

# Simple SHR REST interface module

## Overview

The simple SHR REST interface module that mimics the API expected by the content handler module in a RESTful way. This allows for an easy way to test the system without building a complex interface module.

The module is available at: <https://github.com/jembi/openmrs-module-shr-rest>.

## Requests

It supports POST and GET requests on the following endpoints:

**POST: ws/rest/v1/shr/document**

Parameters:

<b>patientId</b>	The patient identifier
<b>patientIdType</b>	The type of identifier
<b>providerId</b>	The provider identifier
<b>providerIdType</b>	The type of identifier
<b>encounterType</b>	A name describing the type of encounter
<b>formatCode</b>	The format code of the submission
<b>isURL</b>	<b>(optional)</b> isURL=true indicates that the submitted payload is a URL reference

Submit a document for a particular patient.

Examples:

### POST XML Example

```
curl --user user:password "host:8080/openmrs/ws/rest/v1/shr/document?patientId=1&patientIdType=ECID&providerId=1&providerIdType=EPID&encounterType=Test&formatCode=ANC_Summary" -H "Content-Type: text/xml" -X POST -d "<data>...</data>"
```

### POST Image Example

```
curl --user user:password "host:8080/openmrs/ws/rest/v1/shr/document?patientId=1&patientIdType=ECID&providerId=1&providerIdType=EPID&encounterType=TestImage&formatCode=Image" -H "Content-Type: image/jpg" -X POST --data-binary @image.jpg
```

### POST URL Example

```
curl --user user:password "host:8080/openmrs/ws/rest/v1/shr/document?patientId=1&patientIdType=ECID&providerId=1&providerIdType=EPID&encounterType=TestImage&formatCode=Image&isURL=true" -H "Content-Type: image/jpg" -X POST -d "http://imageserver/image.jpg"
```

**GET: ws/rest/v1/shr/document**

Parameters:

<b>contentType</b>	The type of content to retrieve
<b>encounterUUID</b>	The UUID for the specific encounter that needs to be retrieved
<b>uniqueID</b>	<b>(NOT YET IMPLEMENTED)</b> The unique document ID of the encounter that needs to be retrieved

Retrieve a specific encounter. Either encounterUUID must be specified or uniqueID, but not both or none.

GET: ws/rest/v1/shr/documents

contentType	The type of contents to retrieve
patientId	The patient identifier
patientIdType	The type of identifier
dateStart	(optional) Format yyyy-MM-dd'T'HH:mm:ss
dateEnd	(optional) Format yyyy-MM-dd'T'HH:mm:ss

Query for patient documents of a particular content type. Note that the response content type (determined by the **Accept** header) is not the same as the **contentType** of the document and response can either be in XML or JSON.

Examples:

GET XML Example

```
curl --user user:password "host:8080/openmrs/ws/rest/v1/shr/documents?contentType=text%2Fxml&patientId=1&patientIdType=ECID" -H "Accept: text/xml"
```

Response:

```
<linked-list>
  <org.openmrs.module.shr.contenthandler.api.Content>
    <formatCode>ANC_Summary</formatCode>
    <contentType>text/xml</contentType>
    <encoding>UTF-8</encoding>
    <representation>TXT</representation>
    <payloadIsUrl>false</payloadIsUrl>
    <payload>&lt;data&gt;...&lt;/data&gt;</payload>
  </org.openmrs.module.shr.contenthandler.api.Content>
</linked-list>
```

GET Image Example

```
curl --user user:password "host:8080/openmrs/ws/rest/v1/shr/documents?contentType=image%2Fjpg&patientId=1&patientIdType=ECID" -H "Accept: text/xml"
```

Response:

```
<linked-list>
  <org.openmrs.module.shr.contenthandler.api.Content>
    <formatCode>Image</formatCode>
    <contentType>image/jpg</contentType>
    <encoding>Binary</encoding>
    <representation>B64</representation>
    <payloadIsUrl>false</payloadIsUrl>
    <payload>BASE64 Content...</payload>
  </org.openmrs.module.shr.contenthandler.api.Content>
  <org.openmrs.module.shr.contenthandler.api.Content>
    <formatCode>Image</formatCode>
    <contentType>image/jpg</contentType>
    <encoding>Binary</encoding>
    <representation>BINARY</representation>
    <payloadIsUrl>true</payloadIsUrl>
    <payload>http://imageserver/image.jpg</payload>
  </org.openmrs.module.shr.contenthandler.api.Content>
</linked-list>
```