# Backend Documentation

*Updated 11/09/2021*

This backend service utilizes the Spring framework and will be used as the template for our team to build features.

**Amazon Relational Database Service (RDS)**
The Amazon RDS database is currently live and will work with this service.

I have created some basic endpoints so that you can test queries to the database hosted on AWS.

I suggest using Postman and MySQL Workbench when making requests to the endpoints so that you can verify the data.

**GET endpoints**

| | |
|---|---|
| /home | Returns HTML document |
| /get-users | Returns JSON array of all users |
| /get-bugs | Returns JSON array of all bugs |

**POST endpoints**

| | |
|---|---|
| /create-bug | Inserts a bug into the database |

**Request Body**

| | |
|---|---|
| title | string |
| bug_description | string |
| *due_date | string |
| assigned_to | string |
| created_by | string |
| severity | string |
| bug_status | string |

*due_date requires the following formatting: YYYY-MM-DD

**Request example (JSON)**

```
{
   "title": "Error on page",
   "bug_description": "There is an error on the page",
   "due_date": "2021-07-11",
   "assigned_to": "1",
   "created_by": "1",
   "severity": "HIGH",
   "bug_status": "OPEN"
}
```

I added a value generator so that the id in the database entry automatically increments.

**Amazon AWS Role accounts**
~~I have created AWS role accounts with limited functionality for the team. I am currently in the process of testing them and hope to have them ready by the end of the week. With these accounts you will be able to access the database.~~

Instead of using IAM role accounts I think it would be best if we just used one account for the database. The account credentials should be used as environment variables on your computer. I will distribute the credentials later this week.

Your IP address is required so that I can add you to the RDS instance security group.

**Requirements**
- Spring
- Maven
- MySQL Community Workbench
- IDE (I recommend IntelliJ)

**Optional**
- Postman (to test API requests/responses)

**Additional Dependencies**
Dependencies are included in the POM document and should download automatically.

Spring Security dependencies are not included in this template but can be added later.

**Setup**
This service is ready to go with the only exception being the properties file which houses key/value pairs that Spring uses for application configuration.

Spring configuration contains sensitive information so I removed the values. Project configuration details will be provided to the team later this week.

Location of configuration file within the file hierarchy:
```
src → main → resources → application.properties
```

More information on application properties can be found here:
https://docs.spring.io/spring-boot/docs/current/reference/html/application-properties.html

**Getting Files From The Repository**
Once the backend is approved it will be merged to the main branch on Github. From there you will be able to clone the repository to your local machine.

*****IMPORTANT*****
After cloning, please make sure that that you create a separate working branch for yourself. Only approved work should be merged to the main branch.