

Barcelona, 18<sup>th</sup> May 2016

## **DOSSIER APP. BASIC UNDERSANDING**

### **MIGRATION & MAINTENANCE**

## Table of Contents

Introduction .....	3
Server configuration. Structural elements .....	3
Dataelements for the dossiers .....	3
Attribute serviceCode .....	3
Dataset ZZDossier.....	4
Organisation units to introduce dossier definition .....	5
sqlView Dossiers.....	5
Dossier app.....	5
Export of dossier app into a new server .....	5
Creating a new dossier .....	6
User manual .....	7
Potential Troubles .....	9
Dhis2 Menu .....	9
Session ends .....	9
Bad connection.....	9
Adding a language .....	9
First javascript file controller.js.....	9
Add language file to languages folder.....	10
See Dataelements for the dossiers .....	10
Add OU for new language .....	10

## Introduction

Dossiers are an essential part of the HMIS. They are the main reference in terms of health information for both dataelements and indicators. And they help users to know the exact definition of what has to be collected and how the predefined indicators were calculated. An App was developed (thank you Rishi!) to facilitate the task of dossier maintenance and delivery to HMIS users, both in online and offline servers. This document describes :

1. Elements required to configure an HMIS server to be used with the Dossier App.
2. Exporting Dossier App structure to another server.
3. How to create a new dossier or update an existing one.
4. User manual.

## Server configuration. Structural elements

The following sections describe the elements that need to be incorporated in an HMIS server to be able to use the dossier App.

### *Dataelements for the dossiers*

They represent a new dossier. They contain the description of the dossier and a code used to decide the health information that will be included (dataelements and indicators). We need to create one per dossier. They have to be configured taking into account the following considerations :

1. **Name/Shortname:** internal name of the data element. In instance *ZZDossier\_ER* for Emergency Room.
2. **Code:** key to match datasets or indicator groups with dossiers. In instance *DS\_ER* for Emergency Room. **REQUIRED!**
3. **Description:** name that will be shown as title of the dossier displayed to the final user. **REQUIRED!**
4. **Form name:** name that will be shown in the data entry section (during the configuration).
5. **Value type:** Long text. **REQUIRED!**

The field Description has to be translated into the different languages considered, since it is included in the dossier displayed to the final user.

### *Attribute serviceCode*

Required for the use of the App. Created in Apps->Data administration->Attribute as shown below:

Details	
Name *	<input type="text" value="serviceCode"/>
Code	<input type="text"/>
Mandatory	<input type="checkbox"/>
Assigned to	<input type="checkbox"/> Data element <input type="checkbox"/> Data element Group <input type="checkbox"/> Indicator <input checked="" type="checkbox"/> Indicator Group <input checked="" type="checkbox"/> Data Set <input type="checkbox"/> Organisation Unit <input type="checkbox"/> Organisation Unit Group <input type="checkbox"/> Organisation Unit Group Set <input type="checkbox"/> User <input type="checkbox"/> User Group <input type="checkbox"/> Program <input type="checkbox"/> Program Stage <input type="checkbox"/> Tracked Entity <input type="checkbox"/> Tracked Entity Attribute <input type="checkbox"/> Category Option <input type="checkbox"/> Category Option Group
Value Type	<input type="text" value="Text"/> ▼ <input type="text" value="[Please select]"/> ▼

It defines a new text field to be configured in the datasets and indicator groups. Introducing the code of one of the *dossier dataelements*, all the dataelements within the dataset or the indicators within the indicator group will be included in the dossier.

### ***Dataset ZZDossier***

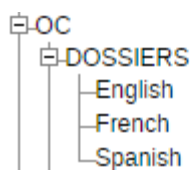
Required during the configuration of the App. It contains basically all *dossier dataelements*, allowing the user to enter the description for each dossier. It was created taking into account the following considerations:

1. **Name/Short name:** ZZDossier
2. **Code:** ZD. **REQUIRED!**
3. **Frequency:** Yearly. **REQUIRED!**
4. **Selected data elements:** all dossier dataelements.

Data has to be collected in 2015. It is there where the App is going to look for the descriptions.

## Organisation units to introduce dossier definition

The following structure is required in the HMIS hierarchy to allow the configuration of the dossiers:



The dataset ZZDossier is assigned to the three OUs at service level: English, French and Spanish. In this way, admin users can enter the description of the dossier for each language. If a new language is required, we would need to add another OU to this hierarchy, with the name of the language (**TBD: Rishi, would the current code recognize this additional language?**).

## sqlView Dossiers

Used internally by the Dossier app. Get dossier based on parameter language code and Service code (see image below).

Name \*

Description

Cache strategy

SQL type

SQL Statement \*

## Dossier app

Finally, the Dossier app has to be installed in the system, in instance through Apps->App Management. The code is contained in the package hmis\_dossier\_app.zip.

## Export of dossier app into a new server

Export the following items:

1. Organization Units:
  - a. DOSSIERS

- b. English
  - c. French
  - d. Spanish
2. Datasets: ZZDossier mandatory, others optional.

**Note:** datasets are exported with their OU assignments. If we decide to export all datasets both servers have to be synchronized previously.

3. Indicator groups. Optional.

**Note:** If we decide to export indicator groups both servers have to be synchronized previously.

4. Attribute: serviceCode
5. sqlView: Dossier.
6. Dataelements: all dossier dataelements.
7. Data: 2015 data from English, French and Spanish OUs.

If datasets and indicator groups were not exported, assign codes to the datasets and indicator groups.

## Creating a new dossier

1. Create a new dataelement as explained in section *Dataelements for the dossiers*.
2. Edit the dataset ZZDossier and add the dataelement created above.
3. Edit the datasets and indicator groups to be included in the dossier and assign the code of the dataelement created in point 1 to the attribute serviceCode (**TBC: note that one dataset/indicator group cannot be assigned to two dossiers**).
4. In the dataEntry section, select the OUs English, French and Spanish in the hierarchy, the dataset ZZDossier and the period 2015. Enter the description of the new dossier in the field available. The description can be written in html code. In order to facilitate this task:
  - a. Open one of the dossiers in Microsoft Word (or create the text in a new Word document).
  - b. Remove the title and all tables, keeping just the description that has to be inserted.
  - c. Save as 'Web page, filtered'.

- d. Open the .htm document in notepad or other text editor, copy all the text and paste it in the appropriate dataelement (see image below).

## Saisie de données ?

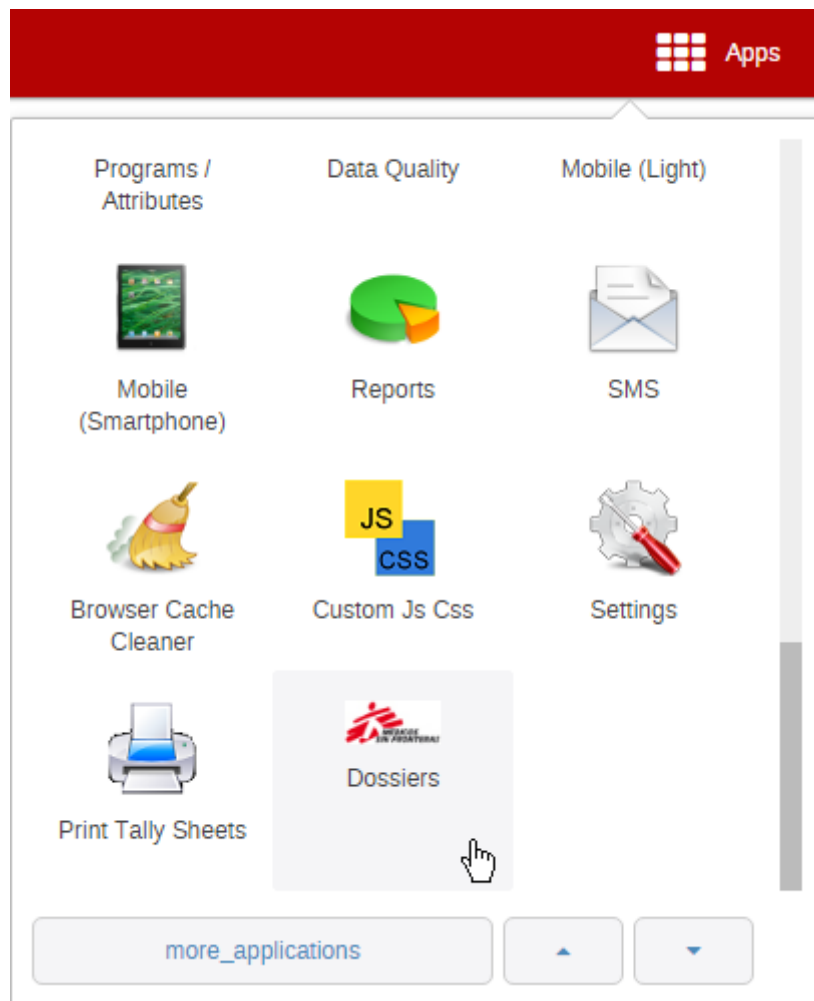
Unite d'organisation	English		
Ensemble de donnees	ZZDossier ▼		
Periode	2015 ▼	Annee prec	Annee suiv.

ZZDossier		
Filter in section	Value	
Emergency Room	<p class=MsoNormal style='margin-bottom:6.0pt;text-align:justify'><span lang=EN-US style='font-size:10.0pt;line-	0
External Consultations	<div class=WordSection1> <p class=MsoNormal align=center style='text-	0
Hospitalization Ward	<p class=MsoNormal style='margin-bottom:6.0pt;text-align:justify'><span lang=EN-US style='font-size:10.0pt;line-	0
Nutritional Program	<div style='border:solid windowtext 1.5pt;padding:0cm 4.0pt 1.0pt 4.0pt'>	0
Observation Room	FGADF	0

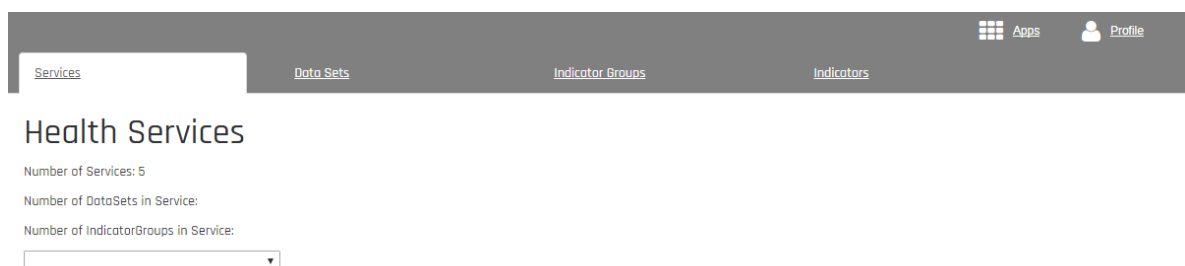
Congratulations. The Dossier app was just configured with another dossier!

## User manual

Press Apps->Dossiers, as shown below.



Four tabs will be displayed:



1. **Services:** where the dossiers will be compiled. Select the service you are interested in by means of the drop-down list. The dossier will be displayed, including:
  - a. Introduction of the dossier.
  - b. Table of content.
  - c. Description of dataelements assigned to the service.
  - d. Description of indicators assigned to the service.



**Note:** the dossier is shown in the same language as it is configured in the HMIS account (Profile->Settings).

2. **Data Sets:** displays all data elements (health information) within a dataset. The dataset is selected by means of a drop-down list.
3. **Indicator Groups:** displays all indicators within an indicator group. The indicator group is selected by means of a drop-down list.
4. **Indicators:** displays all indicators in the system.

## Potential Troubles

### *Dhis2 Menu*

In the developer manual of dhis2 it is described how to implement the OOB menu of dhis2. It requires a bit of javascript to be added to index.html file . The code I copied from the manual. It has one important drawback, the tomcat app name is hard coded, So I assume that when the app name is not dhis, you have to change the following line as well.

```
dhis2.settings.baseUrl = 'dhis';
```

### *Session ends*

There is a build in feature in the app the will direct you to the login page when your session is ended because of a timeout. This however doesn't always works, especially when there is a server reset this feature doesn't work, the user will then get a "weird" result (json result or then "pong" message).

### *Bad connection*

The dossier are dynamic assembled using multiple AJAX calls when there is in the middle a bad interruption the app will still come back with a result but some AJAX call might have failed to return a result this can result in some tables with 0 elements. It only happened once to me, so it could have been freak incident.

## Adding a language

### *First javascript file controller.js*

Check if your language is in the following in controller.js:

```
$translateProvider.registerAvailableLanguageKeys(  
    ['es', 'fr', 'en', 'pt'],
```

```

{
    'en*': 'en',
    'es*': 'es',
    'fr*': 'fr',
    'pt*': 'pt',
    '*': 'en' // must be last!
}
);

```

### ***Add language file to languages folder***

Go to the app subfolder languages and add the language file, check out fr.json as example.

If you see something like “{{'Health Services' | translate}}” in one of the html files you have to look for “Health Services” in fr.json to find the translation at the right side.

For a new language you better copy fr.json rename the copy to something like nl.json (Dutch ☺ ) and replace the right side translation with translations for the new language.

Add translation to zzelement

### ***See Dataelements for the dossiers***

See the paragraph “Dataelements for the dossiers” for adding translation for the new element description, this will as well be the translated name for the service.

### ***Add OU for new language***

We'll see the picture below, for the new language a new OU has to be created and the dataset ZZDossier has to be assigned to the new ou.

