DQX Library documentation

# Introduction

DQX is a framework that assists in the creation of web app with a look and feel that is similar to a desktop application. It was originally created to build the MalariaGEN P. Falciparum population genetics web application, and has some focus on visualization tools for genomic data. It offers the following functionality:

* A layouting mechanism to organize visual components on a page, mimicking a desktop application interface (e.g. no single overall scroll bar).
* Emulation of a multi-page application using a single JavaScript environment, enabling the usage of the browser “back” button.
* Encapsulation of a set of commonly used GUI elements (trees, lists, various controls, etc…).
* A messaging system to assist communication between several application components.
* Blocking and non-blocking popups.
* A wizard creation framework.
* A framework to fetch database data from the server.
* A paged grid viewer that hooks up to a table on the server.
* An interactive query builder that can be used in conjunction with the table grid viewer.
* A graphical genome viewer, able to show a rich variety of channels.
* An abstraction layer for the Google Maps API.
* Various small utilities.

The framework consists in two major components: **DQX** and **DQXServer**. In addition, there also is a sample web application that utilizes most of the functionality in the DQX library in a simple way: **DQXTest**. Most of the functionality can be learned by looking at this sample application.

These components are stored in the following GitHub repositories:

<https://github.com/malariagen/DQXServer>

<https://github.com/malariagen/DQX>

<https://github.com/malariagen/DQXTest>

# Architecture



## Components

**Web app**

Contains the code specific to a particular application. It creates the web application, relying on the functionality provided in DQX.

**DQX**

Contains client-side JavaScript code of the framework, running in the client browser. Functionality in this JS code communicates with corresponding code in DQXServer through Ajax requests.

**Web Server**

Serves the static content and the JS code, and redirects the DQX Ajax requests to DQXServer through a wsgi interface.

**DQXServer**

Contains server-side Python code, handling requests from DQX. This layer communicates with one or more SQL databases, and a subset of the file system. Access parameters to the database and the file system are specified in [config.py](https://github.com/malariagen/DQXServer/blob/master/config.py).

**Database**

Currently, MySQL is supported (using MySQL-Python). Currently, the framework supposes that full access to that database is acceptable (i.e. only public data should be served). It is planned to hook DQXServer to a single sign-on authentication system.

**File system**

DQXServer has read/write access to a subset of the file system, located in a directory called BASEDIR.

# Deploying DQXTest

## Required software components:

* Apache2 web server
* Python 2.7
* Mod\_wsgi
* MySQL
* MySQL-Python

## Code installation

…

In the DQXTest file [MetaData.js](https://github.com/malariagen/DQXTest/blob/master/webapp/scripts/MetaData.js), the the line MetaData.serverUrl=… should

specify url for accessing the DQXServer component.

## Required data components:

Location of the data: WTCHG internal virtual machine ‘panoptes’ (129.67.45.41)

### Database

The content of the database ‘pfpopgen’ should be copied as a MySQL database with the same name on the deployment server. The location of this database, as well as the access credentials should be specified in the DQXServer file [config.py](https://github.com/malariagen/DQXServer/blob/master/config.py).

### Files

The content of the directory /mnt/storage/webapps/Tracks-PfPopGen2.1 should be copied to the **base directory** on the deployment server, under a subdirectory with name Tracks-PfPopGen2.1. The location of the

base directory should be specified in the DQXServer file [config.py](https://github.com/malariagen/DQXServer/blob/master/config.py).

# Major components of DQX

# Other utilities