Web-Application Starter Document

[**Opening the Scriptchain web application on the local device**](#_wvr3s0pwg8mo) **2**

[Pre-Requisites](#_tv4ggggt3ykp) 2

[Running the Angular frontend](#_z534aj65mxp7) 2

[Running the NodeJs backend](#_k1zn0i64saey) 2

[Running Tests for the angular application](#_kvq14nyrwlq7) 2

Enabling the user-specific portal buttons in home page 3

[**Youtube tutorial for setting up web-application in the local environment**](#_xnvjt7rb5lyp) **3**

[**Manually deploying the application to cloud**](#_ytzetxni783n) **3**

[Pre-Requisites](#_s9hknrsrh4dj) 3

[Deploying the application to cloud via CLI](#_647zvkje8x3q) 3

[**Creating & Using Allscripts Projects**](#_e9rkjmv4yns3) **3**

[Creating Allscripts project](#_r12kyvqp1zv1) 3

[Making HTTP requests to the Allscripts EHR developer sandbox(via Postman)](#_uypw13ld3uzf) 4

[**MongoDB Connection & Data models**](#_5jwudv6gknfd) **5**

[Connecting to the database](#_muxsewb4ivj1) 5

[Data models map for the currently used collection(i.e on 5/11/2020)](#_1ithvikgcfq0) 5

[**Important Notes About Web Application**](#_3ymbofvfkcpj) **6**

## Opening the Scriptchain web application on the local device

#### Pre-Requisites

1. Clone the GitHub repository to your local disk. (Currently - <https://github.com/mohnoori/scriptchain>).
2. Open the command prompt and navigate to the folder that you have cloned.
3. Navigate to web-application in the folder (In the command prompt).
4. NPM (available at <https://npmjs.org>).
5. Angular CLI (available at https://angular.io)

### Running the Angular frontend

1. Navigate to the angular folder and use the command ‘npm install’ which will download the necessary node modules.
2. Then, use the command ‘ng serve’ to start the application on the default port 4200 on your computer.
3. Now the front-end of the application can be accessed at <http://localhost:4200> via any web browser.

### Running the NodeJs backend

1. Navigate to the node folder in the web application and use the command ‘npm install’ which will download the necessary modules.
2. Then, use the command ‘node index.js’ to start the backend server on the default port 8080 on your computer.
3. Now the http requests can be made to the backend server.

### Running Tests for the angular application

1. Navigate to the angular folder in the web application and use the command ‘npm install’ which will download the necessary modules.
2. Then, use the command ‘ng test’ which will compile the files for the execution of the tests, and then a browser will automatically open (default chrome) in which tests run.

Enabling the user-specific portal buttons in the Homepage

1. Currently the html code for these buttons are commented in the home.component file under the directory web-application\angular\src\app\home, to enable these buttons you have to uncomment lines 74-82 in the html file.

## Youtube tutorial for setting up web-application in the local environment

Please follow the below tutorial to set up the application in a local environment (Windows).

<https://www.youtube.com/watch?v=hXFf1Ncwkgw>

## Manually deploying the application to cloud

#### Pre-Requisites

* Gcloud CLI - Available in the google cloud downloads page.
* Access to the ScriptChain project via a Gcloud account.
* Clone the GitHub repository to your local disk. (Currently - <https://github.com/mohnoori/scriptchain>)
* Open the command prompt and navigate to the folder that you have cloned.
* Navigate to web-application in the folder (In the command prompt).

### Deploying the application to cloud via CLI

1. Navigate to the angular folder via the command prompt and then use the command ‘npm install’ to download the necessary node modules.
2. Then, run the command ‘ng build --prod’ to generate/update the dist folder in the node folder. The dist folder contains the static files for the frontend
3. Now navigate to the node folder and use the command ‘gcloud init’ and add/use a gcloud account configuration from which you have access to the Scriptchain cloud project.
4. Now, run the command ‘gcloud deploy’ to deploy the application to the cloud.

## Creating & Using Allscripts Projects

### Creating Allscripts project

1. Log in & open your dashboard from https://developer.allscripts.com.
2. Under the my unity applications section of the page, press ‘+’ to open a page with a form to create a new project.
3. Provide the necessary details and select the type of servers that will be used for requests.

### Making HTTP requests to the Allscripts EHR developer sandbox(via Postman)

1. Open the sandboxes page from the resources from <https://developer.allscripts.com>.
2. Copy the allscripts professional EHR server url.
3. Open postman and use POST request to the copiedURL/json/GetToken with the body of the below format

{

"Username":"**UbiquityId**:Scrip-bd74-Scriptchai-test",

"Password":"Scr0ptChc\*\*\*\*\*\*"

}

1. The ubiquity id can be found in the sandboxes page and the username and password can be found at the project page of the application which can be accessed via the dashboard of the allscripts developer page.
2. Then to register the token, you should use the POST request to copiedURL/json/MagicJson with the body of the below format

{

"Action": "GetUserAuthentication",

"AppUserID": "terry",

"Appname": "ScriptChain.ScriptchainWeb.TestApp",

"PatientID": "",

"Token": "A4AC499F-B259-4F5B-8895-6DD77ED8B910",

"Parameter1": "manning",

"Parameter2": "", "parameter3" : "",

"Parameter4":"", "Parameter5":"", "Parameter6":"", "Data":""

}

AppUserID & parameter1(i.e password) can be found in the sandboxes page.

Token is the token that was generated from the first request.

Appname can be found on the project page of the application that can be accessed from the dashboard from the Allscripts developer portal.

## MongoDB Connection & Data models

### Connecting to the database

The NodeJs server of the web application be it locally deployed or cloud-deployed connects to the same Atlas Mongodb online database. The connection details can be found in web-application/node/db.js

If the database cluster needs to be switched, please make sure to update the URI in the above-mentioned folder.

### Data models map for the currently used collection(i.e on 5/11/2020)

|  |  |  |
| --- | --- | --- |
| Model Name in Application | Collection Name in Cluster | Comments |
| ContactUser | contactUsers | Submissions through contact us page |
| Patient | patients | Patient portal users |
| DeactivatedPatient | deactivatedPatients | Deactivated patient portal users |
| HealthcareProvider | healthcareProviders | Healthcareprovider portal users |
| JobCategory | jobCategories | Categories of jobs available in careers portal |
| JobOpening | jobOpenings | Job openings available in each category |
| NewRequestAccessUser | newusers | Submissions through request access page |
| ResetPasswordToken | resetPasswordTokens | The tokens issued out for the reset of passwords |
| VerifiedUser | verifieduser | Patients who are verified |

MongoDb collections with the naming format such as name.chunks & name.files are used for storing documents in databases with the former storing the actual documents and the latter storing the metadata.

## 

## Important Notes About Web Application

Things to know about the web-application

1. The CI/CD pipeline for the application is done using Google Cloud Build.
2. When new code is pushed or merged to the master branch of the online repository (), a cloud build trigger is invoked which would build and deploy the web application in the scriptchain cloud build project.
3. The average time for the build and deploy lifecycle for the web-application is currently 12-14 minutes.
4. Google cloud build provides 120 minutes of free run time per day and anything beyond is billed as how it is specified by google.
5. This implies that roughly 7 -10 builds per day would be free to be continously deployed to the online cloud project.