**Generator Assessment API Documentation**

**BASE URL** = <http://localhost:8080/api>

**Authentication (Basic Auth)**

**User** = gentlest

**Password** = genTest123$

**Github url** - <https://github.com/fredallotey2000/GeneratorTest.git>

**Docker image url** - <https://hub.docker.com/repository/docker/fredallotey2000/generatortest>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Request description** | **Request method** | **Endpoint** | **Example** | **Note** |
| See details of a single power plant | GET | /plants/{plantId} | http://localhost:8080/api/plants/5 | plantId is the sequence number in the excel extract. It is the only unique identifier |
| Filter power plants by location with pagination | GET | /plants?location code&page={n}&size={n} | http://localhost:8080/api/plants?location=AK&page=1&size=2 | You can get the results with or without pagination |
| Get top n generators by annual generation | GET | /plants?top={n}  &sort=(ass|desc) | http://localhost:8080/api/plants?top=5&sort=desc |  |
| Get bottom n generators by annual generation | GET | /plants?bottom={n}  &sort=(ass|desc) | http://localhost:8080/api/plants?bottom=5&sort=asc |  |
|  |  |  |  |  |

Note:

1. **Results are cached** and requests are **run concurrently**
2. **Data is imported** to the **inbuilt h2 database** of spring
3. Results include both the actual and percentage values of the plants annual generation by location
4. Solution was developed using **JAVA and spring boot**
5. Docker image was built from **adoptopenjdk/openjdk11:ubi**
6. **Test units for Repository and Service** layer for all scenarios
7. Ignored HAETOS to increase speed
8. **Percentages** were rounded **to 4 decimal** to show very small percentages
9. **Pagination** has a maximum size of **40**

**Disclaimer**

I wish I had more time to complete my Controller test with MockMvc.

**THANK YOU**