MOST Documentation

Release 1.0alpha

CRS4

May 09, 2017

Contents

Welcome to the documentation for most-medicalrecord 1.0alpha.

This package, part of the MOST project <http://github.com/crs4/most>, is developed to provide management of clinical records of patients. The library is developed as a Django application. The clinical data are represented as openEHR archetype serialized using JSON and are managed using pyEHR

The package offer a REST API to handle patients' electronic health records. An EHR for a patient is composed of ehr records. Notice that the package doesn't handle patients' demographics, but it gives the possibility to map an external demographic database id with the internal patient id.

CHAPTER 1

Contents:

MedicalRecord REST API

Table of Contents

Common Responses

Error reponses are the same for all the methods exposed by the REST API, response has a *SUCCESS* field with value false and an *ERROR* field with a short description of the occurred error. An example is the following:

```
{"success": false,
 "data": {
      "error": "<error_message>",
      "code": "<error_code>"
 }
}
```

Error Codes

The possible error codes are:

- 101 - No token provided: the request misses an OAuth2 token
- 102 - Token doesn"t exist: the OAauth2 token used for the request is not valid
- 501 - Patient doesn"t exist: the id of the patient is not recognized by the system
- 502 - Missing parameters: the request misses some required parameter

Patient

POST /patients/

Create a patient with demographics and ehr id. If the the ehr id is not specified, the server creates it automatically. The demographic_uuid, on the other hand is mandatory and is the id of the patient in an external demographic service (e.g., most-demographics)

Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication
- **demographic_uuid** – identification for demographics data. It should be an id related to a patient of an external demographic service
- **ehr_uuid** – identification for ehr data

Response Headers

- Content-Type – application/json

Example of correcy response:

```
{
    "success": true,
    "patient": {
        "uuid": "drpp32jnmapx3roaf5j7k46saj2ynuba",
        "demographic_uuid": "nlpwv5wcqrqlqr2u3eiodo3qtjghnres"
    }
}
```

GET /patients/

Get the patients related to a taskgroup. The taskgroup is associated with the OAuth2 access token which has to be sent in input.

Query Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication

Response Headers

- Content-Type – application/json

Example of correct responses:

```
{
    "success": true,
    "patients": [{
        "uuid": "drpp32jnmapx3roaf5j7k46saj2ynuba",
        "demographic_uuid": "nlpwv5wcqrqlqr2u3eiodo3qtjghnres"
    }]
}
```

GET /patients/ (**string:** *patient_uuid*) **/**

Get the patient with the specified patient_uuid

Query Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication

Response Headers

- Content-Type – application/json

Example of correct responses:

```
{
    "success": true,
    "patient": {
        "uuid": "drpp32jnmapx3roaf5j7k46saj2ynuba",
        "demographic_uuid": "nlpwv5wcqrqlqr2u3eiodo3qtjghnres"
    }
}
```

PUT /patients/

Update a patient with a new demographic id

Query Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication
- **patient_uuid** – the patient id of the patient to update
- **demographic_uuid** – the new demographic_uuid value

Response Headers

- Content-Type – application/json

Example of correct response:

```
{
    "success": true,
    "patient": {
        "uuid": "drpp32jnmapx3roaf5j7k46saj2ynuba",
        "demographic_uuid": "new_demographic_id"
    }
}
```

DELETE /patients/

Delete a patient

Query Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication
- **patient_uuid** – the patient id of the patient to delete

Response Headers

- Content-Type – application/json

Example of correct response:

```
{
    "success": true,
    "patient": {
        "uuid": "drpp32jnmapx3roaf5j7k46saj2ynuba",
        "demographic_uuid": "new_demographic_id"
    }
}
```

Demographic

GET **/demographic/** (**string:** *demographic_uuid*) **/**
Get the patient with the specified demographic uuid

Query Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication

Response Headers

- Content-Type – application/json

Example of correct responses:

```
{
    "success": true,
    "patient": {
        "uuid": "drpp32jnmapx3roaf5j7k46saj2ynuba",
        "demographic_uuid": "nlpwv5wcqrqlqr2u3eiodo3qtjghnres"
    }
}
```

EHR

GET **/ehr/** (**string:** *patient_uuid*) **/**
Return a the patient's ehr record for the patient identified by patient_uuid

Query Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication

Response Headers

- Content-Type – application/json

Example of correct responses:

```
{
    "success": true,
    "record": {
    "record_id": "gnaiibv2pvca4pufllijbacaengt4i5k",
        "active": true,
        "ehr_records": [],
        "creation_time": 1493994347.727665,
        "last_update": 1493994347.727665
    }
}
```

POST **/ehr/** (**string:** *patient_uuid*) **/**
Creates the ehr for the patient specified by patient_uuid

Query Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication

Response Headers

- Content-Type – application/json

Example of correct responses:

```
{
    "success": true,
    "record": {"record_id": "gnaiibv2pvca4pufllijbacaengt4i5k",
        "active": true,
        "ehr_records": [],
        "creation_time": 1493994347.727665,
        "last_update": 1493994347.727665
    }
}
```

DELETE /ehr/ (**string:** *patient_uuid*) **/**

Delete the ehr for the patient specified by patient_uuid

Query Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication
- **delete_method** – it can be "hide" (default) or "delete". In the first case the record is only deactivated in the second it is actually deleted

Response Headers

- Content-Type – application/json

Example of correct responses:

```
{
    "success": true,
    "record": {"record_id": "gnaiibv2pvca4pufllijbacaengt4i5k",
        "active": false,
        "ehr_records": [],
        "creation_time": 1493994347.727665,
        "last_update": 1493994347.727665
    }
}
```

POST /ehr/ (**string:** *patient_uuid*) **/records/**

Save a new ehr record to the ehr of the patient with id patient_uuid. The record id is created by the system. The data of the request must be a JSON encoded openEHR archetype.

Request Headers

- Content-Type – application/json

Response Headers

- Content-Type – application/json

Example of correct request data:

```
{
    "archetype_class": "openEHR.TEST-EVALUATION.v1",
    "archetype_details": {
        "at0001": "val1",
        "at0002": "val2"
    }
}
```

Example of correct response:

```
{
    "record": {
        "ehr_data": {
            "archetype_class": "openEHR.TEST-EVALUATION.v1",
            "archetype_details": {
                "at0001": "val1",
                "at0002": "val2"
            }
        },
        "creation_time": 1399905956.765149,
        "last_update": 1399905956.765149,
        "record_id": "9a30f6b6a36b49c6b16e249ef35445eb",
        "active": true,
        "version": 1,
    },
    "success": true
}
```

POST /ehr/ (**string:** *patient_uuid*) **/records/**
string: *record_uuid***/** Same as the previous function but in this case the record_uuid is provided by the client.

GET /ehr/ (**string:** *patient_uuid*) **/records/**
string: *record_uuid***/** Return the ehr record identified by record_uuid from the ehr of patient specified by patient_uuid

Query Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication

Response Headers

- Content-Type – application/json

Example of correct response:

```
{
    "record": {
        "ehr_data": {
            "archetype_class": "openEHR.TEST-EVALUATION.v1",
            "archetype_details": {
                "at0001": "val1",
                "at0002": "val2"
            }
        },
        "creation_time": 1399905956.765149,
        "last_update": 1399905956.765149,
        "record_id": "9a30f6b6a36b49c6b16e249ef35445eb",
        "active": true,
        "version": 1,
    },
    "success": true
}
```

DELETE /ehr/ (**string:** *patient_uuid*) **/records/**
string: *record_uuid***/** Delete the ehr record identified by record_uuid from the ehr of patient specified by patient_uuid

Query Parameters

- **access_token** – the OAuth2 access token returned by the server after the authentication

Response Headers

- Content-Type – application/json

Example of correct response:

```
{
    "message": "EHR record with ID cf629c7c51b740fb9776f8c4cc51f293 successfully␣
↪hidden",
    "success": true
}
```

Examples

You can find examples of REST API usage in notebook directory of the repository on GitHub To run the examples you will need to install pyEHR and then launch the most-medicalrecords server

pyEHR installation

To install pyEHR you can either install it locally or use the docker image provided by pyEHR here

Run and configure the server

To run the server launch the following commands from the main repository dir:

```
make devel # it will install dependency projects
make sync # initiliaze the database
make run # it will launch the web/rest server on 0.0.0.0:9000
```

After this, you will need to connect to the admin page and, after logging in using admin/admin as credentials, configure the address of the pyEHR in the section Medicalrecords -> Configurations.

NOTE: the OAuth configuration is already loaded in the database and configuration is the one used by the notebooks

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

HTTP Routing Table

/demographic

```
GET /demographic/(string:demographic_uuid)/,
       6
```

/ehr

```
GET /ehr/(string:patient_uuid)/,6
GET /ehr/(string:patient_uuid)/records/(string:record_uuid)/,
       8
POST /ehr/(string:patient_uuid)/,6
POST /ehr/(string:patient_uuid)/records/,
       7
POST /ehr/(string:patient_uuid)/records/(string:record_uuid)/,
       8
DELETE /ehr/(string:patient_uuid)/,7
DELETE /ehr/(string:patient_uuid)/records/(string:record_uuid)/,
       8
```

/patients

```
GET /patients/,4
GET /patients/(string:patient_uuid)/,4
POST /patients/,4
PUT /patients/,5
DELETE /patients/,5
```