



SPECIAL INSTRUCTIONS

For Team Floriders SMART ON FHIR Survey Web App

Abstract

The following documents is designed for the CDC, this guide would help you deploy the web app in your own server. GA TECH Instructors please review the “Manual” on how to run the web app since the web app is being hosted by a Team Florider AWS controlled web server.

Table of Contents

How to Deploy Web App	2
Requirements	2
How to Deploy Web App using Docker.....	2
How to add Questions and Response to PostgreSQL database.....	3
Background.....	3
How to use the Database SQL files	3

How to Deploy Web App

Requirements

One Linux Server:

- Duo Core Processor
- 4 GB of RAM
- 250 GB Hard Drive

Software:

Docker 1.1 or above

GIT 2.1 or above

Postgresql 9.5 or above

Network:

Port 8080, 5432 must be open on the server

How to Deploy Web App using Docker

1. Open a terminal
2. Run the following command to clone the git public repo
git clone <https://github.com/eloporras/CDC-Population-Health-Informatics-Framework.git>
3. Navigate to the following folder
cd CDC-Population-Health-Informatics-Framework\Final Project\Final Application\Application
4. Run the following command within this folder (including the ".")
docker build -t CDC_Survey_App .
5. Run the following command to run the Docker Container
docker run -t -i -p 8080:8080 CDC_Survey_App /bin/bash
6. Within the shell prompt of the docker container run the following command
service tomcat8 start
7. The app should be running within the following URL
<http://{your-host-name}:8080/fhir-app/>

How to add Questions and Response to PostgreSQL database

Background

There are five tables in our database: question, question answer, subquestion, section and survey.

The “**survey**” table contains three surveys and the “**section**” table contains three sections for our application.

The “**question**” and “**subquestion**” tables list all the questions for the survey and the “**question_answer**” table contains all the answers for these questions

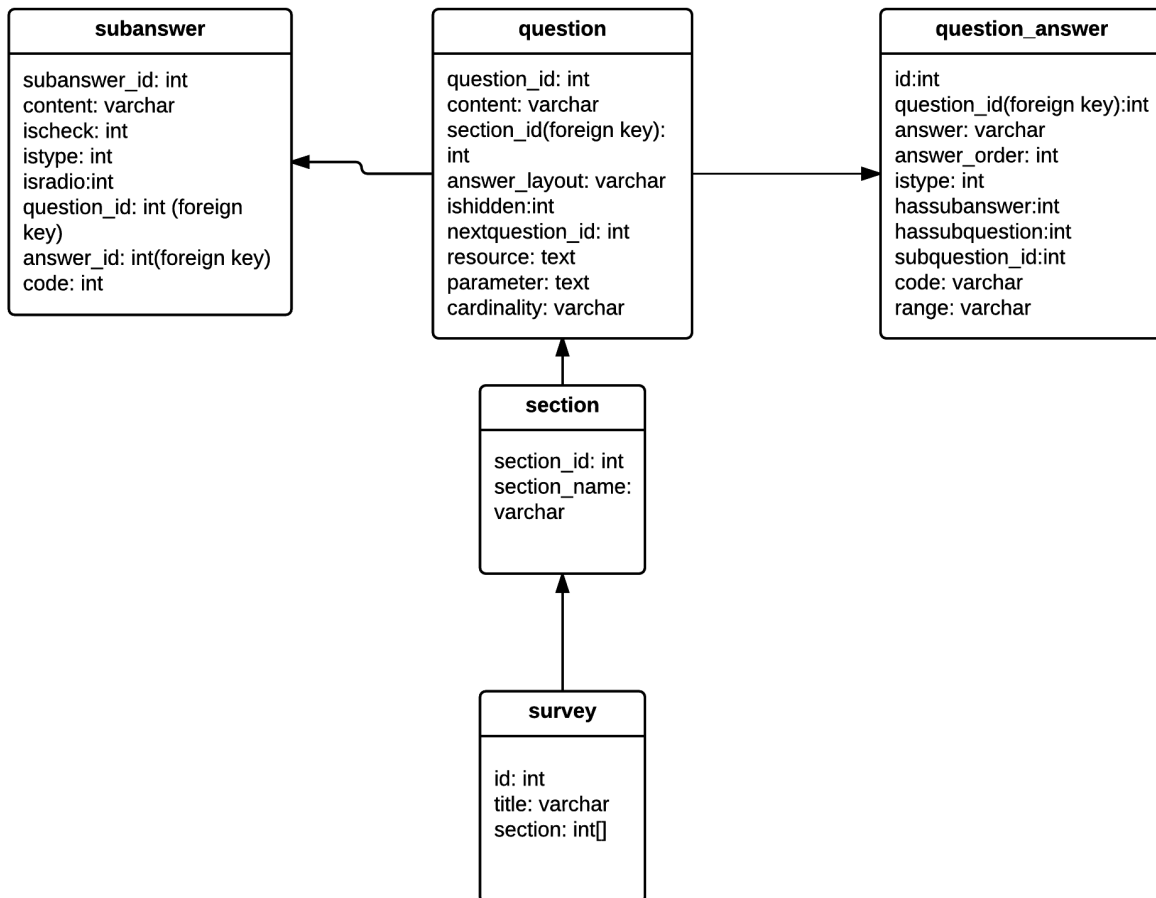
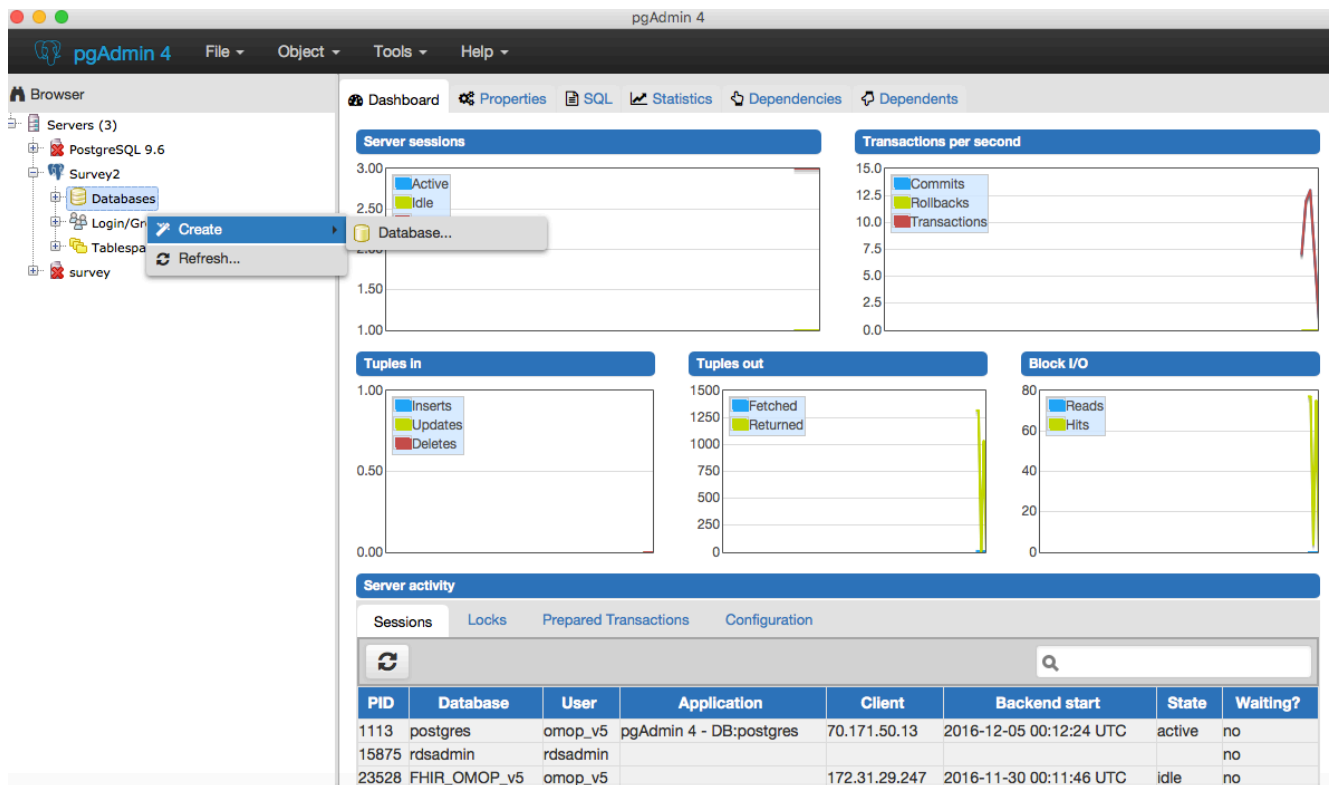


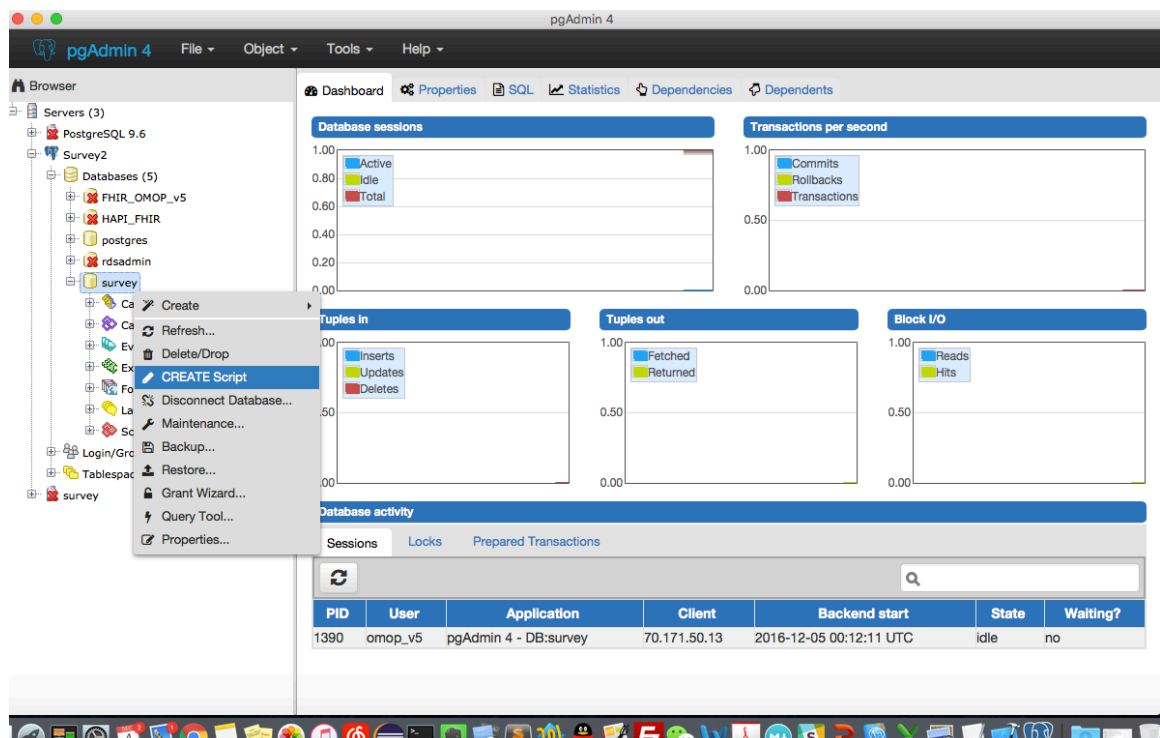
Fig Database schema

How to use the Database SQL files

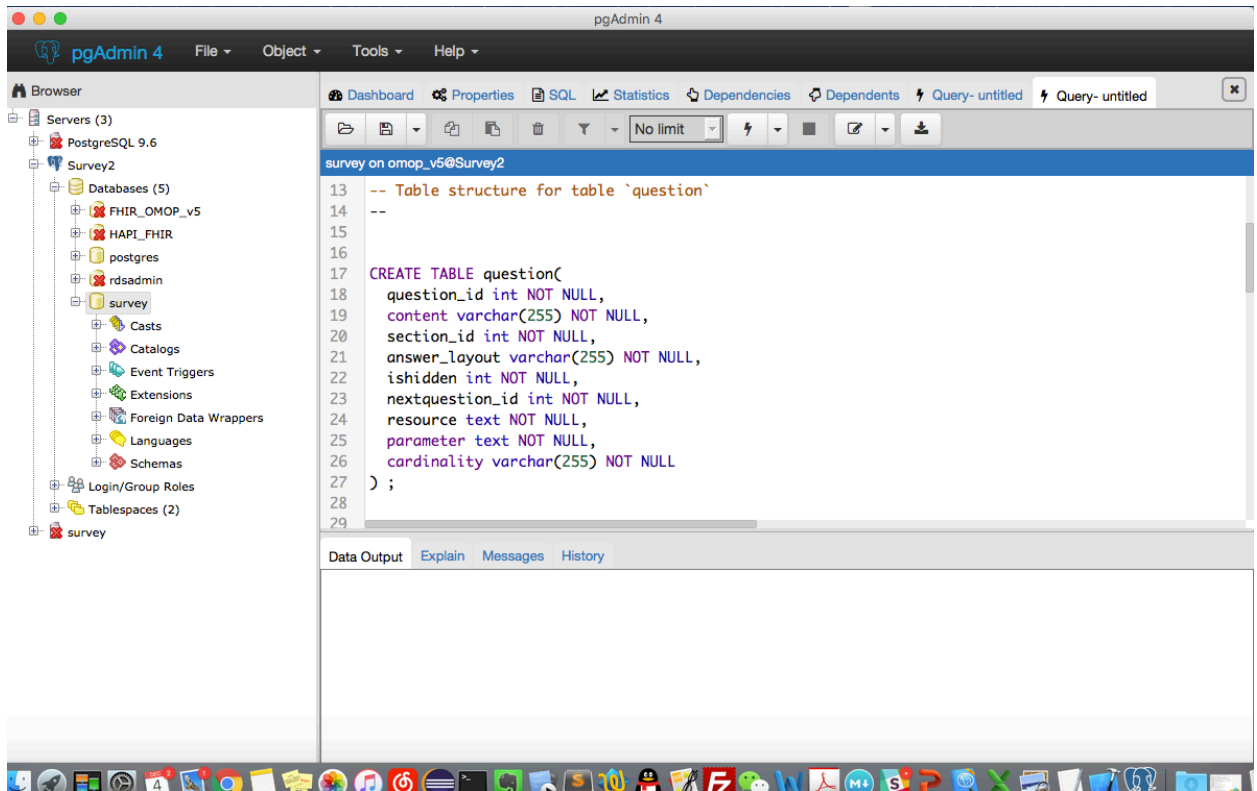
1. From the pgAdmin initial screen, create a new database called Survey by right clicking on the word Database, then selecting ****Create****.



2. Then you can click ****survey****, then right click on the word ****Create Scripts**** at next level menu.

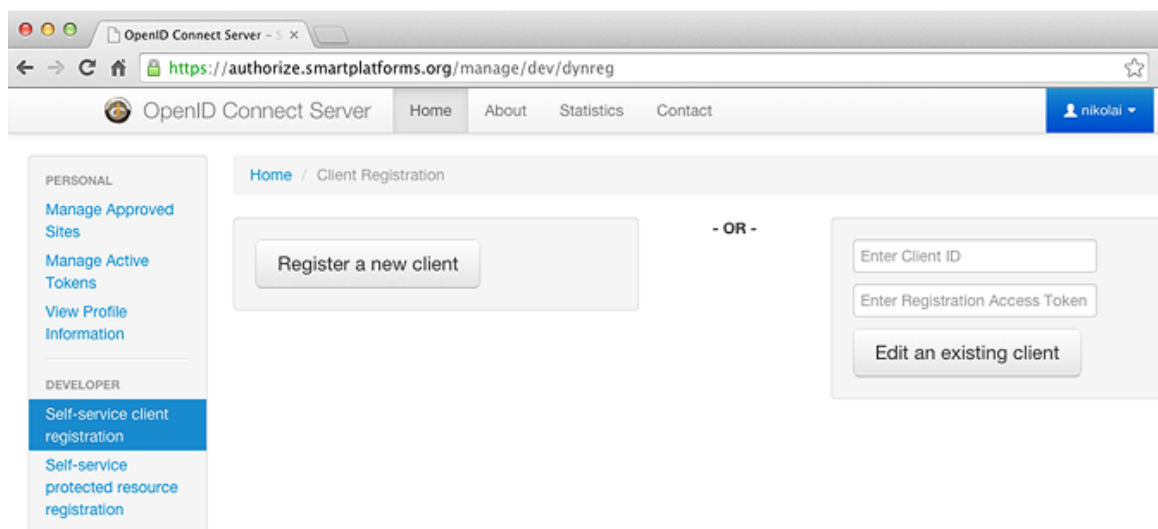


- Run the scripts which is located at (**\\CDC-Population-Health-Informatics-Framework\\Final Project\\Final Application\\Application\\Database**). It will load all the required questions in your survey database.



How to register a Smart on FHIR app

- Log into the OpenID Connect Server [file://localhost/at https://authorize-dstu2.smarthealthit.org](https://localhost/at https://authorize-dstu2.smarthealthit.org) and click on Register a new client in Self-service client registration.



2. In the Main tab enter a user-friendly name for your app in the Client name box. Also, enter the main URI(`http://{your-host-name}:8080/fhir-app/`) of your app which the user client should visit after completing the app authorization process with the OpenID Connect Server. Finally, enter the URL of the app's logo that will be displayed to the user during the authorization process.

The screenshot shows the 'New Client' registration page in the OpenID Connect Server interface. The browser address bar shows `https://authorize.smartplatforms.org/manage/dev/dynreg/new`. The page has a sidebar with 'PERSONAL' and 'DEVELOPER' sections. The 'DEVELOPER' section is active, and 'Self-service client registration' is selected. The main content area is titled 'New Client' and has tabs for 'Main', 'Access', 'Credentials', 'Crypto', 'Other', and 'JSON'. The 'Main' tab is selected. The form contains the following fields:

- Client ID: Will be generated
- Client Secret: Will be generated
- Client Configuration URL: Will be generated
- Registration Access Token: Will be generated
- Client name: Cool SMART App
- Human-readable application name: (empty)
- Redirect URI(s): `https://` and `https://srv.me/app/cool`
- Logo: `https://srv.me/img/cool.png`

Below the Logo field, there is a note: 'URL that points to a logo image, will be displayed on approval page'.

3. In the Access tab make sure that the app is granted the following scopes: launch, launch/patient, launch/encounter, patient/*.read, user/*.*, openid, profile. The code response type should be unchecked.

OpenID Connect Server - x

https://authorize.smartplatforms.org/manage/dev/dynreg/new

OpenID Connect Server Home About Statistics Contact

Home / Client Registration / New

New Client

Save Cancel

Main Access Credentials Crypto Other JSON

Scope

new scope +

launch -

launch/patient -

launch/encounter -

patient/*.read -

user/*.+ -

openid -

profile -

Grant Types

☒ authorization code

☐ client credentials

☐ implicit

☐ redelegate

NYI Response Types

☐ code

☐ token

☐ id_token

4. On the Credentials tab change the authentication method:
 - a. If you're writing a Confidential Client, choose Client Secret over HTTP Basic
 - b. If you're writing a Public Client, choose No authentication (Our method)

OpenID Connect Server - x

https://authorize.smartplatforms.org/manage/dev/dynreg/new

OpenID Connect Server Home About Statistics Contact

Home / Client Registration / New

New Client

Save Cancel

Main Access Credentials Crypto Other JSON

Token Endpoint Authentication Method

☐ Client Secret over HTTP Basic

☐ Client Secret over HTTP POST

☐ Client Secret via symmetrically-signed JWT assertion

☐ Asymmetrically-signed JWT assertion

☒ No authentication

JWK Set

https://

URL for the client's JSON Web Key set

Save Cancel

5. In the Other tab uncheck the Always require that the auth_time claim be sent in the id token and enter the URL(`http://{your-host-name}:8080/fhir-app/`) to your launch.html page in the Initiate Login box.

The screenshot shows the 'New Client' configuration page in the OpenID Connect Server interface. The browser address bar shows `https://authorize.smartplatforms.org/manage/dev/dynreg/new`. The page has a sidebar with 'PERSONAL' and 'DEVELOPER' sections. The 'DEVELOPER' section is active, and 'Self-service client registration' is selected. The main content area is titled 'New Client' and has tabs for 'Main', 'Access', 'Credentials', 'Crypto', 'Other', and 'JSON'. The 'Other' tab is selected. In this tab, there is a checkbox 'Always require that the auth_time claim be sent in the id token' which is unchecked. Below this, there are several fields: 'Default Max Age' (60000), 'Initiate Login' (URL to initiate login on the client), 'Post-Logout Redirect' (URL to redirect the client to after a logout operation), 'Request URIs' (a list with 'https://' and a plus icon), and 'Default ACR Values' (a list with 'new ACR value' and a plus icon). The 'Save' button is at the top left of the configuration area.

6. Click the Save button. The OpenID Connect Server will assign a Client ID and Registration Access Token to your self-service client. Make sure to copy them to a file on your local machine, because you will need them to update your client configuration.
7. You can use the Client ID and Registration Access Token for your self-service client to update or delete its record in the OpenID Connect Server.

PERSONAL

[Manage Approved Sites](#)
[Manage Active Tokens](#)
[View Profile Information](#)

DEVELOPER

[Self-service client registration](#)
[Self-service protected resource registration](#)

[Home](#) / [Client Registration](#) / Edit

Edit Client

Save

Cancel

Main

Access

Credentials

Crypto

Other

JSON

Warning! You MUST protect your **Client ID**, **Client Secret (if provided)**, and your **Registration Access Token**. If you lose your Client ID or Registration Access Token, you will no longer have access to your client's registration records and you will need to register a new client.

Client ID

Client Secret

None (public client)

Client Configuration URL

<https://authorize.smartplatforms.org/register/>

Registration Access Token