

API Documentation – FHIR I/O EMS Patient Data Platform:

1. Authentication

```
@RequestMapping(value="", method=POST, consumes="application/json")
public ResponseEntity<Token> authenticate(@RequestBody
Authentication auth)
```

Usage:

E.g. curl -H "Content-Type: application/json" -X POST -d
'{"userName":"emspersonnel","password":"password"}' http://localhost:8080/api/authentication

Parameters:

auth - Authentication with username and password

Returns:

Token - Generated Authentication Token

2. Get Emergency

```
@RequestMapping(value="/{emergencyUuid}", method=GET)
public ResponseEntity<Emergency>
getEmergency(@RequestParam(value="verbose",required=false,defaultValue="false")
Boolean isVerbose,
@PathVariable
Integer emergencyUuid)
```

Usage:

E.g. curl -X GET http://localhost:8080/api/emergency/1

Parameters:

isVerbose - Boolean

emergencyUuid - Integer

Returns:

Emergency - Emergency corresponding to emergencyUuid

3. Create Emergency

```
@RequestMapping(value="",
method=POST,
consumes="application/json")
public ResponseEntity<Emergency> createEmergency(@RequestBody
```

Emergency emergency)

Usage:

E.g. curl -H "Content-Type: application/json" -X POST -d '{"emergencyTitle":"New Emergency Title","pickupLocation":"5599 Ice Town Dr.", "patient":{"firstName":"Katheryn","lastName":"Adams"},"organizationUuid":1}' http://localhost:8080/api/emergency

Parameters:

emergency - Emergency object having details as given in usage

Returns:

Emergency - Created Emergency

4. Update Emergency

```
@RequestMapping(value="",  
    method=PUT,  
    consumes="application/json")  
public ResponseEntity<Emergency> updateEmergency(@RequestBody  
Emergency emergency)
```

Usage:

E.g. update organizationId: curl -H "Content-Type: application/json" -X PUT -d '{"emergencyUuid":1,"emergencyTitle":"Student Fainted in Class","pickupLocation":"1122 Fowler St. 30309","patient":null,"possiblePatientUuids":[1,2],"possiblePatients":[],"identifiedPatientUuid":null,"emergencyState":"PENDING","organizationUuid":2}' http://localhost:8080/api/emergency

Parameters:

emergency - Emergency having details as given in usage

Returns:

Emergency - Updated Emergency

5. Delete Emergency

```
@RequestMapping(value="/{emergencyUuid}",  
    method=DELETE)  
public ResponseEntity<Emergency> deleteEmergency(@PathVariable Integer  
emergencyUuid)
```

Usage:

E.g. curl -X DELETE http://localhost:8080/api/emergency/1

Parameters:

emergencyUuid - Integer

Returns:

Emergency which is deleted

6. Get Patient

```
@RequestMapping(value="/{patientUuid}",  
                 method=GET)  
public ResponseEntity<Patient> getPatient(@PathVariable  
Integer patientUuid)
```

Usage:

E.g. curl -X GET http://localhost:8080/api/patient/1

Parameters:

patientUuid - Integer

Returns:

Patient corresponding to the given patientUuid

7. Update patient

```
@RequestMapping(value="", method=PUT,  
                 consumes="application/json")  
public ResponseEntity<Patient> updatePatient(@RequestBody  
Patient patient)
```

Usage:

E.g. curl -H "Content-Type: application/json" -X PUT-d
'{"patientUuid":"1","patientNotes":"urgent"}' http://localhost:8080/api/patient/1

Parameters:

patient - Patient object having details as given in usage

Returns:

Patient - Updated Patient

8. Create Note

```
@RequestMapping(value="",  
    method=POST,  
    consumes="application/json")  
public ResponseEntity<Note> createNote(@RequestBody  
    Note note)
```

Usage:

E.g. curl -H "Content-Type: application/json" -X POST -d '{"patientUuid":"1e19bb7a-d990-4924-9fae-be84f19c53c1","medicalNote":"diarrhea", "systolicPressrure":"100", "diastolicPressrure":"80","bpm":"60"}' http://localhost:8080/api/patient

Parameters:

note - Note object having details as given in usage

Returns:

Note - Created Note

9. Get User

```
@RequestMapping(value="/{userUuid}",  
    method=GET)  
public ResponseEntity<User> getUser(@PathVariable Integer userUuid)
```

Usage:

E.g. curl -X GET http://localhost:8080/api/user/1

Parameters:

userUuid - Integer

Returns:

User - User object corresponding to the given userUuid

10. Create User

```
@RequestMapping(value="",  
    method=POST,  
    consumes="application/json")  
public ResponseEntity<User> createUser(@RequestBody User user)
```

Usage:

E.g. curl -H "Content-Type: application/json" -X POST -d
'{"userName":"newuser","password":"mypassword","firstName":"Tim","lastName":"Classyman","organization":{"name":"New Organization"}}' http://localhost:8080/api/user New Org User
(Created by Super User): E.g. curl -H "Content-Type: application/json" -X POST -d
'{"userName":"newuser","password":"mypassword","firstName":"Tim","lastName":"Classyman","organizationUuid":1}' http://localhost:8080/api/user

Parameters:

user - User object having details as given in usage

Returns:

User - Created User

11. Get Patient Medications

```
@RequestMapping(value="/patient/{patientUuid}/medications",  
    method=GET,  
    produces="application/json")
```

@ResponseBody

```
public Medications getPatientMedications(@PathVariable(value="patientUuid")  
String patientUuid)
```

Usage:

E.g. http://localhost:8080/api/patient/1e19bb7a-d990-4924-9fae-be84f19c53c1/medications

Parameters:

patientUuid - String

Returns:

List of Medications for the given patientUuid with details in the format:

```
[ {  
  "medicationUuid" : "702b591b-ec66-4e2e-b37a-b622b1cc7a47",  
  "status" : "Active",  
  "name" : "Nitroglycerin 0.4 MG/ACTUAT [Nitrolingual]",  
  "system" : "http://www.nlm.nih.gov/research/umls/rxnorm",  
  "code" : "564666",  
  "description" : "Nitroglycerin 0.4 MG/ACTUAT [Nitrolingual]",  
  "conditionName" : "Myocardial Infarction",  
  "conditionUuid" : "59a0520-1db9-4636-a72f-967346c10bc4"  
}, {  
  "medicationUuid" : "702b591b-ec66-4e2e-b37a-b622b1cc7a47",  
  "status" : "Stopped",
```

```

"name" : "Nitroglycerin 0.4 MG/ACTUAT [Nitrolingual]",
"system" : "http://www.nlm.nih.gov/research/umls/rxnorm",
"code" : "564666",
"description" : "Nitroglycerin 0.4 MG/ACTUAT [Nitrolingual]",
"conditionName" : "Coronary Heart Disease",
"conditionUuid" : "c69f2f4-d1bd-4d36-93e8-3f449fce03cb"
} ]

```

12. Get Patient Conditions

```

@RequestMapping(value="/patient/{patientUuid}/conditions",
    method=GET,
    produces="application/json")
@ResponseBody
public Conditions getPatientConditions(@PathVariable(value="patientUuid")
String patientUuid)

```

Usage:

<http://localhost:8080/api/patient/1e19bb7a-d990-4924-9fae-be84f19c53c1/conditions>

Parameters:

patientUuid - String

Returns:

List of Conditions for the given patientUuid with details in the format:

```

[ {
"conditionUuid" : "624abe6-8825-4438-966d-da3b93201dca",
"clinicalStatus" : "Active",
"verificationStatus" : "Confirmed",
"name" : "History of myocardial infarction (situation)",
"system" : "http://snomed.info/sct",
"code" : "399211009",
"onsetDateTime" : "1959-01-12T09:44:47-05:00",
"assertedDate" : "Mon Jan 12 00:00:00 CST 1959"
}, {
"conditionUuid" : "c69f2f4-d1bd-4d36-93e8-3f449fce03cb",
"clinicalStatus" : "Active",
"verificationStatus" : "Confirmed",
"name" : "Coronary Heart Disease",
"system" : "http://snomed.info/sct",
"code" : "53741008",
"onsetDateTime" : "1957-05-20T22:01:32-04:00",

```

```
"assertedDate" : "Mon May 20 00:00:00 CDT 1957"  
}, {  
  "conditionUuid" : "59a0520-1db9-4636-a72f-967346c10bc4",  
  "clinicalStatus" : "Active",  
  "verificationStatus" : "Confirmed",  
  "name" : "Myocardial Infarction",  
  "system" : "http://snomed.info/sct",  
  "code" : "22298006",  
  "onsetDateTime" : "1959-01-12T09:44:47-05:00",  
  "assertedDate" : "Mon Jan 12 00:00:00 CST 1959"  
}]
```