Alerta Joven Architecture and Developer Guide

This guide is intended to document the overall architecture of the Alerta Joven system.

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **When** | **Changes** | **Who** |
| 1.0 | 12/8/2014 | Initial creation of this doc merging in the metadata architecture guide into this doc  and adding in additional content. | Randy Eckhoff |
| 1.1 | 1/29/2015 | Added the Utilities , ELT Cron Modules and Reporting sections | Vesselina Bakalov |
|  |  |  |  |

# Technology Stack

1. Drupal ([www.drupal.org](http://www.drupal.org/)). Used as the content management system driving the overall website, it uses a MySQL database server. Critical Drupal modules used:
   1. Coconut – to integrate coconut into Drupal
   2. Entity Construction Kit – to manage the metadata
   3. Forena Reports – for creating the reports
   4. Form Tagger – to customize various aspects of Drupal, especially in the admin area for defining the metadata.
   5. Hierarchical Select – select a hierarchy of items as 1 option
   6. PDF using mPDF – HTML to PDF conversion
   7. Views – Create customized queries from the database
2. Coconut ([www.ictedge.org/coconut](http://www.ictedge.org/coconut)). Used as the data collection system for collecting Alerta Joven Participant and Survey information.
   1. CouchDB ([couchdb.apache.org](http://couchdb.apache.org/)), a document based database, for the form generation and data storage.
   2. backbone.js ([backbonejs.org](http://backbonejs.org))
   3. CoffeeScript ([coffeescript.org](http://www.coffeescript.org))
   4. JSON ([www.json.org](http://www.json.org/))
3. Amazon Web Services ([aws.amazon.com/](http://aws.amazon.com/)) – for hosting the system in the cloud
4. Apache HTTP Server ([httpd.apache.org/](http://httpd.apache.org/)) – the HTTP server
5. Bitnami ([bitnami.com](https://bitnami.com/)) – The application store from which the drupal bundle was installed.
6. Linux – The operating system running the stack

# Useful Links

1. See above for technology stack
2. CoffeeScript Cookbook - [coffeescriptcookbook.com](http://coffeescriptcookbook.com/)
3. CouchDB Cookbook for SQL Jockeys - [guide.couchdb.org/draft/cookbook.html](http://guide.couchdb.org/draft/cookbook.html)

# System

You don’t see Coconut called out in the below diagram as its delivery vehicle is CouchDB.



All webpages start with the Drupal Content Management System. Participant data entry includes an iFrame for the coconut form that comes from CouchDB. All participant data is stored in CouchDB and an Extract, Transform, and Load (ETL) modules integrate the data from CouchDB into MySQL for reporting. These modules are schedule to run nightly as Druap cron jobs. All metadata about the website is administered via Drupal and is described later in this document.

For development purposes, there is a development version of this system to try out new ideas and concepts before implementing them on production.

# Utilities

As in any data entry system due to human or machine errors sometimes wrong data are entered. This creates data integrity and inconsistency issues. To correct and clean the wrong data utilities programs have been implemented. It is important to note that **nothing is deleted** from the system. The data are manipulated by updating some of the attributes. Some of these programs are run as standalone PHP programs and some of them are part of coconut, e.g. run from CouchDB. The following programs have been implemented:

|  |  |  |
| --- | --- | --- |
| **Description** | **Location of the File** | **How to Run** |
| The program marks incomplete duplicate registrations as DUPLICATE. | app\\_attachments\  cleanDuplicateIncompleteSurveys.html | This is hosted inside CouchDB. Run from the browser by referring to the following URL:  http:// 107.20.181.244:5984/coconut/\_design/coconut/ cleanDuplicateIncompleteSurveys.html |
| The program checks every completed registration and when duplicate is found it marks the registration DUPLICATE so it does not show for the end user. | app\\_attachments\  cleanDuplicateCompleteRegs.html | This is hosted inside CouchDB. Run from the browser by referring to the following URL:  http:// 107.20.181.244:5984/coconut/\_design/coconut/cleanDuplicateComplete.html  **NOTE:** First run cleanDuplicateIncomplete.html described above. |
| Marks duplicate surveys as duplicate when more recent participant survey is found. The marked as duplicate surveys will not be visible to the user. | app\\_attachments\  cleanDuplicateSurveys.html | This is hosted inside CouchDB. Run from the browser by referring to the following URL:  http:// 107.20.181.244:5984/coconut/\_design/coconut/ cleanDuplicateSurveys.html |
| Converts incomplete registrations to complete when validation criteria are satisfied. It generates output file with registrations that were not processed because they did not pass the required fields validation. | app\\_attachments\  processIncompleteParicipants.html | This is hosted inside CouchDB. Run from the browser by referring to the following URL:  http:// 107.20.181.244:5984/coconut/\_design/coconut/ processIncompleteParicipants.html |
| Converts incomplete surveys to complete. | app\\_attachments\  processIncompleteSurveys.html | This is hosted inside CouchDB. Run from the browser by referring to the following URL:  http:// 107.20.181.244:5984/coconut/\_design/coconut/ processIncompleteSurveys.html |
| Updates documents based on corrections from an input Excel spreadsheet. | utilities\parsecorrections\parseExcel.php | This is a PHP program and will need PHP environment to run. As an input it expects Excel spreadsheet where the corrections **must be highlighted in green** exactly as it’s done in the any of the Excel files under utilities\parsecorrections\inputdata. The program will look for this markup and generate output file with corrections that is processed by updateCorrections.html program listed below. |
| Iterates through corrections arrays and updates the documents. | app\\_attachments\  updateCorrections.html | This is hosted inside CouchDB. Run from the browser by referring to the following URL:  http:// 107.20.181.244:5984/coconut/\_design/coconut/ updateCorrections.html  **NOTE:** The corrections must be previously generated by running the parseExcel.php program, described in the above row and corrections must be copied to the updateCorrections. html program. The coconut application must be re-deployed to production for the changes to take effect. |
| Process incomplete registrations from an input CSV file. The program will update the document as specified in the Decision column from the input file. | utilities\aj-couchdb-php\ processIncomplete.php | This is a PHP program and will need PHP environment to run. As an input it expects Excel spreadsheet that needs to be in format as in the utilities\aj-couchdb-php\input\ missingRegFields\_10-23-2014.csv file. It will produce output file with status. Look at the utilities\aj-couchdb-php\output\missingRegFields\_10-23-2014\_PROCESSED.csv. |
| Creates participants from a CSV input file. As an output it will produce the same participants but with assigned UUIDS. | utilities\aj-couchdb-php\ createParticipants.php | This is a PHP program and will need PHP environment to run. As an input it expects Excel spreadsheet that needs to be in format as in the utilities\aj-couchdb-php\input\ testAttendanceDEV.csv file. It will produce output file with status. Look at the utilities\aj-couchdb-php\output\ testAttendanceDEV\_WITH\_UUIDS\_2.csv. |
| Assigns existing participants to existing activities. | utilities\aj-couchdb-php\ processAttendees.php | This is a PHP program and will need PHP environment to run. As an input it expects Excel spreadsheet that needs to me in format as in the utilities\aj-couchdb-php\input\ testAttendanceDEV.csv file. NOTE: The activities and participants need to exist in the system otherwise they will not be processed. |

# Drupal Metadata

The Alerta Joven website allows for youth to participate in activities. These activities are organized in a hierarchical structure:

1. Providers (Organización) – the top most tier.
2. Programs – Providers provide programs of various types like health and education. A program is sponsored by only one provider.
3. Activities – Within a program there can be many activities of various types. Activities are tied to only one program.

In addition to the above data elements, other metadata elements to support the website are:

* Program Names – used to help organize Programs.
* Activity Types – used to help organize Activities.
* Users – Who has access to the administrative features for editing Alerta Joven metadata.

Since there are different data elements that need management, a number of user roles have been created. This allows for control over what users can edit what data. Users without the below roles can only use the site; they cannot make changes to the site.

1. **Drupal System Administrator** – Has full access to the entire Drupal system
2. **Evaluator** – Can manage all the metadata that the website requires.



1. **Provider Administrator** – Can manage the metadata only for the provider they are assigned to for Programs, Activities and Users.



1. **Case Manager** – Can only manage activities for the provider they are assigned to, typically the attendance list.
2. **Data Entry** – This role was created to assist with the input of the existing paper registration forms. Can only manage activities for the provider they are assigned to, typically the attendance list.

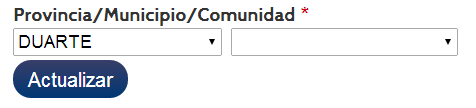


Most of the metadata fields should be self-explanatory (name, telephone, etc). One field that is not obvious is the **Active** field. Data should never be deleted for historical purposes. Instead, the active field exists so that a record can be marked as inactive if it is no longer needed.

## Managing Providers

The edit and create provider pages are identical with their content.

The **Area Served** field is interesting in that it lets you select a Provincia, Municipio, and a Barrio.



The first dropdown list is the Provincia. The 2nd dropdown list is for Municipio and its contents are dependent on the selection in the 1st dropdown list. Selecting an item in the “previous” dropdown list doesn’t force a refresh of the subsequent dropdown lists. After selecting the Provincia, click the Update button and the Municipio (2nd dropdown list) will be refreshed. Select your Municipio and click the Update button again. A 3rd dropdown list for Barrio is displayed. This control is rather odd in that sometimes, just selecting in the dropdown list is enough and sometimes it isn’t and you have to use the Update button.

## Managing Programs and Program Names

The edit and create program pages are identical. Program Names are utilized to provide organization to the programs. When you create a program, instead of giving it a freeform name, you select from a list of program names. You then tie that program to a specific provider.

The provider dropdown list displays all providers unless the person logged in is a Provider Administrator. Then, only their provider is displayed as they can only create programs for the provider they are tied to.

## Managing Activities and Activity Types

The edit and create activity pages are identical. Activity types are utilized to provide organization to the Activities. When you create an activity, you select the activity type from the list of predefined types instead of entering a freeform type.

The Program dropdown list displays all Programs if the logged in person is an Evaluator. Otherwise, only the programs associated to that logged in user is shown.

## Activity Attendance

To manage the list of participants who attended a particular activity, display all the activities and select “Listado de Participantes”.



The resulting page displays all the participants tied to the Activity’s provider. Select the checkbox next to each participant who attended the activity and save.



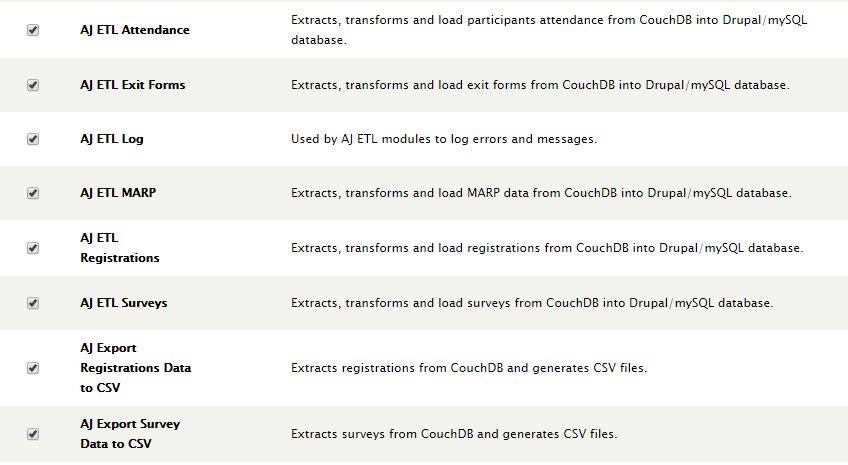
## Managing Users

This is not managing the youth participants but rather managing the users who can log in to the Alerta Joven System. Creating a user here is creating a user in the Drupal System. As described above in the overview section, you can assign a role to a user giving them permissions to do different things on the website.

Each user must be assigned to a provider.

# ETL Drupal Cron Modules

For the reporting component the data from CouchDB are integrated with provider/programs/activities metadata stored into Drupal MySQL database. In order to accomplish this the following Drupal modules have been implemented and run nightly as scheduled Drupal cron jobs. All modules were copied to apps/drupal/htdocs/sites/all/modules directory in the production server under AWS and enabled from Drupal. You can view the modules when login as administrator under Modules (see below):



The modules are run as Cron jobs. The Cron is configured to run nightly. To view the settings go to Configurations->Cron on the Drupal Web Site.

The modules description and locations is presented in the table below:

|  |  |  |
| --- | --- | --- |
| **Description** | **Local Location** | **Production Server Location** |
| Registrations loading module | module\aj-etl\ajetlregs | apps/drupal/htdocs/sites/all/modules/ajetlregs |
| Surveys loading module | module\aj-etl\ ajetlsurveys | apps/drupal/htdocs/sites/all/modules/ ajetlsurveys |
| Participants exit data loading module | module\aj-etl\ ajetlexit | apps/drupal/htdocs/sites/all/modules/ajetlexit |
| Activities attendees loading module | module\aj-etl\ ajetlattendance | apps/drupal/htdocs/sites/all/modules/ ajetlattendance |
| Most at Risk Participants (MARP) data loading module | module\aj-etl\ ajetlmarp | apps/drupal/htdocs/sites/all/modules/ ajetlmarp |
| Error and messages log module | module\aj-etl\ajetllog | apps/drupal/htdocs/sites/all/modules/ajetllog |

# Reporting

Reporting was implemented by deploying the Forena Drupal module. This is a very flexible and low cost solution. To read more about Forena reporting go [here](https://www.drupal.org/project/forena). To access the reports go to Database->Reports page. As an admin you can edit the reports from there or you can edit them from Structure->Forena Reports. The ConfigureForenaReportswithPDF.docx explains how export to PFD was configured. The location of modified Forena files and the queries is:

|  |  |  |
| --- | --- | --- |
| **Local Location** | **Server Location** | **Description** |
| forena\reports | apps/drupal/htdocs/sites/default/files/reports | Customized skin files. |
| forena\sql\alertajoven | apps/drupal/htdocs/sites/default/sql/alertajoven | The queries used to generate the reports. |