

Service Layers (ShineZone)

Service Endpoint and Purpose

<https://s3.us-east-1.amazonaws.com/>

Domain address created using AWS where the webpages will reside.

Purpose: User will be able to navigate to this web address to view and play the game. Every page associated with the game will be hosted here.

Web Domain and Bucket for hosting

[https://s3.us-east-1.amazonaws.com/swdv691cap/\[page_name\]](https://s3.us-east-1.amazonaws.com/swdv691cap/[page_name])

Associates created S3 bucket which hosts the site to correct page related to the user input. The application webpages will all be linked to the domain listed above.

Purpose: User will be able to see all changes made to the site to allow the gameplay to function appropriately.

Examples of Services Used

SubmitScore: Service would be applied once user completes game. The score would be recorded and added to the leaderboards. *Associated with Completion screen and Leaderboard*

type: "POST",

url: 'submit.php',

data: { name: \$('#uName').val(), email: \$('#uEmail').val(), type: type, score: score },

success: submitScoreSuccess,

dataType : 'json'

Sample Response: Database connection error or Submission error,

please try again.

LoadLeaderboard: Used when the user requests to see the leaderboard
Associated with leaderboard

```
type: "POST",  
url: 'leaderboard.php',  
data: { type: type, reverse:scoreReverse, limit:pageLimit },  
success: loadScoreboardSuccess,  
dataType : 'json'
```

Sample Response:

“Database Connection Error”

StartGame: (probably to be renamed) but user would request to play game which would send GET request to load game questions

Associated with home landing screen and game start screen

```
Type: “GET”,  
url: src, note:(index.html and game.js would initialize game)  
datatype: “xml”,
```

Sample response:

```
<item>  
  <category>WORD GAME</category>  
  <landscape>  
    <question type="1"><![CDATA[GAME,MAGE,MEGA]]></question>  
    <description type="text" CDATA[Form word using available  
letters]></description>  
  </landscape>  
</item>
```

ADDED SERVICE POINTS

User Get: Verifying user login

- a. 200 Successful Get

```
{
  "userID" : 0,
  "username": "string",
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "password": "string",
}
```
- b. 400: Invalid Password

```
{
  "code": 1,
  "type": "error",
  "message": "Incorrect Password"
}
```
- c. 404 Error No User Found!

Stretch Feature: Password Reset Post

- a. 200 Successful Post

```
{
  "username": "string",
  "email": "string",
}
```
- b. 400 Invalid Email or User

```
{
  "code": 1,
  "type": "error",
  "message": "Username or Email not found"
}
```
- c. 404: Error No User Found!