SnapClass Installation Guide

## 

## Technology Stack Installation

### Install Node.js + npm

SnapClass runs using Node.js **v14.18.3**. Downloads for Node.js v14.18.3 can be found [here](https://nodejs.org/download/release/v14.18.3/). If you use a different version of Node.js, the project may not compile or dependency installation may fail.

MacOS Users- Select “node-v14.18.3.pkg”

Windows users: Select “win-x86/”

After Node.js is downloaded, open the file and follow the installation wizard. You may optionally download one of the other file options if you know how to install it.

### Install Angular

| npm install -g @angular/cli |
| --- |

(**Note**: Use "sudo npm install -g @angular/cli" to install if you run into any permission errors)

### Install MySQL

An installation for MySQL can be found [here](https://dev.mysql.com/downloads/mysql/). Installation guides for Windows can be found [here](https://dev.mysql.com/doc/refman/5.7/en/windows-installation.html) and MacOS can be found [here](https://dev.mysql.com/doc/refman/5.7/en/macos-installation.html).

When traversing the installation wizard you should install MySQL Server and Workbench. While Workbench is optional it makes managing the database much easier. Port 3306 should be used by MySQL Server. When installing MySQL, do not have to add a password associated with your root account. You can leave the password field blank and select the check button.

(**Note**: If you have errors when logging into SnapClass run this command in the workbench or the terminal ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'insert\_personal\_root\_password'; flush privileges;)

## Project Installation

### Clone the Project with Git

Git will be needed to clone the GitHub repository of SnapClass. Git can be installed [here](https://git-scm.com/downloads). Simply follow the installation wizard or their installation instructions depending on your OS. Find the link to the SnapHub repository and clone it like so:

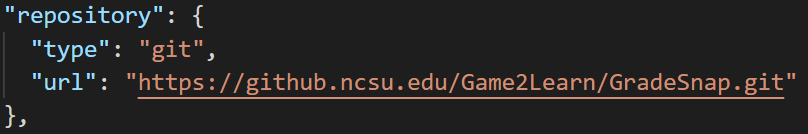
| git clone --recurse-submodules https://github.ncsu.edu/engr-csc-sdc/2022SpringTeam33-STEM.git  Username for 'https://github.ncsu.edu':  Password for 'https://github.ncsu.edu': |
| --- |

It is important to load the submodules so that the Snap environments work correctly. To make sure that the submodules were loaded, check that the “snapclass/cellular” directory is populated with cellular’s source files. If you need to load the submodules after you have already cloned the repository, you can run the following command. Note that submodules may take a while to clone.

| git submodule update --init |
| --- |

### Git Configuration

In "/package.json" update the git repo as needed



## Configuration

### Database Configuration

After MySQL Server is started, you can load the snapclass database with:

Follow the steps below to load the database schema and mock data. If you don’t need the mock data, then skip the second mysql command.

If you have a password associated with your "root" account, run this, and enter you password when it prompts:

| mysql -u root -p < snapclass.sql  mysql -u root -p < snapclass\_mockdata.sql |
| --- |

If you are getting Error 1049, use 'mysql -u root < snapclass.sql' instead.

If you DON’T have a password associated with your "root" account, do this:

| mysql -u root < snapclass.sql  mysql -u root < snapclass\_mockdata.sql |
| --- |

Set the database and server configuration in "snapclass/server/routes/config.js" like so

| var config = {  database: {  host: 'localhost',  user: 'root',  password: '\*password if used\*',  port: 3306,  db: 'snapclass'  },  server: {  host: '127.0.0.1',  port: '8866'  },  secret: 'secret'  }  module.exports = config |
| --- |

#### **Front-end Configuration**

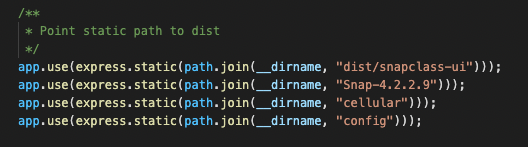
To change the base URL of the Angular app, go to "src/app/services/api.service.ts" and change the baseURL to the appropriate URL. In the case of the stemc server, "<http://stemc.csc.ncsu.edu:8866>". If you’re running locally, use “<http://localhost:8866/>”. Then in "src/index.html" change the base href to "/snapclass".

#### **Backend Configuration**

Configure the host and port in "server/routes/config.js" to the following or as designated by the Game2Learn Lab. There is no specific reason to use "8866", any open port will work. The frontend and database configuration port will need to be changed to use the open port selected.

| var config = {  database: {  host: 'localhost',  user: 'root',  password: '\*password if used\*',  port: 3306,  db: 'snapclass'  },  server: {  host: '127.0.0.1',  port: '8866'  },  secret: 'secret'  }  module.exports = config |
| --- |

The backend also contains a static path reference to the frontend application, Snap*!* version files, and a config file from the old SnapHub project in "/server.js". These do not need to be changed, however, if the route path for the frontend files were to change, the following lines in server.js would need to be updated (may not need to do this).



## Build the App and Start the Server

**Note**: Make sure you are in the snapclass folder directory before building the app

### Install node\_modules

| npm install |
| --- |

### Build the Angular Project

| ng build |
| --- |

### 

### While ng build works, we recommend running the command below instead of ng build before starting the server if you intend to work on the project while it is running. Running this command updates the frontend as changes are made continuously. This way you will not receive an error by rebuilding and running the project after changes are made. If this command is in use, you will need to run the command to start the server in a separate terminal.

| ng build --watch |
| --- |

### Start the Node Server

| node server.js |
| --- |

**Navigate to the App**

Navigate to the appropriate URL that was set in the frontend configuration and the login page for SnapClass should be running.

(If running the application locally)

In your Chrome browser, navigate to "<http://localhost:8866>"