parameters

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

2.46.2. Responses

HTTP Code	Description	Schema
200	OK	Province

2.46.3. Consumes

- [application/json](#)

2.46.4. Produces

- [*/*](#)

2.46.5. Tags

- backoffice-province-controller

2.46.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.47. getList

GET /backoffice/provinces

2.47.1. Parameters

Type	Name	Schema
Query	countryId <i>optional</i>	integer (int32)
Query	description <i>optional</i>	string

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

2.47.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfProvince

2.47.3. Produces

- **/**

2.47.4. Tags

- backoffice-province-controller

2.47.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.48. getOne

```
GET /backoffice/provinces/{id}
```

2.48.1. Parameters

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

2.48.2. Responses

HTTP Code	Description	Schema
200	OK	Province

2.48.3. Produces

- **/**

2.48.4. Tags

- backoffice-province-controller

2.48.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.49. update

```
PUT /backoffice/provinces/{id}
```

2.49.1. Parameters

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

2.49.2. Responses

HTTP Code	Description	Schema
200	OK	Province

2.49.3. Consumes

- [application/json](#)

2.49.4. Produces

- [*/*](#)

2.49.5. Tags

- backoffice-province-controller

2.49.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.50. delete

```
DELETE /backoffice/provinces/{id}
```

2.50.1. Parameters

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

2.50.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfProvince

2.50.3. Produces

- `*/*`

2.50.4. Tags

- backoffice-province-controller

2.50.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.51. create

POST /backoffice/rooms

2.51.1. Parameters

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

2.51.2. Responses

HTTP Code	Description	Schema
200	OK	Room

2.51.3. Consumes

- `application/json`

2.51.4. Produces

- `*/*`

2.51.5. Tags

- backoffice-room-controller

2.51.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.52. getList

GET /backoffice/rooms

2.52.1. Parameters

Type	Name	Schema
Query	description <i>optional</i>	string
Query	dischargeDate <i>optional</i>	string (date)
Query	id <i>optional</i>	integer (int32)
Query	page <i>optional</i>	integer (int32)
Query	sectorId <i>optional</i>	integer (int32)
Query	size <i>optional</i>	integer (int32)
Query	sort <i>optional</i>	string
Query	type <i>optional</i>	string

2.52.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfRoom

2.52.3. Produces

- [*/*](#)

2.52.4. Tags

- backoffice-room-controller

2.52.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.53. getOne

```
GET /backoffice/rooms/{id}
```

2.53.1. Parameters

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

2.53.2. Responses

HTTP Code	Description	Schema
200	OK	Room

2.53.3. Produces

- [*/*](#)

2.53.4. Tags

- backoffice-room-controller

2.53.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.54. update

PUT /backoffice/rooms/{id}

2.54.1. Parameters

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

2.54.2. Responses

HTTP Code	Description	Schema
200	OK	Room

2.54.3. Consumes

- [application/json](#)

2.54.4. Produces

- [*/*](#)

2.54.5. Tags

- backoffice-room-controller

2.54.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.55. delete

DELETE /backoffice/rooms/{id}

2.55.1. Parameters

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

2.55.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfRoom

2.55.3. Produces

- */*

2.55.4. Tags

- backoffice-room-controller

2.55.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.56. create

POST /backoffice/sectors

2.56.1. Parameters

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

2.56.2. Responses

HTTP Code	Description	Schema
200	OK	Sector

2.56.3. Consumes

- [application/json](#)

2.56.4. Produces

- [*/*](#)

2.56.5. Tags

- backoffice-sector-controller

2.56.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.57. getMany

GET /backoffice/sectors

2.57.1. Parameters

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

2.57.2. Responses

HTTP Code	Description	Schema
200	OK	IterableOfSector

2.57.3. Produces

- **/**

2.57.4. Tags

- backoffice-sector-controller

2.57.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.58. getOne

GET /backoffice/sectors/{id}

2.58.1. Parameters

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

2.58.2. Responses

HTTP Code	Description	Schema
200	OK	Sector

2.58.3. Produces

- **/**

2.58.4. Tags

- backoffice-sector-controller

2.58.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.59. update

PUT /backoffice/sectors/{id}

2.59.1. Parameters

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

2.59.2. Responses

HTTP Code	Description	Schema
200	OK	Sector

2.59.3. Consumes

- [application/json](#)

2.59.4. Produces

- [*/*](#)

2.59.5. Tags

- backoffice-sector-controller

2.59.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.60. delete

DELETE /backoffice/sectors/{id}

2.60.1. Parameters

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

2.60.2. Responses

HTTP Code	Description	Schema
200	OK	BackofficeDeleteResponseOfSector

2.60.3. Produces

- **/**

2.60.4. Tags

- backoffice-sector-controller

2.60.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.61. errorHtml

POST /error

2.61.1. Responses

HTTP Code	Description	Schema
200	OK	ModelAndView

2.61.2. Consumes

- *application/json*

2.61.3. Produces

- `text/html`

2.61.4. Tags

- basic-error-controller

2.61.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.62. errorHtml

GET /error

2.62.1. Responses

HTTP Code	Description	Schema
200	OK	ModelAndView

2.62.2. Produces

- `text/html`

2.62.3. Tags

- basic-error-controller

2.62.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.63. errorHtml

PUT /error

2.63.1. Responses

HTTP Code	Description	Schema
200	OK	ModelAndView

2.63.2. Consumes

- `application/json`

2.63.3. Produces

- `text/html`

2.63.4. Tags

- basic-error-controller

2.63.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.64. errorHtml

DELETE /error

2.64.1. Responses

HTTP Code	Description	Schema
200	OK	ModelAndView

2.64.2. Produces

- `text/html`

2.64.3. Tags

- basic-error-controller

2.64.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.65. errorHtml

PATCH /error

2.65.1. Responses

HTTP Code	Description	Schema
200	OK	ModelAndView

2.65.2. Consumes

- [application/json](#)

2.65.3. Produces

- [text/html](#)

2.65.4. Tags

- basic-error-controller

2.65.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.66. errorHtml

HEAD /error

2.66.1. Responses

HTTP Code	Description	Schema
200	OK	ModelAndView

2.66.2. Consumes

- `application/json`

2.66.3. Produces

- `text/html`

2.66.4. Tags

- basic-error-controller

2.66.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.67. errorHtml

OPTIONS /error

2.67.1. Responses

HTTP Code	Description	Schema
200	OK	ModelAndView

2.67.2. Consumes

- `application/json`

2.67.3. Produces

- `text/html`

2.67.4. Tags

- basic-error-controller

2.67.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.68. getLanguages

GET /i18n/support-languages

2.68.1. Responses

HTTP Code	Description	Schema
200	OK	< string > array

2.68.2. Produces

- */*

2.68.3. Tags

- I18n

2.68.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.69. getAllInternmentPatient

GET /institutions/{institutionId}/internments/patients

2.69.1. Parameters

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

2.69.2. Responses

HTTP Code	Description	Schema
200	OK	< InternmentEpisodeDto > array

2.69.3. Produces

- */*

2.69.4. Tags

- Internment Patient

2.69.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.70. getIdentityVerificationStatus

POST /masterdata/patient/withoutIdentityReasons

2.70.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.70.2. Consumes

- application/json

2.70.3. Produces

- */*

2.70.4. Tags

- PatientMasterData

2.70.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.71. getIdentityVerificationStatus

GET /masterdata/patient/withoutIdentityReasons

2.71.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.71.2. Produces

- [*/*](#)

2.71.3. Tags

- PatientMasterData

2.71.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.72. getIdentityVerificationStatus

PUT /masterdata/patient/withoutIdentityReasons

2.72.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.72.2. Consumes

- [application/json](#)

2.72.3. Produces

- [*/*](#)

2.72.4. Tags

- PatientMasterData

2.72.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.73. getIdentityVerificationStatus

DELETE /masterdata/patient/withoutIdentityReasons

2.73.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.73.2. Produces

- [*/*](#)

2.73.3. Tags

- PatientMasterData

2.73.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.74. getIdentityVerificationStatus

PATCH /masterdata/patient/withoutIdentityReasons

2.74.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.74.2. Consumes

- application/json

2.74.3. Produces

- */*

2.74.4. Tags

- PatientMasterData

2.74.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.75. getIdentityVerificationStatus

HEAD /masterdata/patient/withoutIdentityReasons

2.75.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.75.2. Consumes

- [application/json](#)

2.75.3. Produces

- [*/*](#)

2.75.4. Tags

- PatientMasterData

2.75.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.76. getIdentityVerificationStatus

OPTIONS /masterdata/patient/withoutIdentityReasons

2.76.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.76.2. Consumes

- [application/json](#)

2.76.3. Produces

- */*

2.76.4. Tags

- PatientMasterData

2.76.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.77. updatePassword

PATCH /passwords/{userId}

2.77.1. Parameters

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

2.77.2. Responses

HTTP Code	Description	Schema
200	OK	boolean

2.77.3. Consumes

- application/json

2.77.4. Produces

- */*

2.77.5. Tags

- Password

2.77.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.78. addPatient

POST /patient

2.78.1. Parameters

Type	Name	Description	Schema
Body	patientDto <i>required</i>	patientDto	APatientDto

2.78.2. Responses

HTTP Code	Description	Schema
200	OK	BMPatientDto

2.78.3. Consumes

- [application/json](#)

2.78.4. Produces

- [*/*](#)

2.78.5. Tags

- Patient

2.78.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.79. getPatientMinimal

GET /patient/minimalsearch

2.79.1. Parameters

Type	Name	Description	Schema
Query	genderId <i>required</i>	genderId	integer (int32)
Query	identificationNumber <i>required</i>	identificationNumber	string
Query	identificationTypeId <i>required</i>	identificationTypeId	integer (int32)

2.79.2. Responses

HTTP Code	Description	Schema
200	OK	< integer (int32) > array

2.79.3. Produces

- **/**

2.79.4. Tags

- Patient

2.79.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.80. searchPatient

GET /patient/search

2.80.1. Parameters

Type	Name	Description	Schema
Query	searchFilterString <i>required</i>	searchFilterStr	string

2.80.2. Responses

HTTP Code	Description	Schema
200	OK	< PatientSearchDto > array

2.80.3. Produces

- */*

2.80.4. Tags

- Patient

2.80.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.81. getBasicDataPatient

GET /patient/{patientId}/basicdata

2.81.1. Parameters

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

2.81.2. Responses

HTTP Code	Description	Schema
200	OK	BasicPatientDto

2.81.3. Produces

- `*/*`

2.81.4. Tags

- Patient

2.81.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.82. addPerson

POST /person

2.82.1. Parameters

Type	Name	Description	Schema
Body	personDto <i>required</i>	personDto	APersonDto

2.82.2. Responses

HTTP Code	Description	Schema
200	OK	BMPersonDto

2.82.3. Consumes

- `application/json`

2.82.4. Produces

- **/**

2.82.5. Tags

- Person

2.82.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.83. getPersonMinimal

GET /person/minimalsearch

2.83.1. Parameters

Type	Name	Description	Schema
Query	genderId <i>required</i>	genderId	integer (int32)
Query	identificationNumber <i>required</i>	identificationNumber	string
Query	identificationTypeId <i>required</i>	identificationTypeId	integer (int32)

2.83.2. Responses

HTTP Code	Description	Schema
200	OK	< integer (int32) > array

2.83.3. Produces

- **/**

2.83.4. Tags

- Person

2.83.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.84. getPersonalInformation

GET /person/{personId}/personalInformation

2.84.1. Parameters

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

2.84.2. Responses

HTTP Code	Description	Schema
200	OK	PersonalInformationDto

2.84.3. Produces

- **/**

2.84.4. Tags

- Person

2.84.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.85. setRoleUser

POST /roles

2.85.1. Parameters

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

2.85.2. Responses

HTTP Code	Description	Schema
200	OK	boolean

2.85.3. Consumes

- [application/json](#)

2.85.4. Produces

- [*/*](#)

2.85.5. Tags

- Role

2.85.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.86. addUser

POST /users

2.86.1. Parameters

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

2.86.2. Responses

HTTP Code	Description	Schema
200	OK	UserDto

2.86.3. Consumes

- `application/json`

2.86.4. Produces

- `*/*`

2.86.5. Tags

- User

2.86.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.87. getUsers

GET /users/page

2.87.1. Parameters

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

Type	Name	Schema
Query	sort <i>optional</i>	string

2.87.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfPageableUsers

2.87.3. Produces

- **/**

2.87.4. Tags

- User

2.87.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.88. changeStatusAccount

PATCH /users/{id}

2.88.1. Parameters

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

2.88.2. Responses

HTTP Code	Description	Schema
200	OK	boolean

2.88.3. Consumes

- `application/json`

2.88.4. Produces

- `*/*`

2.88.5. Tags

- User

2.88.6. 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. APatientDto

Name	Description	Schema
apartment <i>optional</i>		string
birthDate <i>required</i>	Example : "yyyy-MM-dd"	string
cityId <i>optional</i>		integer (int32)
cuil <i>optional</i>		string
email <i>optional</i>		string
ethnic <i>optional</i>		string
firstName <i>required</i>		string
floor <i>optional</i>		string
genderId <i>required</i>		integer (int32)
genderSelfDet erminationId <i>optional</i>		integer (int32)
identification Number <i>optional</i>		string
identification TypeId <i>required</i>		integer (int32)

Name	Description	Schema
lastName <i>required</i>		string
middleNames <i>optional</i>		string
mothersLastName <i>optional</i>		string
nameSelfDetermination <i>optional</i>		string
number <i>optional</i>		string
otherLastNames <i>optional</i>		string
phoneNumber <i>optional</i>		string
postcode <i>optional</i>		string
quarter <i>optional</i>		string
religion <i>optional</i>		string
street <i>optional</i>		string

4.2. APersonDto

Name	Description	Schema
apartment <i>optional</i>		string

Name	Description	Schema
birthDate <i>required</i>	Example : "yyyy-MM-dd"	string
cityId <i>optional</i>		integer (int32)
cuil <i>optional</i>		string
email <i>optional</i>		string
ethnic <i>optional</i>		string
firstName <i>required</i>		string
floor <i>optional</i>		string
genderId <i>required</i>		integer (int32)
genderSelfDet erminationId <i>optional</i>		integer (int32)
identification Number <i>optional</i>		string
identification TypeId <i>required</i>		integer (int32)
lastName <i>required</i>		string
middleNames <i>optional</i>		string

Name	Description	Schema
mothersLastName <i>optional</i>		string
nameSelfDetermination <i>optional</i>		string
number <i>optional</i>		string
otherLastNames <i>optional</i>		string
phoneNumber <i>optional</i>		string
postcode <i>optional</i>		string
quarter <i>optional</i>		string
religion <i>optional</i>		string
street <i>optional</i>		string

4.3. AddUserDto

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

4.4. Address

Name	Schema
apartment <i>optional</i>	string
cityId <i>optional</i>	integer (int32)
floor <i>optional</i>	string
id <i>optional</i>	integer (int32)
number <i>optional</i>	string
postcode <i>optional</i>	string
quarter <i>optional</i>	string
street <i>optional</i>	string

4.5. AddressDto

Name	Schema
apartment <i>optional</i>	string
city <i>optional</i>	CityDto
floor <i>optional</i>	string
id <i>optional</i>	integer (int32)

Name	Schema
number <i>optional</i>	string
postcode <i>optional</i>	string
province <i>optional</i>	ProvinceDto
quarter <i>optional</i>	string
street <i>optional</i>	string

4.6. AddressProjection

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

4.7. BMPatientDto

Name	Description	Schema
apartment <i>optional</i>		string
birthDate <i>required</i>	Example : "yyyy-MM-dd"	string
cityId <i>optional</i>		integer (int32)
cuil <i>optional</i>		string

Name	Description	Schema
email <i>optional</i>		string
ethnic <i>optional</i>		string
firstName <i>required</i>		string
floor <i>optional</i>		string
genderId <i>required</i>		integer (int32)
genderSelfDet erminationId <i>optional</i>		integer (int32)
id <i>optional</i>		integer (int32)
identification Number <i>optional</i>		string
identification TypeId <i>required</i>		integer (int32)
lastName <i>required</i>		string
middleNames <i>optional</i>		string
mothersLastN ame <i>optional</i>		string
nameSelfDete rmination <i>optional</i>		string

Name	Description	Schema
number <i>optional</i>		string
otherLastNames <i>optional</i>		string
phoneNumber <i>optional</i>		string
postcode <i>optional</i>		string
quarter <i>optional</i>		string
religion <i>optional</i>		string
street <i>optional</i>		string

4.8. BMPersonDto

Name	Description	Schema
apartment <i>optional</i>		string
birthDate <i>required</i>	Example : "yyyy-MM-dd"	string
cityId <i>optional</i>		integer (int32)
cuil <i>optional</i>		string
email <i>optional</i>		string

Name	Description	Schema
ethnic <i>optional</i>		string
firstName <i>required</i>		string
floor <i>optional</i>		string
genderId <i>required</i>		integer (int32)
genderSelfDet erminationId <i>optional</i>		integer (int32)
id <i>optional</i>		integer (int32)
identification Number <i>optional</i>		string
identification TypeId <i>required</i>		integer (int32)
lastName <i>required</i>		string
middleNames <i>optional</i>		string
mothersLastN ame <i>optional</i>		string
nameSelfDete rmination <i>optional</i>		string
number <i>optional</i>		string

Name	Description	Schema
otherLastNames <i>optional</i>		string
phoneNumber <i>optional</i>		string
postcode <i>optional</i>		string
quarter <i>optional</i>		string
religion <i>optional</i>		string
street <i>optional</i>		string

4.9. BackofficeDeleteResponseOfAddress

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

4.10. BackofficeDeleteResponseOfBed

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

4.11. BackofficeDeleteResponseOfCity

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

4.12. BackofficeDeleteResponseOfClinicalSpecialty

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

4.13.

BackofficeDeleteResponseOfClinicalSpecialtySector

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

4.14. BackofficeDeleteResponseOfDepartment

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

4.15. BackofficeDeleteResponseOfInstitution

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

4.16. BackofficeDeleteResponseOfProvince

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

4.17. BackofficeDeleteResponseOfRoom

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

4.18. BackofficeDeleteResponseOfSector

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

4.19. BasicDataPersonDto

Name	Schema
age <i>optional</i>	integer (int32)
firstName <i>optional</i>	string
genderId <i>optional</i>	integer (int32)
id <i>optional</i>	integer (int32)
lastName <i>optional</i>	string

4.20. BasicPatientDto

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

4.21. Bed

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

4.22. BedDto

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

4.23. Character

Type : object

4.24. City

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

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

4.25. CityDto

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

4.26. ClinicalSpecialty

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

4.27. ClinicalSpecialtyDto

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

4.28. ClinicalSpecialtySector

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

4.29. Department

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

4.30. HealthInsuranceDto

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

4.31. IdentificationTypeDto

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

4.32. IdentityVerificationStatus

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

4.33. Institution

Name	Schema
addressId <i>optional</i>	integer (int32)
cuit <i>optional</i>	string
email <i>optional</i>	string
id <i>optional</i>	integer (int32)
name <i>optional</i>	string
phone <i>optional</i>	string
sisCode <i>optional</i>	string

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

4.34. InternmentEpisodeDto

Name	Schema
bed <i>optional</i>	BedDto
doctor <i>optional</i>	ResponsibleDoctorDto
id <i>optional</i>	integer (int32)
patient <i>optional</i>	PatientDto

4.35. IterableOfAddress

Type : object

4.36. IterableOfBed

Type : object

4.37. IterableOfCity

Type : object

4.38. IterableOfClinicalSpecialty

Type : object

4.39. IterableOfClinicalSpecialtySector

Type : object

4.40. IterableOfDepartment

Type : object

4.41. IterableOfInstitution

Type : object

4.42. IterableOfProvince

Type : object

4.43. IterableOfRoom

Type : object

4.44. IterableOfSector

Type : object

4.45. JWTokenDto

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

4.46. Link

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

4.47. LoginDto

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

4.48. MapOfStringAndLink

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

4.49. ModelAndView

Name	Schema
empty <i>optional</i>	boolean
model <i>optional</i>	object
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, 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)
view <i>optional</i>	View

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

4.50. PageOfAddress

Name	Schema
content <i>optional</i>	< Address > 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)

4.51. PageOfBed

Name	Schema
content <i>optional</i>	< Bed > 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)

4.52. PageOfCity

Name	Schema
content <i>optional</i>	< City > 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)

4.53. PageOfClinicalSpecialty

Name	Schema
content <i>optional</i>	< ClinicalSpecialty > array
empty <i>optional</i>	boolean
first <i>optional</i>	boolean

Name	Schema
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)

4.54. PageOfClinicalSpecialtySector

Name	Schema
content <i>optional</i>	< ClinicalSpecialtySector > array
empty <i>optional</i>	boolean
first <i>optional</i>	boolean
last <i>optional</i>	boolean
number <i>optional</i>	integer (int32)

Name	Schema
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)

4.55. 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

Name	Schema
size <i>optional</i>	integer (int32)
sort <i>optional</i>	Sort
totalElements <i>optional</i>	integer (int64)
totalPages <i>optional</i>	integer (int32)

4.56. PageOfInstitution

Name	Schema
content <i>optional</i>	< Institution > 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

Name	Schema
totalElements <i>optional</i>	integer (int64)
totalPages <i>optional</i>	integer (int32)

4.57. PageOfPageableUsers

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

4.58. PageOfProvince

Name	Schema
content <i>optional</i>	< Province > 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)

4.59. PageOfRoom

Name	Schema
content <i>optional</i>	< Room > 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)

4.60. PageOfSector

Name	Schema
content <i>optional</i>	< Sector > array
empty <i>optional</i>	boolean
first <i>optional</i>	boolean

Name	Schema
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)

4.61. Pageable

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

4.62. 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

4.63. PatientDto

Name	Schema
name <i>optional</i>	string
patientId <i>optional</i>	integer (int32)
surname <i>optional</i>	string

4.64. PatientSearchDto

Name	Schema
person <i>optional</i>	BMPersonDto
ranking <i>optional</i>	number (float)

4.65. PersonalInformationDto

Name	Schema
address <i>optional</i>	AddressDto
birthDate <i>optional</i>	string (date)
cuil <i>optional</i>	string
email <i>optional</i>	string
healthInsurance <i>optional</i>	HealthInsuranceDto
id <i>optional</i>	integer (int32)
identificationNumber <i>optional</i>	string
identificationType <i>optional</i>	IdentificationTypeDto
phoneNumber <i>optional</i>	string

4.66. Province

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

4.67. ProvinceDto

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

4.68. RequestUserRoleDto

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

4.69. ResponsibleDoctorDto

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

4.70. Room

Name	Schema
description <i>optional</i>	string
dischargeDate <i>optional</i>	string (date)

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

4.71. RoomDto

Name	Schema
id <i>optional</i>	integer (int32)
roomNumber <i>optional</i>	integer (int32)
sector <i>optional</i>	SectorDto
specialty <i>optional</i>	ClinicalSpecialtyDto

4.72. Sector

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

4.73. SectorDto

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

4.74. Sort

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

4.75. UserDto

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

4.76. View

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