

Florida International University
School of Computing and Information Sciences

Software Engineering Focus

Installation Guide

Project Title: Interactive Smart Table

Team Members: William Escudero

Product Owner(s): Hadi Alhaffar

Mentor(s):

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This document provides a step by step guide on how to install the backend and the application for this project. This document assumes the user has a basic understanding of UNIX/Linux tools and the command line. The first part of this document describes how to install the backend in a Linux environment while the second part describes how to get the application running for macOS.

BACKEND

The backend was written in PHP using the yii2 framework. It uses Apache2 webserver and MySQL as the database. The OS of choice is Ubuntu 18.04.

Installing Apache2, PHP and MySQL

First lets install the latest updates for the OS

```
root@132f793ac7df:/# apt update && apt upgrade -y
```

Once completed lets install Apache2

```
root@132f793ac7df:/# apt install apache2 -y
```

Let's install PHP7

```
root@132f793ac7df:/# apt install php7.2 -y
```

After php7 is installed we have to install the module for Apache2 so that apache is able to understand PHP files

```
root@132f793ac7df:/# apt install libapache2-mod-php
```

Our webserver installation is completed so let's bring up the server to test it

```
root@132f793ac7df:/# service apache2 start
* Starting Apache httpd web server apache2
```

If it gives a warning such as

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain name
```

It can be safely ignore. Now open a web browser and visit <http://localhost> a Apache2 Ubuntu default webpage should appear



At this point the webserver is running with default configuration. Let's test if the PHP installation is working correctly. Go into the default directory where Apache2 looks for files to serve.

```
root@132f793ac7df:~# cd /var/www/html
root@132f793ac7df:/var/www/html#
```

We are going to create a php file that will call the `phpinfo()` function. Type the following

```
root@132f793ac7df:/var/www/html# echo "<?=phpinfo();?>" >> info.php
```

a new file called 'info.php' should exist. Open your web browser again but this time go to <http://localhost/info.php>



Now that we now Apache2 and PHP are installed and working correctly is time to install the MySQL server

```
root@132f793ac7df:/var/www/html# apt install mysql-server -y
```

Let's bring up the service so that we can configure it

```
root@132f793ac7df:/var/www/html# service mysql start
* Starting MySQL database server mysqld
```

Type

```
root@132f793ac7df:/var/www/html# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.25-0ubuntu0.18.04.2 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| performance_schema      |
| sys                     |
+-----+
4 rows in set (0.00 sec)

mysql> quit
Bye
root@132f793ac7df:/var/www/html#
```

At this point we now that mysql is up and running. Next we need to configure it in order to be able to use.

```
root@132f793ac7df:/var/www/html# mysql_secure_installation

Securing the MySQL server deployment.
```

```
Connecting to MySQL using a blank password.
```

```
VALIDATE PASSWORD PLUGIN can be used to test passwords  
and improve security. It checks the strength of password  
and allows the users to set only those passwords which are  
secure enough. Would you like to setup VALIDATE PASSWORD plugin?
```

```
Press y|Y for Yes, any other key for No: n  
Please set the password for root here.
```

```
New password:
```

Set the new password. In this guide I have set the password as test1234

```
By default, a MySQL installation has an anonymous user,  
allowing anyone to log into MySQL without having to have  
a user account created for them. This is intended only for  
testing, and to make the installation go a bit smoother.  
You should remove them before moving into a production  
environment.
```

```
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y  
Success.
```

```
Normally, root should only be allowed to connect from  
'localhost'. This ensures that someone cannot guess at  
the root password from the network.
```

```
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y  
Success.
```

```
By default, MySQL comes with a database named 'test' that  
anyone can access. This is also intended only for testing,  
and should be removed before moving into a production  
environment.
```

```
Remove test database and access to it? (Press y|Y for Yes, any other key for  
No) : y  
- Dropping test database...  
Success.
```

```
- Removing privileges on test database...
```

```
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
root@132f793ac7df:/var/www/html#
```

Now that MySQL server is installed and configured let's install phpMyAdmin which is a web application that will allow you to manage the databases.

```
root@132f793ac7df:/var/www/html# apt install phpmyadmin -y
```

The installation will ask for some configuration parameters

```
Configuring phpmyadmin
-----

The phpmyadmin package must have a database installed and configured before
it can be used. This can be optionally handled with dbconfig-common.

If you are an advanced database administrator and know that you want to
perform this configuration manually, or if your database has already been
installed and configured, you
should refuse this option. Details on what needs to be done should most
likely be provided in /usr/share/doc/phpmyadmin.

Otherwise, you should probably choose this option.

Configure database for phpmyadmin with dbconfig-common? [yes/no] yes

Please provide a password for phpmyadmin to register with the database
server. If left blank, a random password will be generated.

MySQL application password for phpmyadmin:
Password confirmation:
Determining localhost credentials from /etc/mysql/debian.cnf: succeeded.
```

```
Please choose the web server that should be automatically configured to run phpMyAdmin.
```

```
1. apache2 2. lighttpd
```

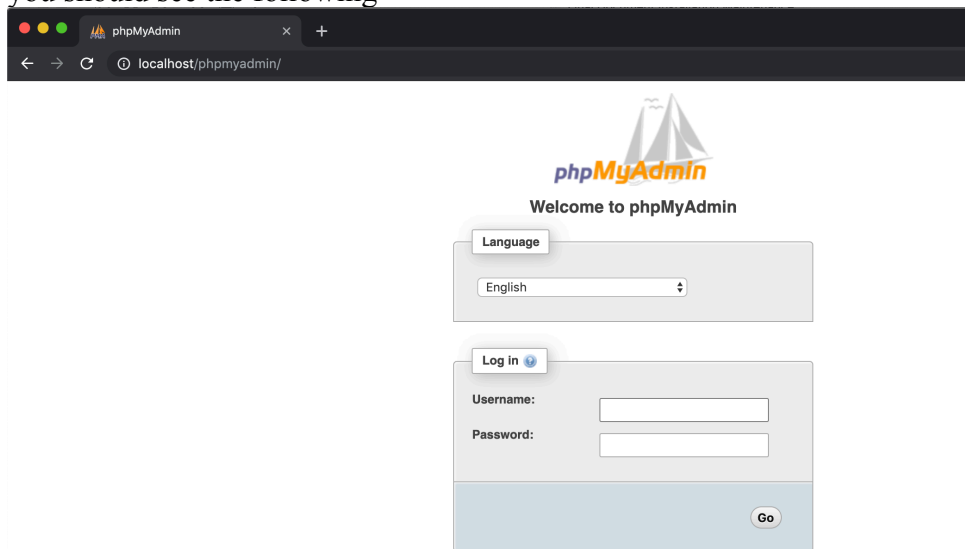
```
(Enter the items you want to select, separated by spaces.)
```

```
Web server to reconfigure automatically: 1
```

Make sure you select apache2 as the web server. When it asks for the password type test1234
In order for the new configuration to take effect we must restart the Apache2 webserver

```
root@132f793ac7df:/var/www/html# service apache2 restart
* Restarting Apache httpd web server apache2
```

After restarting is complete open your web browser and type <http://localhost/phpMyAdmin>
you should see the following



You can log in by using username: phpmyadmin and password: test1234
But we are going to create a new username for the project database which we will create as well.

Go into the terminal and type

```
root@132f793ac7df:/var/www/html# mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 5.7.25-0ubuntu0.18.04.2 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql>
```

Issue the following MySQL command to create the project database

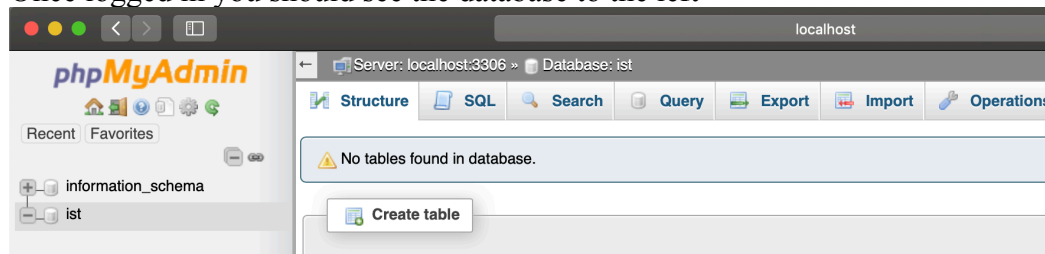
```
mysql> create database ist;
Query OK, 1 row affected (0.00 sec)

mysql>
```

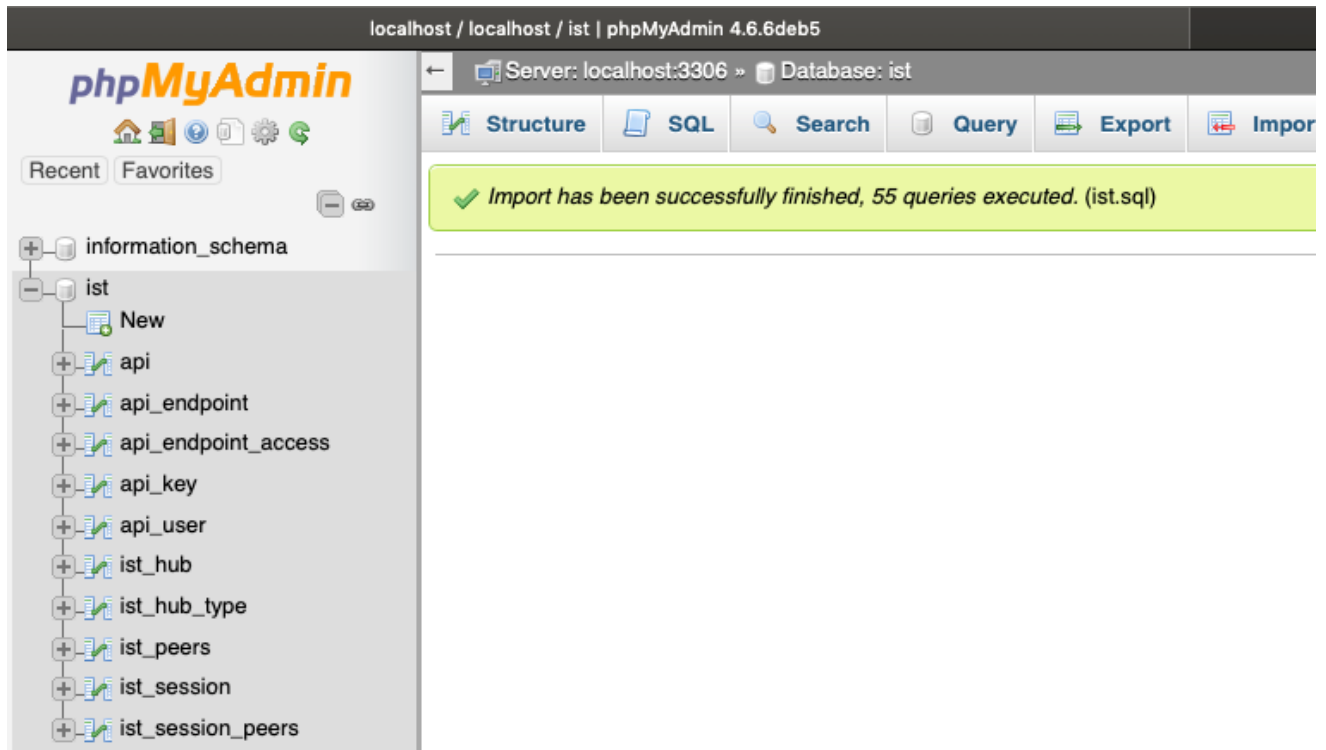
Now we are going to create the user for that database. This user will only have access to the 'ist' database and can be used to manage it through phpmyadmin

```
mysql> grant all privileges on ist.* to 'ist'@'%' identified by 'test1234';
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

Go back to the browser and login into phpMyAdmin with username: ist and password: test1234
Once logged in you should see the database to the left



We are now going to populate the database with tables and testing data to be able to run the project. Click on the database 'ist' and then 'import' on the right, click where it says 'Choose File' and look for the file called 'ist.sql' in the project folder



The tables and the data are now imported. Next we are going to install the source code of the project. Go to the folder where apache looks for files

```
root@132f793ac7df:~# cd /var/www/html
root@132f793ac7df:/var/www/html#
```

If we are getting the source code from JIRA we can go ahead and do git clone from bitbucket

```
root@132f793ac7df:/var/www/html# git clone https://sp-jira.cis.fiu.edu:7443/scm/intr/ist-backend.git
Cloning into 'ist-backend'...
```

```
root@132f793ac7df:/var/www/html# ls
index.html info.php ist-backend
root@132f793ac7df:/var/www/html# cd ist-backend/
root@132f793ac7df:/var/www/html/ist-backend# ls -l
total 184
drwxr-xr-x 2 root root 4096 Apr 27 00:07 assets
-rw-r--r-- 1 root root 773 Apr 27 00:07 codeception.yml
drwxr-xr-x 2 root root 4096 Apr 27 00:07 commands
drwxr-xr-x 2 root root 4096 Apr 27 00:07 components
-rw-r--r-- 1 root root 2116 Apr 27 00:07 composer.json
-rw-r--r-- 1 root root 121328 Apr 27 00:07 composer.lock
drwxr-xr-x 2 root root 4096 Apr 27 00:07 config
drwxr-xr-x 2 root root 4096 Apr 27 00:07 controllers
drwxr-xr-x 3 root root 4096 Apr 27 00:07 mail
drwxr-xr-x 4 root root 4096 Apr 27 00:07 modules
drwxr-xr-x 2 root root 4096 Apr 27 00:07 runtime
drwxr-xr-x 9 root root 4096 Apr 27 00:07 tests
drwxr-xr-x 4 root root 4096 Apr 27 00:07 views
drwxr-xr-x 5 root root 4096 Apr 27 00:07 web
drwxr-xr-x 2 root root 4096 Apr 27 00:07 widgets
-rwxr-xr-x 1 root root 553 Apr 27 00:07 yii
-rw-r--r-- 1 root root 515 Apr 27 00:07 yii.bat
root@132f793ac7df:/var/www/html/ist-backend#
```

Now we have to reconfigure apache2 in order to change the directory in which it looks for files to /var/www/html/ist-backend/web

So let's edit the configuration file. Any editor would work, I am using vim in this instance

```
root@132f793ac7df:/var/www/html/ist-backend# vim /etc/apache2/sites-enabled/000-default.conf
```

Change the line

```
12 DocumentRoot /var/www/html
```

to

```
12 DocumentRoot /var/www/html/ist-backend/web
```

Save and exit. We now have to restart the apache web server for the changes to take effect.

```
root@132f793ac7df:/var/www/html/ist-backend# service apache2 restart
* Restarting Apache httpd web server apache2
```

We are now going to work on configuring the application. We first need to install 'composer' which is package manager for PHP. This application is required to update the code base and install required packages.

```
root@132f793ac7df:/var/www/html/ist-backend# apt install composer -y
```

Now we need to update the current directory in order to download all dependencies required

```
root@132f793ac7df:/var/www/html/ist-backend# composer update
```

A long list should start installing. Once it finishes with no error all dependencies have been installed. Now we need to configure the database connection for the application by going into the config folder and editing db.php

Set username to 'ist' and password to 'test1234'

```
root@132f793ac7df:/var/www/html/ist-backend# cd config
root@132f793ac7df:/var/www/html/ist-backend/config# ls
console.php db.php params.php test.php test_db.php web.php
root@132f793ac7df:/var/www/html/ist-backend/config#
```

We need to grant access to the web server to some of the folders within the base code.

```
root@132f793ac7df:/var/www/html/ist-backend/config# cd ..
root@132f793ac7df:/var/www/html/ist-backend# chown www-data:www-data -R runtime
```

www-data is the user under which the Apache2 daemon is running.

Lastly we need enable the url rewrite module for apache so that the application can make use of pretty urls (/hub/session instead of index.php?r=hub/default/session)

```
root@132f793ac7df:/var/www/html/ist-backend# a2enmod
Which module(s) do you want to enable (wildcards ok)?
rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
    service apache2 restart
root@132f793ac7df:/var/www/html/ist-backend#
```

```
root@132f793ac7df:/var/www/html/ist-backend# service apache2 restart
* Restarting Apache httpd web server apache2
```

After restarting apache we must create a .htaccess file where apache looks for files.

```
root@132f793ac7df:/var/www/html/ist-backend# cd web
root@132f793ac7df:/var/www/html/ist-backend/web# touch .htaccess
root@132f793ac7df:/var/www/html/ist-backend/web# ls -la
total 36
drwxr-xr-x  5 root root 4096 Apr 27 00:29 .
drwxr-xr-x 16 root root 4096 Apr 27 00:17 ..
-rw-r--r--  1 root root    0 Apr 27 00:29 .htaccess
drwxr-xr-x  2 root root 4096 Apr 27 00:07 assets
drwxr-xr-x  2 root root 4096 Apr 27 00:07 css
-rw-r--r--  1 root root  318 Apr 27 00:07 favicon.ico
-rw-r--r--  1 root root  498 Apr 27 00:07 index-test.php
-rw-r--r--  1 root root  367 Apr 27 00:07 index.php
drwxr-xr-x  2 root root 4096 Apr 27 00:07 js
-rw-r--r--  1 root root   23 Apr 27 00:07 robots.txt
root@132f793ac7df:/var/www/html/ist-backend/web#
```

Edit the file and add the following lines

```
1 RewriteEngine On
2 RewriteBase /opt/ist/ist/
3
4 RewriteCond %{REQUEST_FILENAME} !-f
5 RewriteCond %{REQUEST_FILENAME} !-d
6 # Otherwise forward it to index.php
7 RewriteRule . /index.php
~
```

Save the changes. Now we need to add a <Directory> directive to the apache configuration to make the new .htaccess work.

Edit the file and add the lines below

```
root@132f793ac7df:/var/www/html/ist-backend/web# vim /etc/apache2/sites-enabled/000-default.conf
```

```
15 <Directory /var/www/html/ist-backend/web>
16     AllowOverride All
17 </Directory>
```

At this point the backend should be installed and fully configured.

APPLICATION

The application was written using the Electron JS framework so in order to get the application running we must install node JS and NPM. NPM is the package manager for node.

```
wbox:~ william$ brew install npm
```

Now we can download the source from the repository

```
wbox:guide william$ git clone https://sp-jira.cis.fiu.edu:7443/scm/intr/ist-electron.git
```

```
wbox:guide william$ cd ist-electron/  
wbox:ist-electron william$
```

Now we need to run `npm update` in order to download the application dependencies

```
wbox:ist-electron william$ npm update
```

We are now ready to run the application

```
wbox:ist-electron william$ npm start  
  
> ist@1.0.0 start /Users/william/guide/ist-electron  
> electron .
```

At this point the application should be running and ready to be used.