Welcome to emNutt

emNutt - mHero Connector - is a mAcm implementation.

Last update: May 13, 2020

This page is still under construction

Last update: May 13, 2020

Prerequisites

Required

1. A FHIR compatible server such as hapi
2. Elasticsearch instance
3. Kibana

Optional

1. OpenHIM - Only when you want to use emNutt as a mediator running behind openHIM

Communication channels

To use emNutt, you will need to install/configure any or all of the below supported communication channels

1. Rapidpro

Last update: May 13, 2020

Installation

Clone the repository

```
git clone https://github.com/intrahealth/emNutt.git
```

Enter the server directory and install node packages.

```
cd emNutt/server && npm install
```

Copy and edit the configuration file to your liking.

```
cp config/config_development_template.json config/
config_development.json
```

Start server

Before you start server, you may need to adjust some configuration variables, see Configuration page

```
npm start
```

Last update: May 13, 2020

Prerequisites

1. Docker
2. Docker Compose

Last update: May 13, 2020

Docker installation

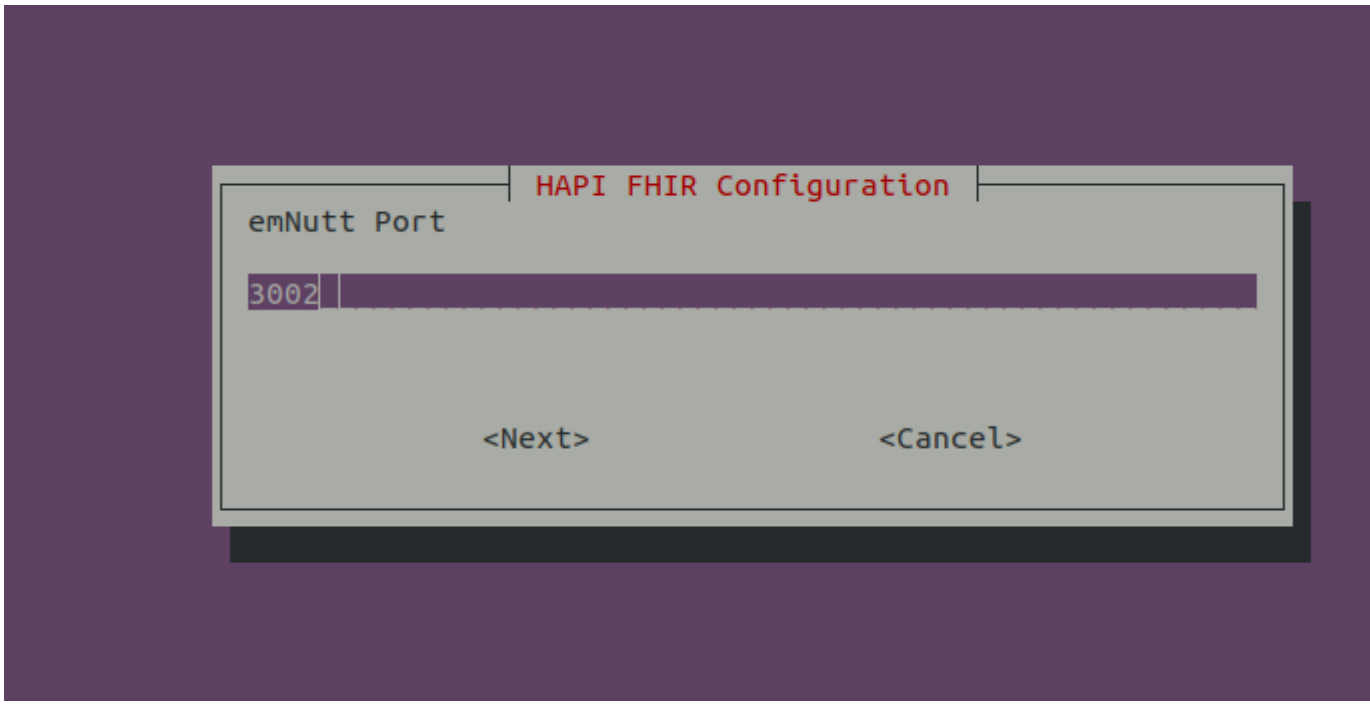
Clone the repository

```
git clone https://github.com/intrahealth/emNutt.git
```

```
cd emNutt && sudo ./install.sh
```

The install script will ask some few questions as below

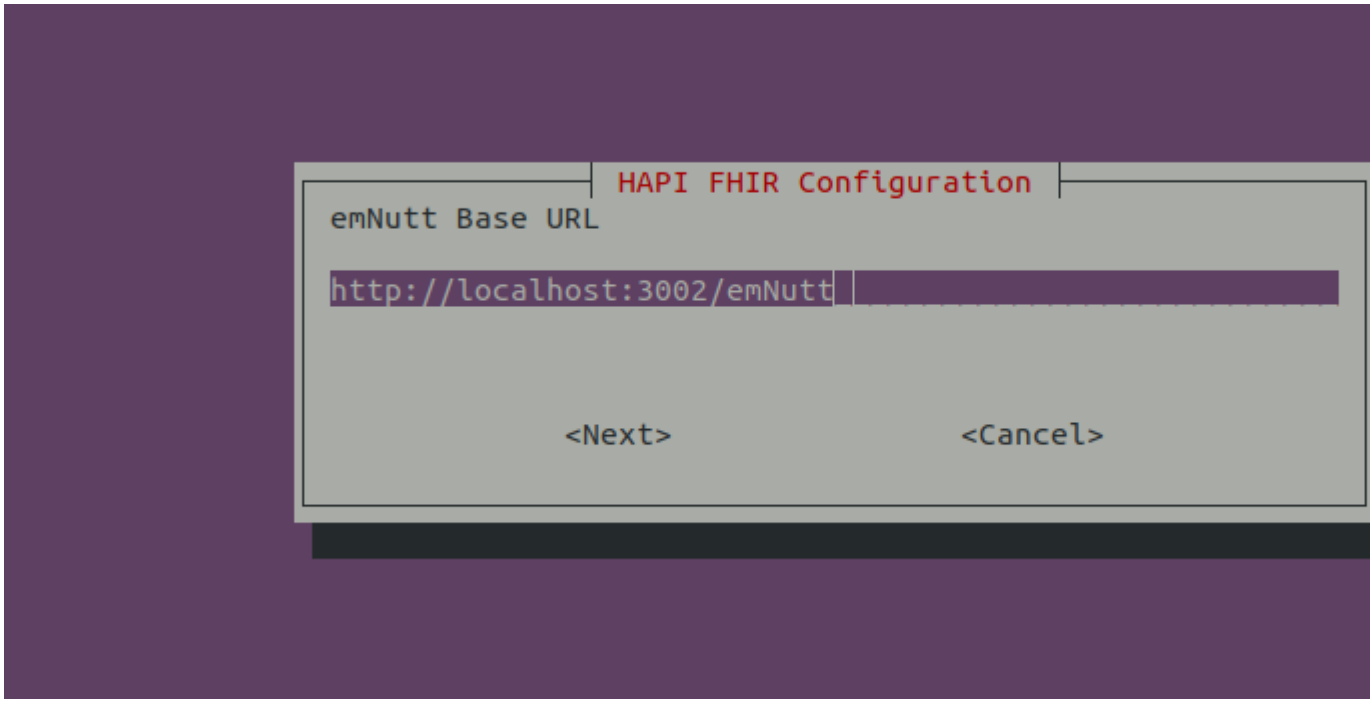
emNutt port



port number that emNutt server will be listening

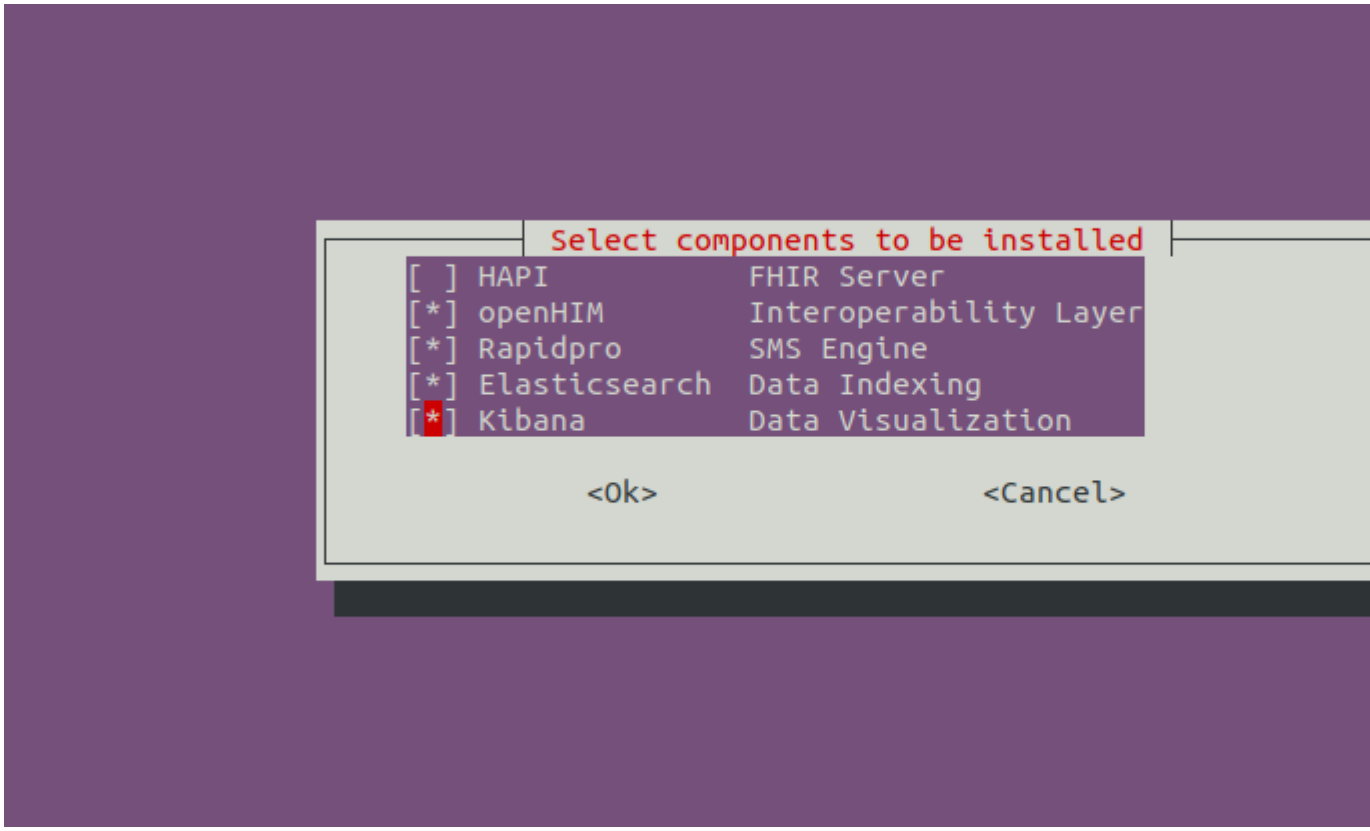
emNutt Base URL

Put base URL that people use to access emNutt. if emNutt is behind any proxy, then it should be t
access emNutt through proxy



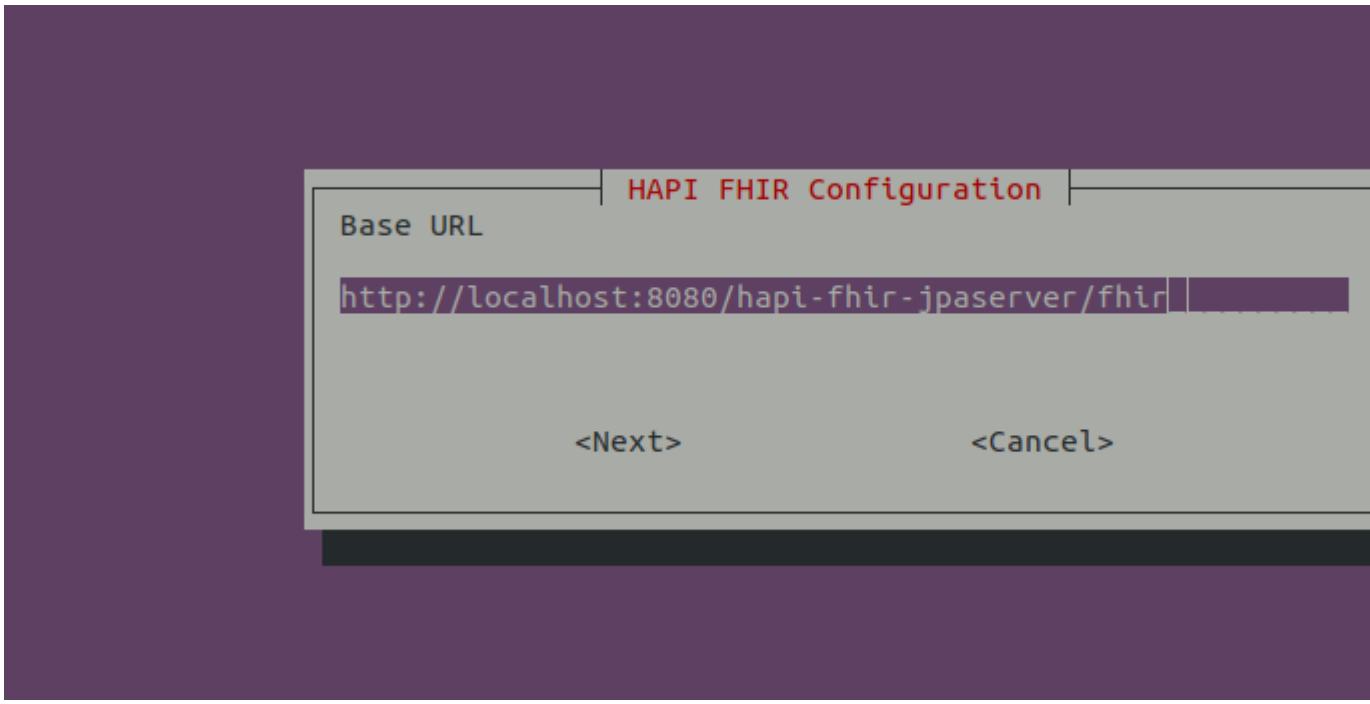
emNutt Components

This allows you to select what components that should be installed locally, the installer will ask fo
components



url of unselected components

You will be asked for the url, username and password of any component that were not selected fr



The installer will now pull all docker images for selected components, install them and start the s
generate two files,

1. docker_env_vars - This has all environment variables that can be used to configure all the em components and emNutt itself. Change them to twist some behaviours of emNutt
2. start.sh - Use this script to start emNutt.

```
sudo ./start.sh
```

Last update: May 13, 2020

Configuration

Below are emNutt configuration parameters

App Config

```
"app": {
  "port": 3002,
  "installed": false,
  "baseURL": "http://localhost:3002/emNutt",
  "contactGroupsSource": "pos"
}
```

- app.port - is the port number for emNutt
- app.installed - when false, emNutt will load all default settings and set app.installed to true. If you want to reload default settings then set this to false at any time.
- app.baseURL - is the base URL that is used to access the emNutt server, if emNutt is behind any proxy, then it should be the address used to access emNutt through proxy
- app.contactGroupsSource - tells emNutt the system that is used to manage contacts groups, values can either be pos or the name of the communication channel like rapidpro. If the value is pos then contact groups will be managed by Point of Service system like iHRIS, openMRS, DHIS2 etc, other wise then contacts groups will be managed through communications channel i.e rapidpro

Mediator Config

```
"mediator": {
  "api": {
    "username": "root@openhim.org",
    "password": "openhim-password",
    "apiURL": "https://localhost:8080",
```

```
    "routerURL": "http://localhost:5001",
    "trustSelfSigned": true,
    "urn": ""
  }
  "register": false
}
```

- mediator.api.username is the openHIM username for emNutt to register itself as a openHIM mediator
- mediator.api.password is the openHIM password
- mediator.api.apiURL is the openHIM API URL
- mediator.api.routerURL is the openHIM URL used to send to access mediator channels,default port is 5001 for http and 5000 for https
- mediator.register controls on whether emNutt should be used as a openHIM mediator or not, if set to false then emNutt will be used as a stand alone app.

Rapidpro Config

```
"rapidpro": {
  "baseURL": "http://app.rapidpro.io",
  "token": "1c443695d3bdhgeaf3e89b52dyg56e2886fa8uh2",
  "syncAllContacts": false
}
```

- rapidpro.baseURL is the rapidpro base URL that is used by emNutt for starting workflows, sync contacts etc
- rapidpro.token is the security token that can be obtained from inside rapidpro
- rapidpro.syncAllContacts - if set to true then emNutt will sync all contacts from iHRIS or DHIS2 etc and save them to Rapidpro. If set to false then only contacted contacts will be saved into Rapidpro.

FHIR Server Config

```
"macm": {
  "baseURL": "http://localhost:8080/fhir",
  "username": "",
  "password": ""
}
```

- macm.baseURL - This is the base URL for the FHIR server
- macm.username - This is the username for the FHIR server
- macm.password - This is the password for the FHIR server

Elasticsearch Config

```
"elastic": {
  "baseURL": "http://localhost:9200",
  "username": "",
  "password": ""
  "max_compilations_rate": "10000/1m"
}
```

- elastic.baseURL - Is the base URL of Elasticsearch server
- elastic.username - Is the elasticsearch username
- elastic.password - Is the elasticsearch password
- elastic.max_compilations_rate - this sets maximum scripts (requests) per minute that ES can execute, default is 15/minute which doesnt work well with emNutt

Kibana Config

```
"kibana": {
  "baseURL": "http://localhost:5601",
  "username": "",
  "password": ""
}
```

- kibana.baseURL - Is the base URL for Kibana
- kibana.username - Is the kibana username

- kibana.password - Is the kibana password

Start server

```
npm start
```

Last update: May 13, 2020

End Points

CommunicationRequest

```
/emNutt/fhir/CommunicationRequest - POST
```

Use this end point to POST CommunicationRequest (sending Messages or Starting a workflow)
Below is a sample CommunicationRequest - When emNutt and POS are using the same FHIR Server i.e emNutt knows where to get Practitioner/P6194

```
{
  "resourceType": "CommunicationRequest",
  "payload": [{
    "contentAttachment": {
      "url": "b7a4770c-d034-4055-9f21-b17632ef311e"
    }
  }],
  "recipient": [{
    "reference": "Practitioner/P6194"
  }, {
    "reference": "Practitioner/P8699"
  }]
}
```

OR (This is mostly when emNutt and POS are using different FHIR server - i.e emNutt does not know how to resolve Practitioner/P6194

```
{
  "resourceType": "CommunicationRequest",
  "contained": [{
    "resourceType": "Practitioner",
    "id": "P6194",
    "name": [{
      "use": "official",
      "text": "Jousaesto Joutousle",
      "family": "Joutousle",
      "given": [
        "Jousaesto"
      ]
```

```
    }],
    "telecom": [{
      "system": "phone",
      "value": "+27-555-8344-23"
    }]
  }, {
    "resourceType": "Practitioner",
    "id": "P8699",
    "name": [{
      "use": "official",
      "text": "Taraeceaf Thiuaewiasou",
      "family": "Thiuaewiasou",
      "given": [
        "Taraeceaf"
      ]
    }],
    "telecom": [{
      "system": "phone",
      "value": "+27-555-9621-44"
    }]
  }],
  "payload": [{
    "contentAttachment": {
      "url": "b7a4770c-d034-4055-9f21-b17632ef311e"
    }
  }],
  "recipient": [{
    "reference": "#P6194"
  }, {
    "reference": "#P8699"
  }]
}
```

payload.contentAttachment.url is the workflow id to be started

syncWorkflows - GET

```
/emNutt/syncWorkflows - GET
```

Use this end point to synchronize workflows between emNutt and Rapidpro

Getting workflows from emNutt

```
/emNutt/fhir/Basic?_profile=http://mhero.org/fhir/
StructureDefinition/mHeroWorkflows - GET
```

Use this end point to get all workflows from emNutt

syncWorkflowRunMessages

```
/emNutt/syncWorkflowRunMessages - GET
```

Use this end point to synchronize Messages between rapidpro and emNutt

syncContacts - GET

```
/emNutt/syncContacts - GET
```

Use this end point to sync contacts between emNutt and Rapidpro.

syncContacts - POST

```
/emNutt/syncContacts - POST
```

Use this end point to sync contacts between POS i.e iHRIS, DHIS2 openMRS etc and rapidpro. The request body must be a FHIR bundle of contacts i.e Practitioner or Person or Patient resource. This is especially when emNutt and POS are using different FHIR Servers.

syncContactsGroups

```
/emNutt/syncContactsGroups
```

Use this end point to sync contacts groups between emNutt and communication channels i.e rapidpro and emNutt, this will depend with the system that is configured to manage contacts groups. if POS is set as a system to manage

contacts groups then contact groups will be taken from POS and saved to rapidpro and viceversa

cacheFHIR2ES

```
/emNutt/cacheFHIR2ES
```

Use this end point to cache FHIR data into elasticsearch for visualizstion

Getting any resource

```
/emNutt/fhir/:resource?/:id? - GET
```

Use to get resource data from emNutt i.e /emNutt/fhir/Communication (lists all communications) OR /emNutt/fhir/Communication/123 retrieves communication that has ID 123

Last update: May 13, 2020