Sentiment-detector :

Introduction : This is an api program developed using quarkus java framework. By using this api users/developers can fetch the toxicity of the sentences programmatically.

Syntax - - https://sentiment-detector.herokuapp.com/sentiment/sentences=<sentences>

Here in above link you can pass any number of sentences to check their toxicity.
Below is the request and response information which gets transmitted to/from the client machine/server.

Example : https://sentiment-detector.herokuapp.com/sentiment/sentence=You are the shittiest person I ever seen. Have a great day ahead

So the result would be as below :


```
[
    {
        "sentence":"You are the shittiest person I ever seen",
        "result":"0"
    },
    {
        "sentence":"Have a great day ahead",
        "result":"1"
    }
]
```


```
Here in above result you can see the sentence with result 0 is toxic
while other is not.
```

Link to deployed solution

https://sentiment-detector.herokuapp.com/

This is fully functional API for parsing text to check if its sensitive(which simply leads to stress) or normal.

To test this on your local machine

The source code for this is present on github :

https://github.com/ravindra98/sentiment-detector

You can clone this repository to your local machine using command

git clone https://github.com/ravindra98/sentiment-detector.git

If result gets 1 the sentence is not sensitive if result gets 0 the sentence is sensitive.
#You can pass as much as sentences Separated by full stop

Sentiment-detection with quarkus

This is a minimal CRUD service exposing a couple of endpoints over REST.

Under the hood, this demo uses:

- RESTEasy to expose the REST endpoints
- REST-assured and JUnit 5 for endpoint testing

Requirements

To compile and run this demo you will need:

- JDK 1.8+
- GraalVM

Configuring GraalVM and JDK 1.8+

Make sure that both the `GRAALVM_HOME` and `JAVA_HOME` environment variables have been set, and that a JDK 1.8+ `java` command is on the path.

See the Building a Native Executable guide for help setting up your environment.

Its not necessary to install GRAALVM to test this application. You can skip this part.

Building the application

Launch the Maven build on the checked out sources of this demo:

```
./mvnw install
```

Live coding with Quarkus

The Maven Quarkus plugin provides a development mode that supports live coding. To try this out:

```
./mvnw quarkus:dev
```

This command will leave Quarkus running in the foreground listening on port 8080.

1. Visit the default endpoint: http://127.0.0.1:8080.
    - You will see the page accepting the text in text input box
    - button to submit
2. Visit the `/sentiment/` endpoint: http://127.0.0.1:8080/sentiment
    - If you see success then the sentiment detector is ready to be accessed
3. Try to send the queries to http://127.0.0.1:8080/sentiment/sentence={text}

- make sure to replace {text} part with your text which you want to check if its abusive/sensitive or not

Run Quarkus in JVM mode

When you're done iterating in developer mode, you can run the application as a conventional jar file.

First compile it:

```
./mvnw install
```

Then run it:

```
java -jar ./target/getting-started-1.0-SNAPSHOT-runner.jar
```

Have a look at how fast it boots, or measure the total native memory consumption.

Run Quarkus as a native executable

You can also create a native executable from this application without making any source code changes. A native executable removes the dependency on the JVM: everything needed to run the application on the target platform is included in the executable, allowing the application to run with minimal resource overhead.

Compiling a native executable takes a bit longer, as GraalVM performs additional steps to remove unnecessary codepaths. Use the `native` profile to compile a native executable:

```
./mvnw install -Dnative
```

After getting a cup of coffee, you'll be able to run this executable directly:

```
./target/sentiment-detector-1.0-SNAPSHOT-runner
```