**Word Game**

*Word Game is a Scrabble-like word game played through a REST API*

**Source Code**

The source code is hosted by GitHub.

* Web URL: <https://github.com/michaelboyd/wordgame>
* Repo URI: <https://github.com/michaelboyd/wordgame.git>

**Software Requirements**

* Java 1.8
* Maven
* Internet connection

**Build and Deployment**

Word Game is built with Spring Boot and uses Maven for building and deploying.

To Build: Build and deploy Word Game from the command line.

Navigate to the project’s root folder and execute the command:

mvn spring-boot:run

The above command will download the necessary dependencies for running the application.

Data is stored in an in-memory H2 database that is empty each time the application is started.

After executing spring-boot:run, the application should be running.

The database can be viewed using the H2 console at the following URL:

<http://localhost:8080/h2-console/>

Note: set JDBD URL to jdbc:h2:mem:testdb

**Playing Word Game**

The following examples use CURL to send requests to the Word Game REST API.

Please note: the project file, “commands” has copy-and-paste-to-terminal examples.

**Create a new Game with Default board size of 15x15**

curl -X POST http://localhost:8080/api/match/create

**Create a new Game with custom Board Size**

(Recommend board size of between 10 and 50)

curl -X POST [http://localhost:8080/api/match/create?boardSize={int}](http://localhost:8080/api/match/create?boardSize=%7bint%7d)

**View the Game Board**

curl -X GET http://localhost:8080/api/match/{id}/board

**Play a Word!**

Required Parameters:

* X – horizontal coordinate
* Y – vertical coordinate
* word – word to place on board
* direction – either down or right

The top left corner of the Board is coordinate 0,0.

Coordinates must be positive and move right and down on the board.

Plays are validated. If validation fails, 404 (bad request) is returned.

Example of sending Play Word request:

curl -i -X POST -H "Content-Type:application/json"

-d '{"x":0,"y":0,"word":"example","direction":"right"}' http://localhost:8080/api/match/1/play

**Other Functions**

**List all Active Games**

curl -X GET http://localhost:8080/api/match/list

**List Games by their active status**

curl -X GET http://localhost:8080/api/match/list?active={true/false}

**Get a Match's Play History (most recent first)**

curl -X GET http://localhost:8080/api/match/{id}/play/history

**Update the active status for a Match**

curl -X PUT http://localhost:8080/api/match/{id}/update?active={true/false}

**Delete a Match**

curl -i -X DELETE http://localhost:8080/api/match/{id}/delete