

CDA Generator Module Documentation

Documentation

Overview

The aim of this project is to produce CDA documents from OpenMRS data objects based upon user requests.

Clinical Document Architecture(CDA) is an XML-based standard for exchange of health information records, it is meant to ensure interoperability.

The CDA generator module uses the [MDHT API](#) to produce and validate CDA Documents.

This document will cover following information :-

1. What types of CDA Documents or Messages are currently supported by the CDA Generator module
2. Design of the Module
3. How users can extend on CDA Generator module
4. How we are populating sections of APHP CDA Message
5. [Installation of the Module](#)
6. Potential Improvements
7. Screenshots
8. Quick Demonstration
9. Sample APHP Documents produced by the CDA Generator module

What types of CDA Documents or Messages are currently generated by the CDA Generator module ?

This release of module only supports the generation of an [APHP CDA Document type](#). However, developers are free to write up new CDA documents which can be added to the module. There are many types of CDA documents (APHP, APS, APL ,APE, e.t.c) that may be found on the [IHE profile page](#).

Design of the Module

During this summer our focus was to generate APHP CDA message, but there are many types of IHE profile or CDA documents types.

However, our module was designed to be configured in a manner that allows users to write their own CDA message handlers.

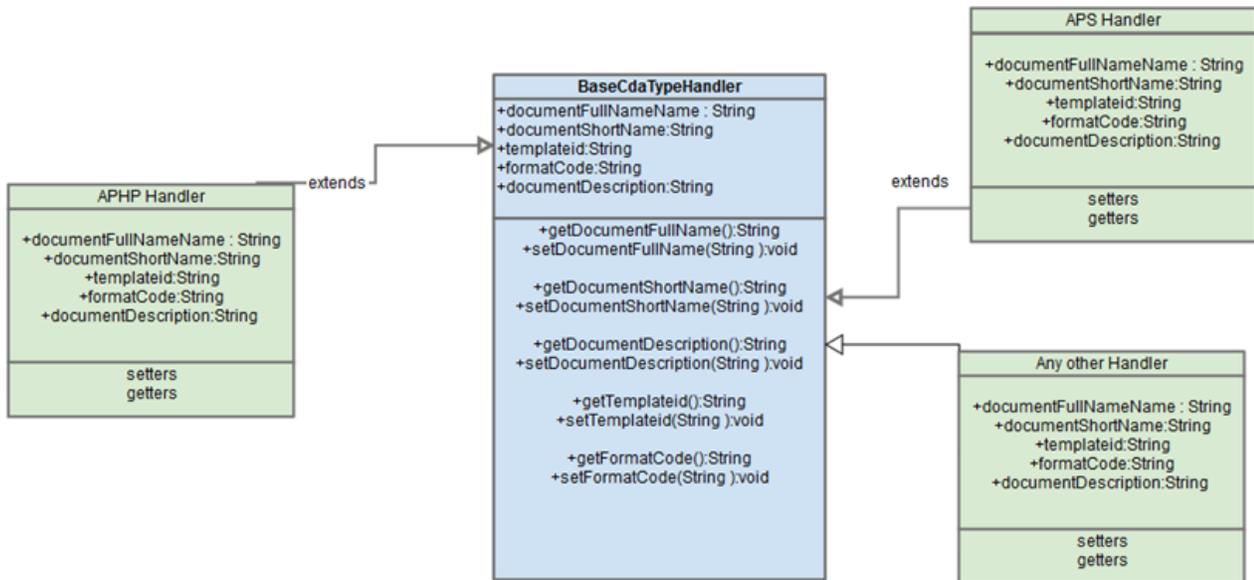
We have designed our module to ensure that our work is flexible, and can be extended further. If someone wants to add new CDA type documents other than APHP, our module accommodates this change. Our design consists of an hierarchy of classes (or as we call them, an hierarchy of handlers. BaseCdaTypeHandler serves as the base class which contains all attributes necessary for CDA document. All other handlers extend this BaseCdaTypeHandler (example APHP handler, APS handler or some other handler).

Our module uses the BaseCDATypeHandler to make our work as generic as possible.

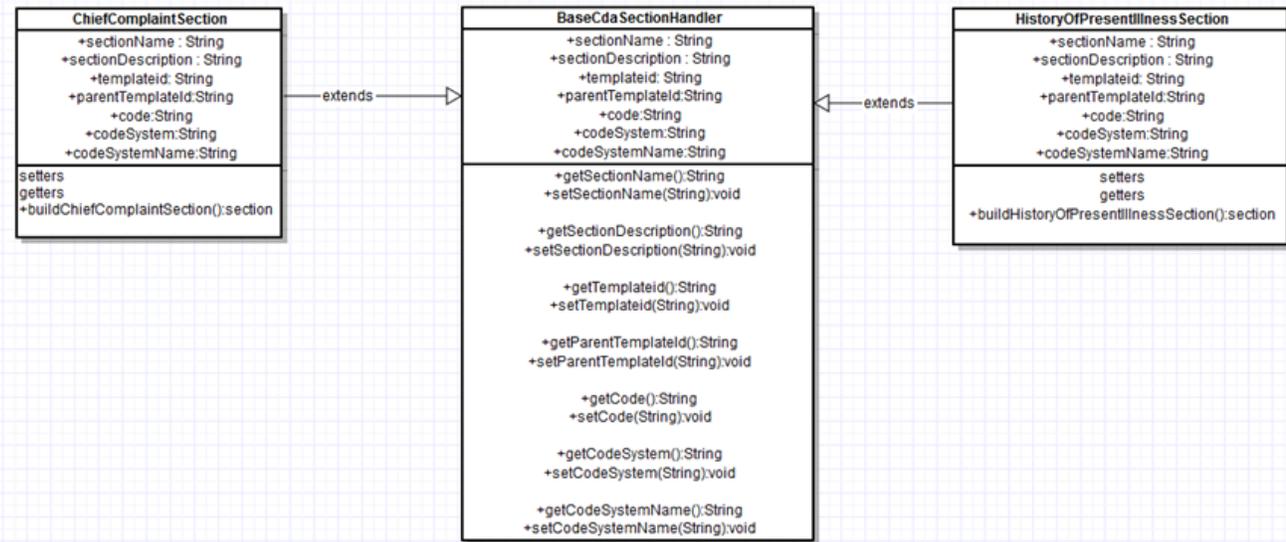
If we want to add a new CDA Document Type, all we need to do is create a class, extend from this BasCdaTypeHandler class and use setter methods to add values. Its as simple as that!

The same design approach is followed for Section as well. There we have BaseCdaSectionhandler and child handlers extending upon it.

Class Diagrams



The class diagram above depicts how different Document handler classes extend on the BaseCdaTypeHandler.



How can users extend on the CDA Generator module?

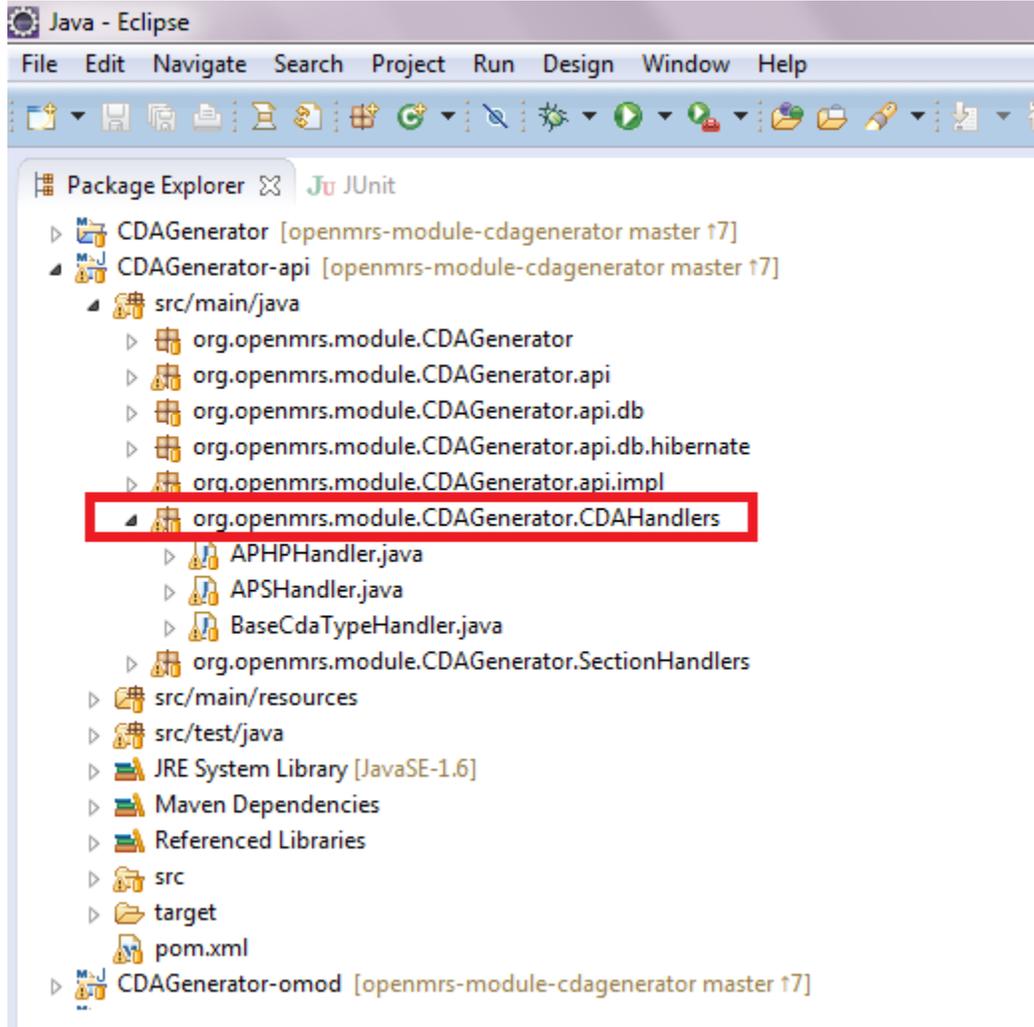
Users can extend the module by adding new CDA document Types or Editing existing ones (not recommended). Each CDA document consist of few sections and users can also add new sections to the module or Edit existing ones. The Editing/Adding/Deleting of Sections and CDA document types can be done programmatically through code not the User Interface.

1) Steps to Add a new CDA Document Type

- fork our module from gitub [here](#) .
- open "CDAGenerator-api" folder.
- In src/main/java package try to locate "org.openmrs.module.CDAGenerator.CDAHandlers" package.
- Now in this "org.openmrs.module.CDAGenerator.CDAHandlers" package, Create class for new CDA Document Type you need and Note that this class must extend BaseCdaTypeHandler class.

- Add details about documents either using setter methods or adding in a constructor. Thats it and new CDA document type is ready! .

Image showing location to add new CDA Document Type



Sample Code

Code showing how New CDA Document type Class Must look like

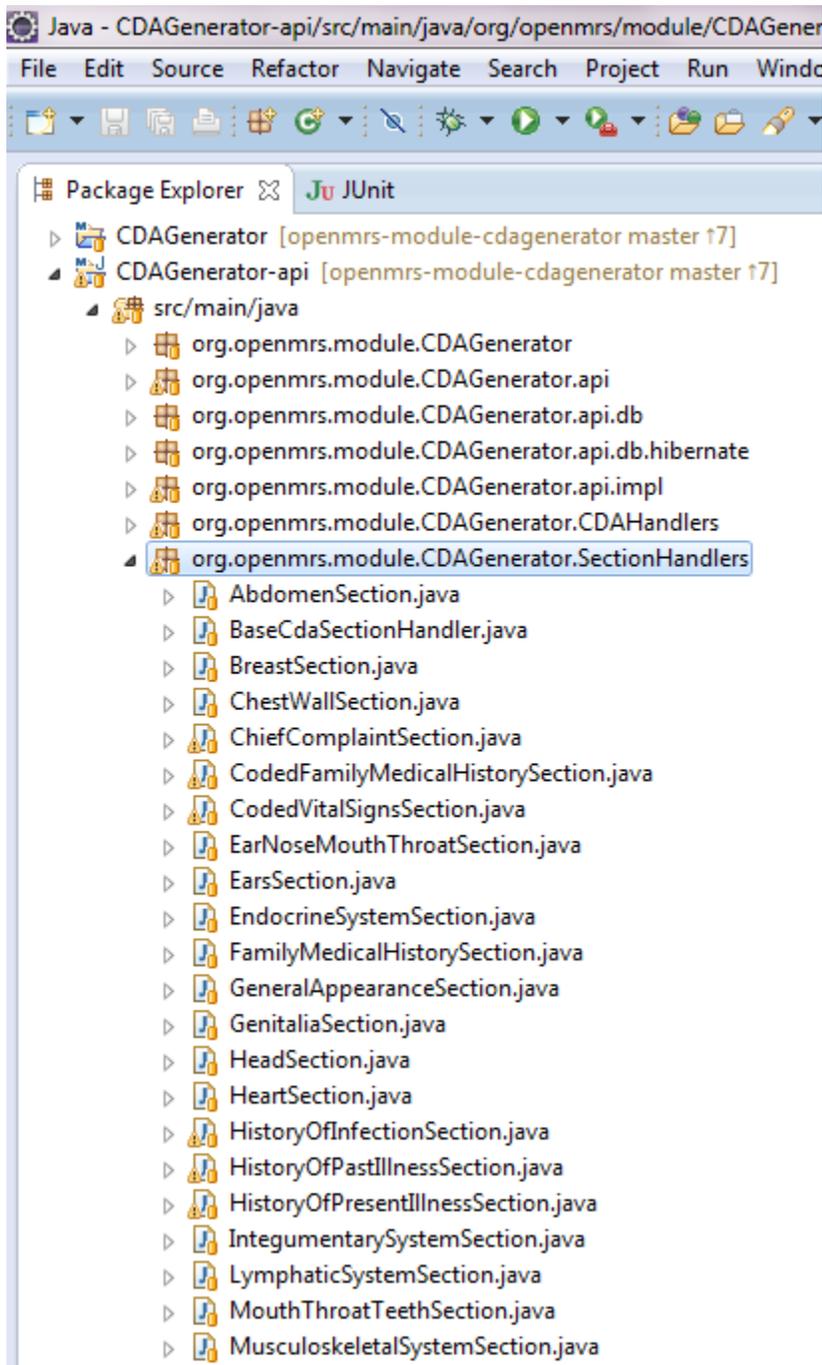
```
package org.openmrs.module.CDAGenerator.CDAHandlers;

public class MyCDADocument extends BaseCdaTypeHandler
{
    public MyCDADocument()
    {
        this.documentFullName="MyCDADocument";
        this.documentShortName="MCD";
        this.documentDescription="to add new cda document type";
        this.templateid="some free text";//required attribute,not null
        this.formatCode=null; //optional attribute
    }
}
```

2) Steps to Add a new Section

- fork our module from gitub [here](#) .
- open "CDAGenerator-api" folder.
- In src/main/java package try to locate "org.openmrs.module.CDAGenerator.SectionHandlers" package.
- Now in this "org.openmrs.module.CDAGenerator.SectionHandlers" package, Create class for new section you need and Note that this class must extend BaseCdaSectionHandler class.
- Add details about section either using setter methods or adding in a constructor. Thats it and new Section is ready! .

Image showing location to add new sections



Sample Code

Code showing how New Section Class Must look like

```
package org.openmrs.module.CDAGenerator.SectionHandlers;
public class MySection extends BaseCdaSectionHandler
{

public ChiefComplaintSection()
{
    this.sectionName="My Section";
    this.templateid="";//ihe specific id
    this.sectionDescription="This is my new section";
    this.code="";//Ihe specific code for section
}
}
```

Similarly one can edit or delete any Sections or CDA document type.

Populating sections of the APHP CDA Message

We have used OpenMRS Concept Dictionary to populate APHP CDA message section with answers. We used IHE APHP Document [value sets](#) which list out specific concepts that must be used for specific sections. We are updating the MVP-CIEL Concept dictionary with LOINC codes, SNOMED codes and new concept that our module needs.

Installation of the Module

[Installation of the OpenMRS CDA Generator Module](#)

Potential Improvements

- We can Add/Edit/Delete CDA Documents and Sections Programmatically but not through the UI.
- Create clients to consume CDAs by hitting a REST Web Service with a patient identifier and cda document type to download their CDA instead of users downloading through administrative tab.
- We are using gazelle as an external validator in addition to mdht cda validation code in our module. I'm manually using GUI gazelle validator to check our cda documents. So, we can call gazelle validator via a web service from our module.

Screenshots

1) Administration tab (After the module is loaded)

The screenshot shows the OpenMRS Administration tab interface. At the top, there is a navigation bar with the OpenMRS logo on the left and the text "Currently logged in as Super User | [Log out](#) | [My Profile](#) | [Help](#)" on the right. Below the navigation bar, the "Administration" tab is selected, and the page content is organized into several columns of links:

- Users**
 - [Manage Users](#)
 - [Manage Roles](#)
 - [Manage Privileges](#)
 - [Manage Alerts](#)
- Patients**
 - [Manage Patients](#)
 - [Manage Tribes](#)
 - [Find Patients to Merge](#)
 - [Manage Identifier Types](#)
- Person**
 - [Manage Persons](#)
 - [Manage Relationship Types](#)
 - [Manage Person Attribute Types](#)
- Visits**
 - [Manage Visit Types](#)
 - [Manage Visit Attribute Types](#)
 - [Configure Visits](#)
- Encounters**
 - [Manage Encounters](#)
 - [Manage Encounter Types](#)
 - [Manage Encounter Roles](#)
- Providers**
- Concepts**
 - [View Concept Dictionary](#)
 - [Manage Concept Drugs](#)
 - [Manage Proposed Concepts](#)
 - [Update Concept Index](#)
 - [Manage Concept Classes](#)
 - [Manage Concept Datatypes](#)
 - [Manage Concept Sources](#)
 - [Manage Concept Stop Word](#)
 - [Manage Reference Terms](#)
- Forms**
 - [Manage Forms](#)
 - [Manage Fields](#)
 - [Manage Field Types](#)
 - [Merge Duplicate Fields](#)
- HL7 Messages**
 - [Manage HL7 Sources](#)
 - [Manage Queued Messages](#)
 - [Manage Held Messages](#)
 - [Manage HL7 Errors](#)
 - [Manage HL7 Archives](#)
 - [Migrate HL7 Archives](#)
- Maintenance**
 - [Set Implementation Id](#)
- Modules**
 - [Manage Modules](#)
 - [Module Properties](#)
- Logic Module**
 - [Token Registration](#)
 - [Rule Definitions](#)
 - [Test Logic Expressions](#)
 - [Initial Set-Up](#)
- CDA Generator Module**
 - [Export CDA Document](#)
 - [Manage CDA Sections](#)
 - [Manage CDA Types](#)

2) Export CDA Document Link

OpenMRS Currently logged in as Super User | [Log out](#) | [My Profile](#) | [Help](#)

[Home](#) | [Find/Create Patient](#) | [Dictionary](#) | [Administration](#)

[Admin](#) | [Export CDA Document](#) | [Manage CDA Sections](#) | [Manage CDA Types](#)

CDA Generator Module

Generate CDA

Export CDA Document

Patient Name

CDA Document Type

3) Validation of Export CDA Form

OpenMRS Currently logged in as Super User | [Log out](#) | [My Profile](#) | [Help](#)

[Home](#) | [Find/Create Patient](#) | [Dictionary](#) | [Administration](#)

[Admin](#) | [Export CDA Document](#) | [Manage CDA Sections](#) | [Manage CDA Types](#)

CDA Generator Module

Generate CDA

Export CDA Document

Patient Name error, please select a patient

CDA Document Type

OpenMRS Currently logged in as Super User | [Log out](#) | [My Profile](#) | [Help](#)

[Home](#) | [Find/Create Patient](#) | [Dictionary](#) | [Administration](#)

[Admin](#) | [Export CDA Document](#) | [Manage CDA Sections](#) | [Manage CDA Types](#)

CDA Generator Module

Generate CDA

Export CDA Document

Patient Name

CDA Document Type error, please select a cda document type

4) Successful Generation of CDA Document

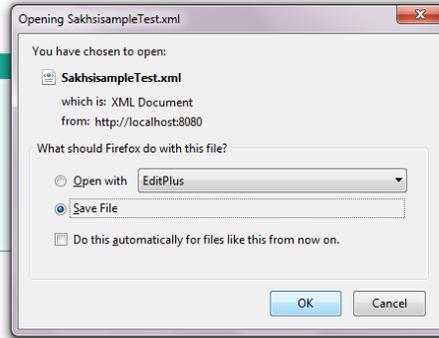
CDA Generator Module

Generate CDA

Export CDA Document

Patient Name

CDA Document Type



5) Error report (If generated document is invalid)

CDA Generator Module

CDA Document Error Page

Generated CDA Document is Invalid, please resolve errors reported in the table below

Errors in CDA Document

| Severity | Message |
|----------|---|
| ERROR | Error: The required feature 'code' of 'org.openhealthtools.mdht.uml.cda.impl.ObservationImpl@18380f3{#///@component/@structuredBody/@component.2/@section/@entry.0/@observation}' must be set |
| ERROR | Error: The required feature 'code' of 'org.openhealthtools.mdht.uml.cda.impl.ObservationImpl@1490d54{#///@component/@structuredBody/@component.2/@section/@entry.1/@observation}' must be set |
| ERROR | Error: The required feature 'code' of 'org.openhealthtools.mdht.uml.cda.impl.ObservationImpl@13ad9a6{#///@component/@structuredBody/@component.2/@section/@entry.2/@observation}' must be set |
| ERROR | Error: The required feature 'code' of 'org.openhealthtools.mdht.uml.cda.impl.ObservationImpl@b01b9d{#///@component/@structuredBody/@component.2/@section/@entry.3/@observation}' must be set |

6) Manage CDA Section Link

It lists out all sections present in our module and also provide details of each section

CDA Generator Module

Manage CDA Sections

Existing CDA Section

| Display Name | Template Id | Code | CodeSystem | CodeSystemName |
|---|------------------------------------|---------|-----------------------|----------------|
| ABDOMEN | 1.3.6.1.4.1.19376.1.5.3.1.1.9.31 | 10191-5 | 2.16.840.1.113883.6.1 | LOINC |
| BREASTS | 1.3.6.1.4.1.19376.1.5.3.1.1.9.28 | 10193-1 | 2.16.840.1.113883.6.1 | LOINC |
| CHEST WALL | 1.3.6.1.4.1.19376.1.5.3.1.1.9.27 | 11392-8 | 2.16.840.1.113883.6.1 | LOINC |
| Chief Complaint | 1.3.6.1.4.1.19376.1.5.3.1.1.13.2.1 | 10154-3 | 2.16.840.1.113883.6.1 | LOINC |
| HISTORY OF FAMILY MEMBER DISEASES | 1.3.6.1.4.1.19376.1.5.3.1.3.15 | 10157-6 | 2.16.840.1.113883.6.1 | LOINC |
| Coded Vital Signs | 1.3.6.1.4.1.19376.1.5.3.1.1.5.3.2 | 8716-3 | 2.16.840.1.113883.6.1 | LOINC |
| Ears, Nose, Mouth and Throat | 1.3.6.1.4.1.19376.1.5.3.1.1.9.20 | 11393-6 | 2.16.840.1.113883.6.1 | LOINC |
| EAR | 1.3.6.1.4.1.19376.1.5.3.1.1.9.21 | 10195-6 | 2.16.840.1.113883.6.1 | LOINC |
| ENDOCRINE SYSTEM | 1.3.6.1.4.1.19376.1.5.3.1.1.9.25 | 29307-6 | 2.16.840.1.113883.6.1 | LOINC |
| HISTORY OF FAMILY MEMBER DISEASES | 1.3.6.1.4.1.19376.1.5.3.1.3.14 | 10157-6 | 2.16.840.1.113883.6.1 | LOINC |
| General Appearance Section | 1.3.6.1.4.1.19376.1.5.3.1.1.9.16 | 10210-3 | 2.16.840.1.113883.6.1 | LOINC |
| GENITALIA | 1.3.6.1.4.1.19376.1.5.3.1.1.9.36 | 11400-9 | 2.16.840.1.113883.6.1 | LOINC |

CDA Generator Module

Edit CDA Section

Manage CDA Section

| | |
|---------------------|---|
| Display Name | <input type="text" value="Chief Complaint"/> |
| Template Id | <input type="text" value="1.3.6.1.4.1.19376.1.5.3.1.1.13.2.1"/> |
| Parent TemplateId | <input type="text"/> |
| Code | <input type="text" value="10154-3"/> |
| CodeSystem | <input type="text" value="2.16.840.1.113883.6.1"/> |
| CodeSystemName | <input type="text" value="LOINC"/> |
| Section Description | <input type="text" value="This contains a narrative description of the patient's chief complaint"/> |

7) Manage CDA Type Link

It lists out all CDA Document types present in our module and also provide details of each document type

CDA Generator Module

Manage CDA Types

Existing CDA Types

| Document Full Name | Document Short Name | Format Code | Document Description | Template Id |
|------------------------------------|---------------------|-----------------------|--|------------------------------------|
| Antepartum History and Physical | APHP | urn:ihe:pcc:aphp:2008 | Contains a record of initial history and physical | 1.3.6.1.4.1.19376.1.5.3.1.1.16.1.1 |
| Antepartum Summary | APS | urn:ihe:pcc:aps:2007 | information regarding the status of a patients pregnancy | 1.3.6.1.4.1.19376.1.5.3.1.1.11.2 |

CDA Generator Module

Edit CDA Type

Manage CDA Type

| | |
|----------------------|--|
| Document Full Name | <input type="text" value="Antepartum History and Physical"/> |
| Document Short Name | <input type="text" value="APHP"/> |
| Document Description | <input type="text" value="Contains a record of initial history and physical"/> |
| Format Code | <input type="text" value="urn:ihe:pcc:aphp:2008"/> |
| Template Id | <input type="text" value="1.3.6.1.4.1.19376.1.5.3.1.1.16.1.1"/> |
| Parent Template Id | <input type="text" value="1.3.6.1.4.1.19376.1.5.3.1.1.16.1.4"/> |

Quick Demonstration

Please watch demonstration of module :

Sample CDA Documents for review

For a sample document generated using the OpenMRS CDA generator module, see [here](#)

Downloads

Source code: <https://github.com/openmrs/openmrs-module-cdgenerator>