

# QuizZing by Undefined Parameter

## README

### About the Project Setup:

QuizZing is built on the Dropwizard framework, which is a Java based framework that packages several popular libraries together to make building RESTful services easier. Our directory structure is set up according to the suggested structure laid out by the creators of Dropwizard.

### The Structure:

**Java files can be found in the following directory:**

**`/UndefinedParameter/QuizZing/src/main/java/com/UndefinedParameter`**

In this directory there are a few sub folders.

The `/app/resources` subfolder contains the controller level objects used to act as communication from the HTTP server and the model. They are responsible for handling the HTTP requests, communicating to the model, and then returning the formatted view.

The `/app/core` subfolder contains all of the java objects used to support the data. Some of these are pojo classes, and some have a bit more logic to them. Also in this folder you will find the Manager classes, which are the model level classes that the resources (controllers) use to communicate with the database. No resource is allowed to touch the database objects, it must go through a manager class.

The `quizzing` subfolder contains objects used to setup and run the Dropwizard framework, as well as the authenticator used to handle authentication.

The `/views` subfolder is the java objects responsible for serving the view. These work together with Freemarker to template java methods into the html response.

The `/jdbi` subfolder contains the database objects and the mappers used to communicate to the mysql server, and map the results into the core objects.

**The HTML / CSS / JS and other assets are found in the following directory:**

**`/UndefinedParameter/QuizZing/src/main/resources`**

From here there are two main branches. First the `/com/UndefinedParameter` contains both the views and includes folder. These are the ftl files that Freemarker uses as templates to convert into the HTML responses. For the most part these look just like HTML.

The other branch is the /assets folder, which contains all our plugins, scripts, css, backgrounds, and images. These are all included in the .ftl files as needed.

## How to build the project:

**Note: the following steps will talk about how to build this project from scratch. However provided is a .jar file that is prebuilt and can launch the server. The only thing necessary to have is Java – Since the Dropwizard framework packages all dependencies into the jar file. Skip to how to run the server if you wish to skip the setup.**

## Requirements:

QuizZing is powered by Dropwizard. In order to build the project, maven is required. Here is the instructions on how to set up maven:

Maven can be downloaded here:

<http://maven.apache.org/download.cgi>

Instructions on how to set up maven are here:

[http://maven.apache.org/run-maven/index.html#Quick\\_Start](http://maven.apache.org/run-maven/index.html#Quick_Start)

## Building the Project:

We have provided a .pom file that maven will use to build all our dependencies required. Once Maven is successfully installed, through the command prompt navigate to /UndefinedParamter/QuizZing directory. From this directory run the command 'mvn package.'

Running the maven package will set up all the resources and create a file called 'QuizZing-1.0-SNAPSHOT' which can be found in the /UndefinedParameter/QuizZing/target directory.

## Running the Project:

Once the jar is created all you need to do is run the jar. Note that dropwizard packages all necessary apache libraries into the jar, therefore no further set up is needed to run this. Simply navigate to the /UndefinedParameter/QuizZing/target folder and type the command 'java -jar QuizZing-1.0-SNAPSHOT' and provide the command line arguments of 'server <path to QuizZing.yml>.' The QuizZing.yml file is found under the /UndefinedParameter/QuizZing folder. So for example in linux the command could be 'java -jar QuizZing-1.0-SNAPSHOT server ../QuizZing.yml.'

If you are still having problems running it, simply copy the yml file into the target folder and then you can run it like java -jar QuizZing-1.0-SNAPSHOT server QuizZing.yml.

Once the server is running, you can now switch to a web browser, we suggest Google Chrome, and go to localhost:8080 to access the page. If for some reason your port 8080 is in use, you may change this in the QuizZing.yml file and rerun the server using the same previous steps.

**That is everything necessary to build and run our project.**