

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	18
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. getMany	19
2.22.1. Parameters	20
2.22.2. Responses	20
2.22.3. Produces	20
2.22.4. Tags	20
2.22.5. Security	20
2.23. getOne	20
2.23.1. Parameters	20
2.23.2. Responses	20
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	21
2.24.3. Consumes	21
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	22
2.25.3. Produces	22
2.25.4. Tags	22
2.25.5. Security	22

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

2.32.2. Responses	28
2.32.3. Produces	28
2.32.4. Tags	28
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	29
2.33.5. Security	29
2.34. update	29
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	30
2.35. delete	30
2.35.1. Parameters	30
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	31
2.36.2. Responses	31
2.36.3. Consumes	32
2.36.4. Produces	32
2.36.5. Tags	32
2.36.6. Security	32
2.37. getList	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	34
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	35
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	36
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. getList	36
2.42.1. Parameters	36
2.42.2. Responses	37
2.42.3. Produces	37
2.42.4. Tags	38
2.42.5. Security	38
2.43. getOne	38
2.43.1. Parameters	38
2.43.2. Responses	38
2.43.3. Produces	38
2.43.4. Tags	38
2.43.5. Security	38
2.44. update	39
2.44.1. Parameters	39
2.44.2. Responses	39
2.44.3. Consumes	39
2.44.4. Produces	39
2.44.5. Tags	39
2.44.6. Security	39
2.45. delete	39

2.45.1. Parameters	40
2.45.2. Responses	40
2.45.3. Produces	40
2.45.4. Tags	40
2.45.5. Security	40
2.46. create	40
2.46.1. Parameters	40
2.46.2. Responses	40
2.46.3. Consumes	41
2.46.4. Produces	41
2.46.5. Tags	41
2.46.6. Security	41
2.47. getMany	41
2.47.1. Parameters	41
2.47.2. Responses	41
2.47.3. Produces	41
2.47.4. Tags	42
2.47.5. Security	42
2.48. getOne	42
2.48.1. Parameters	42
2.48.2. Responses	42
2.48.3. Produces	42
2.48.4. Tags	42
2.48.5. Security	42
2.49. update	43
2.49.1. Parameters	43
2.49.2. Responses	43
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	44
2.50.2. Responses	44
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	45

2.51.4. Produces	45
2.51.5. Tags	45
2.51.6. Security	45
2.52. getList	45
2.52.1. Parameters	45
2.52.2. Responses	46
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	47
2.53.3. Produces	47
2.53.4. Tags	47
2.53.5. Security	47
2.54. update	47
2.54.1. Parameters	47
2.54.2. Responses	47
2.54.3. Consumes	48
2.54.4. Produces	48
2.54.5. Tags	48
2.54.6. Security	48
2.55. delete	48
2.55.1. Parameters	48
2.55.2. Responses	48
2.55.3. Produces	48
2.55.4. Tags	49
2.55.5. Security	49
2.56. create	49
2.56.1. Parameters	49
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	50
2.57.1. Parameters	50
2.57.2. Responses	50
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	51
2.58.3. Produces	51
2.58.4. Tags	51
2.58.5. Security	51
2.59. update	51
2.59.1. Parameters	51
2.59.2. Responses	51
2.59.3. Consumes	52
2.59.4. Produces	52
2.59.5. Tags	52
2.59.6. Security	52
2.60. delete	52
2.60.1. Parameters	52
2.60.2. Responses	52
2.60.3. Produces	52
2.60.4. Tags	52
2.60.5. Security	53
2.61. error	53
2.61.1. Responses	53
2.61.2. Consumes	53
2.61.3. Produces	53
2.61.4. Tags	53
2.61.5. Security	53
2.62. error	53
2.62.1. Responses	53
2.62.2. Produces	54
2.62.3. Tags	54
2.62.4. Security	54
2.63. error	54
2.63.1. Responses	54
2.63.2. Consumes	54
2.63.3. Produces	54
2.63.4. Tags	54
2.63.5. Security	55
2.64. error	55
2.64.1. Responses	55
2.64.2. Produces	55
2.64.3. Tags	55
2.64.4. Security	55

2.65. error	55
2.65.1. Responses	55
2.65.2. Consumes	56
2.65.3. Produces	56
2.65.4. Tags	56
2.65.5. Security	56
2.66. error	56
2.66.1. Responses	56
2.66.2. Consumes	56
2.66.3. Produces	56
2.66.4. Tags	56
2.66.5. Security	56
2.67. error	57
2.67.1. Responses	57
2.67.2. Consumes	57
2.67.3. Produces	57
2.67.4. Tags	57
2.67.5. Security	57
2.68. getLanguages	57
2.68.1. Responses	57
2.68.2. Produces	58
2.68.3. Tags	58
2.68.4. Security	58
2.69. getAllInternmentPatient	58
2.69.1. Parameters	58
2.69.2. Responses	58
2.69.3. Produces	58
2.69.4. Tags	59
2.69.5. Security	59
2.70. createAnamnesis	59
2.70.1. Parameters	59
2.70.2. Responses	59
2.70.3. Consumes	59
2.70.4. Produces	59
2.70.5. Tags	60
2.70.6. Security	60
2.71. getAnamnesis	60
2.71.1. Parameters	60
2.71.2. Responses	60
2.71.3. Produces	60
2.71.4. Tags	60

2.71.5. Security	61
2.72. updateAnamnesis	61
2.72.1. Parameters	61
2.72.2. Responses	61
2.72.3. Consumes	61
2.72.4. Produces	61
2.72.5. Tags	62
2.72.6. Security	62
2.73. internmentEpisodeSummary	62
2.73.1. Parameters	62
2.73.2. Responses	62
2.73.3. Produces	62
2.73.4. Tags	62
2.73.5. Security	62
2.74. getAllergyCategory	63
2.74.1. Responses	63
2.74.2. Produces	63
2.74.3. Tags	63
2.74.4. Security	63
2.75. getAllergyClinical	63
2.75.1. Responses	63
2.75.2. Produces	64
2.75.3. Tags	64
2.75.4. Security	64
2.76. getAllergyVerification	64
2.76.1. Responses	64
2.76.2. Produces	64
2.76.3. Tags	64
2.76.4. Security	65
2.77. getDocumentStatus	65
2.77.1. Responses	65
2.77.2. Produces	65
2.77.3. Tags	65
2.77.4. Security	65
2.78. getDocumentType	65
2.78.1. Responses	65
2.78.2. Produces	66
2.78.3. Tags	66
2.78.4. Security	66
2.79. getInternmentEpisode	66
2.79.1. Responses	66

2.79.2. Produces	66
2.79.3. Tags	66
2.79.4. Security	67
2.80. getHealthClinical	67
2.80.1. Responses	67
2.80.2. Produces	67
2.80.3. Tags	67
2.80.4. Security	67
2.81. getHealthProblem	67
2.81.1. Responses	67
2.81.2. Produces	68
2.81.3. Tags	68
2.81.4. Security	68
2.82. getHealthVerification	68
2.82.1. Responses	68
2.82.2. Produces	68
2.82.3. Tags	68
2.82.4. Security	69
2.83. getInmunization	69
2.83.1. Responses	69
2.83.2. Produces	69
2.83.3. Tags	69
2.83.4. Security	69
2.84. getHealthMedication	69
2.84.1. Responses	69
2.84.2. Produces	70
2.84.3. Tags	70
2.84.4. Security	70
2.85. getObservation	70
2.85.1. Responses	70
2.85.2. Produces	70
2.85.3. Tags	70
2.85.4. Security	71
2.86. getIdentityVerificationStatus	71
2.86.1. Responses	71
2.86.2. Consumes	71
2.86.3. Produces	71
2.86.4. Tags	71
2.86.5. Security	71
2.87. getIdentityVerificationStatus	71
2.87.1. Responses	71

2.87.2. Produces	72
2.87.3. Tags	72
2.87.4. Security	72
2.88. getIdentityVerificationStatus	72
2.88.1. Responses	72
2.88.2. Consumes	72
2.88.3. Produces	72
2.88.4. Tags	72
2.88.5. Security	73
2.89. getIdentityVerificationStatus	73
2.89.1. Responses	73
2.89.2. Produces	73
2.89.3. Tags	73
2.89.4. Security	73
2.90. getIdentityVerificationStatus	73
2.90.1. Responses	73
2.90.2. Consumes	74
2.90.3. Produces	74
2.90.4. Tags	74
2.90.5. Security	74
2.91. getIdentityVerificationStatus	74
2.91.1. Responses	74
2.91.2. Consumes	74
2.91.3. Produces	74
2.91.4. Tags	75
2.91.5. Security	75
2.92. getIdentityVerificationStatus	75
2.92.1. Responses	75
2.92.2. Consumes	75
2.92.3. Produces	75
2.92.4. Tags	75
2.92.5. Security	75
2.93. updatePassword	75
2.93.1. Parameters	76
2.93.2. Responses	76
2.93.3. Consumes	76
2.93.4. Produces	76
2.93.5. Tags	76
2.93.6. Security	76
2.94. addPatient	76
2.94.1. Parameters	76

2.94.2. Responses	77
2.94.3. Consumes	77
2.94.4. Produces	77
2.94.5. Tags	77
2.94.6. Security	77
2.95. getAllPatientsData	77
2.95.1. Responses	77
2.95.2. Produces	77
2.95.3. Tags	77
2.95.4. Security	78
2.96. getPatientMinimal	78
2.96.1. Parameters	78
2.96.2. Responses	78
2.96.3. Produces	78
2.96.4. Tags	78
2.96.5. Security	78
2.97. searchPatient	79
2.97.1. Parameters	79
2.97.2. Responses	79
2.97.3. Produces	79
2.97.4. Tags	79
2.97.5. Security	79
2.98. getBasicDataPatient	79
2.98.1. Parameters	80
2.98.2. Responses	80
2.98.3. Produces	80
2.98.4. Tags	80
2.98.5. Security	80
2.99. addPerson	80
2.99.1. Parameters	80
2.99.2. Responses	80
2.99.3. Consumes	81
2.99.4. Produces	81
2.99.5. Tags	81
2.99.6. Security	81
2.100. getGenders	81
2.100.1. Responses	81
2.100.2. Produces	81
2.100.3. Tags	81
2.100.4. Security	81
2.101. getIdentificationTypes	82

2.101.1. Responses	82
2.101.2. Produces	82
2.101.3. Tags	82
2.101.4. Security	82
2.102. getPersonMinimal	82
2.102.1. Parameters	82
2.102.2. Responses	83
2.102.3. Produces	83
2.102.4. Tags	83
2.102.5. Security	83
2.103. getPersonalInformation	83
2.103.1. Parameters	83
2.103.2. Responses	84
2.103.3. Produces	84
2.103.4. Tags	84
2.103.5. Security	84
2.104. setRoleUser	84
2.104.1. Parameters	84
2.104.2. Responses	84
2.104.3. Consumes	84
2.104.4. Produces	85
2.104.5. Tags	85
2.104.6. Security	85
2.105. addUser	85
2.105.1. Parameters	85
2.105.2. Responses	85
2.105.3. Consumes	85
2.105.4. Produces	85
2.105.5. Tags	85
2.105.6. Security	86
2.106. getUsers	86
2.106.1. Parameters	86
2.106.2. Responses	86
2.106.3. Produces	86
2.106.4. Tags	86
2.106.5. Security	86
2.107. changeStatusAccount	87
2.107.1. Parameters	87
2.107.2. Responses	87
2.107.3. Consumes	87
2.107.4. Produces	87

2.107.5. Tags	87
2.107.6. Security	87
3. Security	88
3.1. JWT	88
4. Definitions	89
4.1. APatientDto	89
4.2. APersonDto	91
4.3. AddUserDto	93
4.4. Address	93
4.5. AddressDto	93
4.6. AddressProjection	94
4.7. AllergyConditionDto	94
4.8. AnamnesisDto	95
4.9. AnthropometricDataDto	96
4.10. BMPatientDto	96
4.11. BMPersonDto	98
4.12. BackofficeDeleteResponseOfAddress	100
4.13. BackofficeDeleteResponseOfBed	100
4.14. BackofficeDeleteResponseOfCity	101
4.15. BackofficeDeleteResponseOfClinicalSpecialty	101
4.16. BackofficeDeleteResponseOfClinicalSpecialtySector	101
4.17. BackofficeDeleteResponseOfDepartment	101
4.18. BackofficeDeleteResponseOfInstitution	101
4.19. BackofficeDeleteResponseOfProvince	102
4.20. BackofficeDeleteResponseOfRoom	102
4.21. BackofficeDeleteResponseOfSector	102
4.22. BasicDataPersonDto	102
4.23. BasicPatientDto	102
4.24. Bed	103
4.25. BedDto	103
4.26. Character	103
4.27. City	104
4.28. CityDto	104
4.29. ClinicalSpecialty	104
4.30. ClinicalSpecialtyDto	105
4.31. ClinicalSpecialtySector	105
4.32. Department	105
4.33. GenderDto	105
4.34. HealthConditionDto	106
4.35. HealthHistoryConditionDto	106
4.36. HealthInsuranceDto	106

4.37. IdentificationTypeDto	107
4.38. IdentityVerificationStatus	107
4.39. ImmunizationDto	107
4.40. Institution	108
4.41. InternmentEpisodeDto	108
4.42. InternmentMasterDataProjection	109
4.43. InternmentSummaryDto	109
4.44. IterableOfAddress	110
4.45. IterableOfBed	110
4.46. IterableOfCity	110
4.47. IterableOfClinicalSpecialty	110
4.48. IterableOfClinicalSpecialtySector	110
4.49. IterableOfDepartment	110
4.50. IterableOfInstitution	110
4.51. IterableOfProvince	110
4.52. IterableOfRoom	110
4.53. IterableOfSector	110
4.54. JWTTokenDto	110
4.55. Link	111
4.56. LoginDto	111
4.57. MapOfstringAndLink	111
4.58. MedicationDto	111
4.59. ModelAndView	112
4.60. ObservationsDto	114
4.61. PageOfAddress	114
4.62. PageOfBed	115
4.63. PageOfCity	116
4.64. PageOfClinicalSpecialty	117
4.65. PageOfClinicalSpecialtySector	118
4.66. PageOfDepartment	119
4.67. PageOfInstitution	119
4.68. PageOfPageableUsers	120
4.69. PageOfProvince	121
4.70. PageOfRoom	122
4.71. PageOfSector	123
4.72. Pageable	124
4.73. PageableUsers	124
4.74. PatientDto	124
4.75. PatientSearchDto	125
4.76. PersonalInformationDto	125
4.77. Province	126

4.78. ProvinceDto	126
4.79. RequestUserRoleDto	126
4.80. ResponseAnamnesisDto	127
4.81. ResponsibleDoctorDto	128
4.82. Room	128
4.83. RoomDto	128
4.84. Sector	129
4.85. SectorDto	129
4.86. SnomedDto	129
4.87. Sort	130
4.88. UserDto	130
4.89. View	130
4.90. VitalSignDto	130

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
- Anamnesis : Anamnesis Controller
- Authorization : Authentication Controller
- I18n : I 18 N Controller
- Internment Episode : Internment Episode Controller
- Internment Patient : Internment Patient Controller
- Internments Master Data : Internment Masterdata Controller
- Password : User Password Controller
- Patient : Patient Controller
- PatientMasterData : Patient Master Data Controller
- Person : Person Controller
- Person Master Data : person Master Data 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	bedNumber <i>optional</i>	string
Query	id <i>optional</i>	integer (int32)
Query	page <i>optional</i>	integer (int32)
Query	roomId <i>optional</i>	integer (int32)

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

GET /backoffice/cities

2.22.1. Parameters

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

2.22.2. Responses

HTTP Code	Description	Schema
200	OK	IterableOfCity

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. getMany

GET /backoffice/clinicalspecialties

2.27.1. Parameters

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

2.27.2. Responses

HTTP Code	Description	Schema
200	OK	IterableOfClinicalSpecialty

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. getList

GET /backoffice/clinicalspecialtysectors

2.32.1. Parameters

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

2.32.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfClinicalSpecialtySector

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. getList

GET /backoffice/departments

2.37.1. 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)

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

2.37.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfDepartment

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. getList

GET /backoffice/institutions

2.42.1. Parameters

Type	Name	Schema
Query	addressId <i>optional</i>	integer (int32)
Query	cuit <i>optional</i>	string
Query	email <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	phone <i>optional</i>	string
Query	sisCode <i>optional</i>	string
Query	size <i>optional</i>	integer (int32)
Query	sort <i>optional</i>	string
Query	website <i>optional</i>	string

2.42.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfInstitution

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. getMany

GET /backoffice/provinces

2.47.1. Parameters

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

2.47.2. Responses

HTTP Code	Description	Schema
200	OK	IterableOfProvince

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	clinicalSpecialtySectorId <i>optional</i>	integer (int32)
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)

Type	Name	Schema
Query	roomNumber <i>optional</i>	string
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. error

POST /error

2.61.1. Responses

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

2.61.2. Consumes

- application/json

2.61.3. Produces

- */*

2.61.4. Tags

- basic-error-controller

2.61.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.62. error

GET /error

2.62.1. Responses

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

2.62.2. Produces

- */*

2.62.3. Tags

- basic-error-controller

2.62.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.63. error

PUT /error

2.63.1. Responses

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

2.63.2. Consumes

- application/json

2.63.3. Produces

- */*

2.63.4. Tags

- basic-error-controller

2.63.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.64. error

DELETE /error

2.64.1. Responses

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

2.64.2. Produces

- */*

2.64.3. Tags

- basic-error-controller

2.64.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.65. error

PATCH /error

2.65.1. Responses

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

2.65.2. Consumes

- `application/json`

2.65.3. Produces

- `*/*`

2.65.4. Tags

- `basic-error-controller`

2.65.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.66. error

HEAD /error

2.66.1. Responses

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

2.66.2. Consumes

- `application/json`

2.66.3. Produces

- `*/*`

2.66.4. Tags

- `basic-error-controller`

2.66.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.67. error

OPTIONS /error

2.67.1. Responses

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

2.67.2. Consumes

- application/json

2.67.3. Produces

- */*

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. createAnamnesis

POST /institutions/{institutionId}/internments/{internmentEpisodeId}/anamnesis

2.70.1. Parameters

Type	Name	Description	Schema
Path	institutionId <i>required</i>	institutionId	integer (int32)
Path	internmentEpisodeId <i>required</i>	internmentEpisodeId	integer (int32)
Body	anamnesisDto <i>required</i>	anamnesisDto	AnamnesisDto

2.70.2. Responses

HTTP Code	Description	Schema
200	OK	ResponseAnamnesisDto

2.70.3. Consumes

- [application/json](#)

2.70.4. Produces

- [*/*](#)

2.70.5. Tags

- Anamnesis

2.70.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.71. getAnamnesis

```
GET
/institutions/{institutionId}/internments/{internmentEpisodeId}/anamnesis/{anamnesisId}
}
```

2.71.1. Parameters

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

2.71.2. Responses

HTTP Code	Description	Schema
200	OK	ResponseAnamnesisDto

2.71.3. Produces

- **/**

2.71.4. Tags

- Anamnesis

2.71.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.72. updateAnamnesis

```
PUT
/institutions/{institutionId}/internments/{internmentEpisodeId}/anamnesis/{anamnesisId}
}
```

2.72.1. Parameters

Type	Name	Description	Schema
Path	anamnesisId <i>required</i>	anamnesisId	integer (int32)
Path	institutionId <i>required</i>	institutionId	integer (int32)
Path	internmentEpisodeId <i>required</i>	internmentEpisodeId	integer (int32)
Body	anamnesisDto <i>required</i>	anamnesisDto	AnamnesisDto

2.72.2. Responses

HTTP Code	Description	Schema
200	OK	ResponseAnamnesisDto

2.72.3. Consumes

- [application/json](#)

2.72.4. Produces

- [*/*](#)

2.72.5. Tags

- Anamnesis

2.72.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.73. internmentEpisodeSummary

GET /institutions/{institutionId}/internments/{internmentEpisodeId}/summary

2.73.1. Parameters

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

2.73.2. Responses

HTTP Code	Description	Schema
200	OK	InternmentSummaryDto

2.73.3. Produces

- **/**

2.73.4. Tags

- Internment Episode

2.73.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.74. getAllergyCategory

GET /internments/masterdata/allergy/category

2.74.1. Responses

HTTP Code	Description	Schema
200	OK	<pre>< InternmentMaster DataProjection > array</pre>

2.74.2. Produces

- */*

2.74.3. Tags

- Internments Master Data

2.74.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.75. getAllergyClinical

GET /internments/masterdata/allergy/clinical

2.75.1. Responses

HTTP Code	Description	Schema
200	OK	< InternmentMaster DataProjection > array

2.75.2. Produces

- */*

2.75.3. Tags

- Internments Master Data

2.75.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.76. getAllergyVerification

GET /internments/masterdata/allergy/verification

2.76.1. Responses

HTTP Code	Description	Schema
200	OK	< InternmentMaster DataProjection > array

2.76.2. Produces

- */*

2.76.3. Tags

- Internments Master Data

2.76.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.77. getDocumentStatus

GET /internments/masterdata/document/status

2.77.1. Responses

HTTP Code	Description	Schema
200	OK	< InternmentMasterDataProjection > array

2.77.2. Produces

- [*/*](#)

2.77.3. Tags

- Internments Master Data

2.77.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.78. getDocumentType

GET /internments/masterdata/document/type

2.78.1. Responses

HTTP Code	Description	Schema
200	OK	< InternmentMaster DataProjection > array

2.78.2. Produces

- */*

2.78.3. Tags

- Internments Master Data

2.78.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.79. getInternmentEpisode

GET /internments/masterdata/episode

2.79.1. Responses

HTTP Code	Description	Schema
200	OK	< InternmentMaster DataProjection > array

2.79.2. Produces

- */*

2.79.3. Tags

- Internments Master Data

2.79.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.80. getHealthClinical

GET /internments/masterdata/health/clinical

2.80.1. Responses

HTTP Code	Description	Schema
200	OK	< InternmentMasterDataProjection > array

2.80.2. Produces

- [*/*](#)

2.80.3. Tags

- Internments Master Data

2.80.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.81. getHealthProblem

GET /internments/masterdata/health/problem

2.81.1. Responses

HTTP Code	Description	Schema
200	OK	< InternmentMaster DataProjection > array

2.81.2. Produces

- */*

2.81.3. Tags

- Internments Master Data

2.81.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.82. getHealthVerification

GET /internments/masterdata/health/verification

2.82.1. Responses

HTTP Code	Description	Schema
200	OK	< InternmentMaster DataProjection > array

2.82.2. Produces

- */*

2.82.3. Tags

- Internments Master Data

2.82.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.83. getInmunization

GET /internments/masterdata/inmunization

2.83.1. Responses

HTTP Code	Description	Schema
200	OK	< InternmentMasterDataProjection > array

2.83.2. Produces

- [*/*](#)

2.83.3. Tags

- Internments Master Data

2.83.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.84. getHealthMedication

GET /internments/masterdata/medication

2.84.1. Responses

HTTP Code	Description	Schema
200	OK	<pre>< InternmentMaster DataProjection > array</pre>

2.84.2. Produces

- */*

2.84.3. Tags

- Internments Master Data

2.84.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.85. getObservation

```
GET /internments/masterdata/observation
```

2.85.1. Responses

HTTP Code	Description	Schema
200	OK	<pre>< InternmentMaster DataProjection > array</pre>

2.85.2. Produces

- */*

2.85.3. Tags

- Internments Master Data

2.85.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.86. getIdentityVerificationStatus

POST /masterdata/patient/withoutIdentityReasons

2.86.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.86.2. Consumes

- application/json

2.86.3. Produces

- */*

2.86.4. Tags

- PatientMasterData

2.86.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.87. getIdentityVerificationStatus

GET /masterdata/patient/withoutIdentityReasons

2.87.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.87.2. Produces

- */*

2.87.3. Tags

- PatientMasterData

2.87.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.88. getIdentityVerificationStatus

PUT /masterdata/patient/withoutIdentityReasons

2.88.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.88.2. Consumes

- application/json

2.88.3. Produces

- */*

2.88.4. Tags

- PatientMasterData

2.88.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.89. getIdentityVerificationStatus

DELETE /masterdata/patient/withoutIdentityReasons

2.89.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.89.2. Produces

- [*/*](#)

2.89.3. Tags

- PatientMasterData

2.89.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.90. getIdentityVerificationStatus

PATCH /masterdata/patient/withoutIdentityReasons

2.90.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.90.2. Consumes

- `application/json`

2.90.3. Produces

- `*/*`

2.90.4. Tags

- PatientMasterData

2.90.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.91. getIdentityVerificationStatus

HEAD /masterdata/patient/withoutIdentityReasons

2.91.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.91.2. Consumes

- `application/json`

2.91.3. Produces

- `*/*`

2.91.4. Tags

- PatientMasterData

2.91.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.92. getIdentityVerificationStatus

OPTIONS /masterdata/patient/withoutIdentityReasons

2.92.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentityVerificationStatus > array

2.92.2. Consumes

- application/json

2.92.3. Produces

- */*

2.92.4. Tags

- PatientMasterData

2.92.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.93. updatePassword

PATCH /passwords/{userId}

2.93.1. Parameters

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

2.93.2. Responses

HTTP Code	Description	Schema
200	OK	boolean

2.93.3. Consumes

- `application/json`

2.93.4. Produces

- `*/*`

2.93.5. Tags

- Password

2.93.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.94. addPatient

POST /patient

2.94.1. Parameters

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

2.94.2. Responses

HTTP Code	Description	Schema
200	OK	BMPatientDto

2.94.3. Consumes

- [application/json](#)

2.94.4. Produces

- [*/*](#)

2.94.5. Tags

- Patient

2.94.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.95. getAllPatientsData

GET /patient/basicdata

2.95.1. Responses

HTTP Code	Description	Schema
200	OK	< BMPatientDto > array

2.95.2. Produces

- [*/*](#)

2.95.3. Tags

- Patient

2.95.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.96. getPatientMinimal

GET /patient/minimalsearch

2.96.1. 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)

2.96.2. Responses

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

2.96.3. Produces

- **/**

2.96.4. Tags

- Patient

2.96.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.97. searchPatient

GET /patient/search

2.97.1. Parameters

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

2.97.2. Responses

HTTP Code	Description	Schema
200	OK	< PatientSearchDto > array

2.97.3. Produces

- **/**

2.97.4. Tags

- Patient

2.97.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.98. getBasicDataPatient

GET /patient/{patientId}/basicdata

2.98.1. Parameters

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

2.98.2. Responses

HTTP Code	Description	Schema
200	OK	BasicPatientDto

2.98.3. Produces

- **/**

2.98.4. Tags

- Patient

2.98.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.99. addPerson

POST /person

2.99.1. Parameters

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

2.99.2. Responses

HTTP Code	Description	Schema
200	OK	BMPersonDto

2.99.3. Consumes

- [application/json](#)

2.99.4. Produces

- [*/*](#)

2.99.5. Tags

- Person

2.99.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.100. getGenders

GET /person/masterdata/genders

2.100.1. Responses

HTTP Code	Description	Schema
200	OK	< GenderDto > array

2.100.2. Produces

- [*/*](#)

2.100.3. Tags

- Person Master Data

2.100.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.101. getIdentificationTypes

GET /person/masterdata/identificationTypes

2.101.1. Responses

HTTP Code	Description	Schema
200	OK	< IdentificationType Dto > array

2.101.2. Produces

- */*

2.101.3. Tags

- Person Master Data

2.101.4. Security

Type	Name	Scopes
apiKey	JWT	global

2.102. getPersonMinimal

GET /person/minimalsearch

2.102.1. Parameters

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

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

2.102.2. Responses

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

2.102.3. Produces

- **/**

2.102.4. Tags

- Person

2.102.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.103. getPersonalInformation

```
GET /person/{personId}/personalInformation
```

2.103.1. Parameters

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

2.103.2. Responses

HTTP Code	Description	Schema
200	OK	PersonalInformationDto

2.103.3. Produces

- `*/*`

2.103.4. Tags

- Person

2.103.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.104. setRoleUser

POST /roles

2.104.1. Parameters

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

2.104.2. Responses

HTTP Code	Description	Schema
200	OK	boolean

2.104.3. Consumes

- `application/json`

2.104.4. Produces

- `*/*`

2.104.5. Tags

- Role

2.104.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.105. addUser

POST /users

2.105.1. Parameters

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

2.105.2. Responses

HTTP Code	Description	Schema
200	OK	UserDto

2.105.3. Consumes

- `application/json`

2.105.4. Produces

- `*/*`

2.105.5. Tags

- User

2.105.6. Security

Type	Name	Scopes
apiKey	JWT	global

2.106. getUsers

GET /users/page

2.106.1. Parameters

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

2.106.2. Responses

HTTP Code	Description	Schema
200	OK	PageOfPageableUsers

2.106.3. Produces

- **/**

2.106.4. Tags

- User

2.106.5. Security

Type	Name	Scopes
apiKey	JWT	global

2.107. changeStatusAccount

PATCH /users/{id}

2.107.1. Parameters

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

2.107.2. Responses

HTTP Code	Description	Schema
200	OK	boolean

2.107.3. Consumes

- `application/json`

2.107.4. Produces

- `*/*`

2.107.5. Tags

- User

2.107.6. Security

Type	Name	Scopes
apiKey	<code>JWT</code>	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'T'HH:mm:ss.SSS'Z'"	string
cityId <i>optional</i>		integer (int32)
comments <i>optional</i>		string
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

Name	Description	Schema
identificationTypeId <i>required</i>		integer (int32)
identityVerificationStatusId <i>optional</i>		integer (int32)
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

Name	Description	Schema
street <i>optional</i>		string
typeId <i>optional</i>		integer (int32)

4.2. APersonDto

Name	Description	Schema
apartment <i>optional</i>		string
birthDate <i>required</i>	Example : "yyyy-MM-dd'T'HH:mm:ss.SSS'Z' "	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)

Name	Description	Schema
identificationNumber <i>optional</i>		string
identificationTypeId <i>required</i>		integer (int32)
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

4.3. 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

Name	Schema
city <i>optional</i>	CityDto
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
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. AllergyConditionDto

Name	Schema
categoryId <i>optional</i>	string
date <i>optional</i>	string
severity <i>optional</i>	string
snomed <i>optional</i>	SnomedDto
statusId <i>optional</i>	string
verificationId <i>optional</i>	string

4.8. AnamnesisDto

Name	Schema
allergy <i>optional</i>	< AllergyConditionDto > array
anthropometricData <i>optional</i>	AnthropometricDataDto
diagnosis <i>optional</i>	< HealthConditionDto > array
documentStatusId <i>optional</i>	string
familyHistory <i>optional</i>	< HealthHistoryConditionDto > array
immunization <i>optional</i>	< ImmunizationDto > array
medication <i>optional</i>	< MedicationDto > array

Name	Schema
notes <i>optional</i>	ObservationsDto
personalHistory <i>optional</i>	< HealthHistoryConditionDto > array
vitalSigns <i>optional</i>	VitalSignDto

4.9. AnthropometricDataDto

Name	Schema
bloodType <i>optional</i>	string
height <i>optional</i>	string
weight <i>optional</i>	string

4.10. BMPatientDto

Name	Description	Schema
apartment <i>optional</i>		string
birthDate <i>required</i>	Example : "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"	string
cityId <i>optional</i>		integer (int32)
comments <i>optional</i>		string
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)
identityVerifi cationStatusI d <i>optional</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
typeId <i>optional</i>		integer (int32)

4.11. BMPersonDto

Name	Description	Schema
apartment <i>optional</i>		string
birthDate <i>required</i>	Example : "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"	string

Name	Description	Schema
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)
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

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

4.12. BackofficeDeleteResponseOfAddress

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

4.13. BackofficeDeleteResponseOfBed

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

4.14. BackofficeDeleteResponseOfCity

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

4.15. BackofficeDeleteResponseOfClinicalSpecialty

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

4.16. BackofficeDeleteResponseOfClinicalSpecialtySector

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

4.17. BackofficeDeleteResponseOfDepartment

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

4.18. BackofficeDeleteResponseOfInstitution

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

4.19. BackofficeDeleteResponseOfProvince

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

4.20. BackofficeDeleteResponseOfRoom

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

4.21. BackofficeDeleteResponseOfSector

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

4.22. BasicDataPersonDto

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

4.23. BasicPatientDto

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

4.24. Bed

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

4.25. BedDto

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

4.26. Character

Type : object

4.27. 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)

4.28. CityDto

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

4.29. ClinicalSpecialty

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

4.30. ClinicalSpecialtyDto

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

4.31. 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)

4.32. Department

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

4.33. GenderDto

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

4.34. HealthConditionDto

Name	Schema
snomed <i>optional</i>	SnomedDto
statusId <i>optional</i>	string
verificationId <i>optional</i>	string

4.35. HealthHistoryConditionDto

Name	Schema
date <i>optional</i>	string
note <i>optional</i>	string
snomed <i>optional</i>	SnomedDto
statusId <i>optional</i>	string
verificationId <i>optional</i>	string

4.36. HealthInsuranceDto

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

4.37. IdentificationTypeDto

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

4.38. IdentityVerificationStatus

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

4.39. ImmunizationDto

Name	Schema
date <i>optional</i>	string
note <i>optional</i>	string
snomed <i>optional</i>	SnomedDto

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

4.40. 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
sisacode <i>optional</i>	string
website <i>optional</i>	string

4.41. InternmentEpisodeDto

Name	Schema
bed <i>optional</i>	BedDto
doctor <i>optional</i>	ResponsibleDoctorDto

Name	Schema
id <i>optional</i>	integer (int32)
patient <i>optional</i>	PatientDto
specialty <i>optional</i>	ClinicalSpecialtyDto

4.42. InternmentMasterDataProjection

Name	Schema
description <i>optional</i>	string
id <i>optional</i>	string

4.43. InternmentSummaryDto

Name	Schema
bed <i>optional</i>	BedDto
createdOn <i>optional</i>	string (date-time)
doctor <i>optional</i>	ResponsibleDoctorDto
id <i>optional</i>	integer (int32)
specialty <i>optional</i>	ClinicalSpecialtyDto
totalInternmentDays <i>optional</i>	integer (int32)

4.44. IterableOfAddress

Type : object

4.45. IterableOfBed

Type : object

4.46. IterableOfCity

Type : object

4.47. IterableOfClinicalSpecialty

Type : object

4.48. IterableOfClinicalSpecialtySector

Type : object

4.49. IterableOfDepartment

Type : object

4.50. IterableOfInstitution

Type : object

4.51. IterableOfProvince

Type : object

4.52. IterableOfRoom

Type : object

4.53. IterableOfSector

Type : object

4.54. JWTokenDto

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

4.55. Link

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

4.56. LoginDto

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

4.57. MapOfStringAndLink

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

4.58. MedicationDto

Name	Schema
note <i>optional</i>	string
snomed <i>optional</i>	SnomedDto

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

4.59. 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

4.60. ObservationsDto

Name	Schema
clinicalReport <i>optional</i>	string
evolution <i>optional</i>	string
extra <i>optional</i>	string
physicalExamination <i>optional</i>	string
presentIllness <i>optional</i>	string
procedure <i>optional</i>	string

4.61. PageOfAddress

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

Name	Schema
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)

4.62. 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)

Name	Schema
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)

4.63. 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)

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

4.64. PageOfClinicalSpecialty

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

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

4.65. 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)
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)

4.66. 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)

4.67. PageOfInstitution

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

4.68. PageOfPageableUsers

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

4.69. 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)

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)

4.70. PageOfRoom

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

4.71. PageOfSector

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

4.72. Pageable

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

4.73. 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

4.74. PatientDto

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

4.75. PatientSearchDto

Name	Schema
activo <i>optional</i>	boolean
idPatient <i>optional</i>	integer (int32)
person <i>optional</i>	BMPersonDto
ranking <i>optional</i>	number (float)

4.76. PersonalInformationDto

Name	Schema
address <i>optional</i>	AddressDto
birthDate <i>optional</i>	string (date)
cuil <i>optional</i>	string
email <i>optional</i>	string

Name	Schema
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

4.77. Province

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

4.78. ProvinceDto

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

4.79. RequestUserRoleDto

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

4.80. ResponseAnamnesisDto

Name	Schema
allergy <i>optional</i>	< AllergyConditionDto > array
anamnesisId <i>optional</i>	integer (int32)
anthropometricData <i>optional</i>	AnthropometricDataDto
diagnosis <i>optional</i>	< HealthConditionDto > array
documentStatusId <i>optional</i>	string
familyHistory <i>optional</i>	< HealthHistoryConditionDto > array
immunization <i>optional</i>	< ImmunizationDto > array
medication <i>optional</i>	< MedicationDto > array
notes <i>optional</i>	ObservationsDto
personalHistory <i>optional</i>	< HealthHistoryConditionDto > array
vitalSigns <i>optional</i>	VitalSignDto

4.81. ResponsibleDoctorDto

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

4.82. Room

Name	Schema
clinicalSpecialtySectorId <i>optional</i>	integer (int32)
description <i>optional</i>	string
dischargeDate <i>optional</i>	string (date)
id <i>optional</i>	integer (int32)
roomNumber <i>optional</i>	string
type <i>optional</i>	string

4.83. RoomDto

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

Name	Schema
roomNumber <i>optional</i>	string
sector <i>optional</i>	SectorDto

4.84. Sector

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

4.85. SectorDto

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

4.86. SnomedDto

Name	Schema
fsn <i>required</i>	string
id <i>required</i>	string
parentFsn <i>optional</i>	string

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

4.87. Sort

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

4.88. UserDto

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

4.89. View

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

4.90. VitalSignDto

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

Name	Schema
diastolicBloodPresure <i>optional</i>	string
heartRate <i>optional</i>	string
meanPresure <i>optional</i>	string
respiratoryRate <i>optional</i>	string
systolicBloodPresure <i>optional</i>	string
temperature <i>optional</i>	string