Name:Akshay Gadhave
Roll No:221076 SY
Group No: 15
Comp A3

PBL

Topic:

Coronavirus tracker made using Java and Spring Boot.(Worldwide)

Abstract:

The coronavirus COVID-19 is affecting 213 countries and territories around the world and 2 international conveyances. It’s very important to keep track about daily statistics, so we created a tracker to keep track of the total number of positive cases all around the world. Our application gets updated after midnight GMT+0.

Discripton:

We created this application in Java 14 and using spring boot framework.

Spring Boot

Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can just run. You can get started with minimum configurations without the need for an entire Spring configuration setup.

Spring Boot automatically configures your application based on the dependencies you have added to the project by using @EnableAutoConfiguration annotation. For example, if MySQL database is on your classpath, but you have not configured any database connection, then Spring Boot auto-configures an in-memory database.

The entry point of the spring boot application is the class contains @SpringBootApplication annotation and the main method.Spring Boot automatically scans all the components included in the project by using @ComponentScan annotation

Handling dependency management is a difficult task for big projects. Spring Boot resolves this problem by providing a set of dependencies for developers convenience. Below is example of dependencies in our project in porm.xml file.

```
 <name>CoronaVirus</name>
<description>Corona Virus tracker with java and spring   boot</description>

<properties>
    <java.version>14</java.version>
</properties>

<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
        <optional>true</optional>
    </dependency>

    <dependency>
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-csv</artifactId>
        <version>1.8</version>
    </dependency>


    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
        <exclusions>
            <exclusion>
                <groupId>org.junit.vintage</groupId>
                <artifactId>junit-vintage-engine</artifactId>
            </exclusion>
        </exclusions>
    </dependency>
</dependencies>
```

Data

All data is gathered from Covid19 CSV and is updated on a daily basis.

Main Class

CoronaVirusApplication.java manages all other classes in our program. We have used @SpringBootApplication to define it as a spring application and @EnableScheduling to run live forever.

//CoronaVirusApplication.java

```
package com.example.CoronaVirus;


import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.scheduling.annotation.EnableScheduling;
```

```
@SpringBootApplication
@EnableScheduling
public class CoronaVirusApplication {

    public static void main(String[] args) {
        SpringApplication.run(CoronaVirusApplication.class, args);
    }

}
```

Services

Service Components are the class file which contains @Service annotation. These class files are used to write business logic in a different layer, separated from @RestController class file. We are not creating a REST Service rather we are processing data and displaying it.

```
//CoronaDataServices.java
package com.example.CoronaVirus.services;

import com.example.CoronaVirus.models.LocationStats;
import org.apache.commons.csv.CSVFormat;
import org.apache.commons.csv.CSVRecord;
import org.springframework.scheduling.annotation.Scheduled;
import org.springframework.stereotype.Service;

import javax.annotation.PostConstruct;
import java.io.FileReader;
import java.io.IOException;
import java.io.Reader;
import java.io.StringReader;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.*;
import java.net.http.HttpResponse;
import java.util.ArrayList;
import java.util.List;



@Service
public class CoronaVirusDataServices {


private static String VIRUS_DATA_URL =
"https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv";


private List<LocationStats> allStats = new ArrayList<>();
public List<LocationStats> getAllStats(){
return allStats;
}
@PostConstruct
@Scheduled(cron="* * 1 * * *")
public void fetchVirusData() throws IOException, InterruptedException {


List<LocationStats> newStats = new ArrayList<>();


HttpClient client = HttpClient.newHttpClient();
HttpRequest request =
HttpRequest.newBuilder().uri(URI.create(VIRUS_DATA_URL)).build();


HttpResponse<String> httpResponse = client.send(request,
HttpResponse.BodyHandlers.ofString());

//System.out.println(httpResponse.body());

StringReader csvBodyReader = new StringReader(httpResponse.body());

Iterable<CSVRecord> records =
CSVFormat.DEFAULT.withFirstRecordAsHeader().parse(csvBodyReader);

for (CSVRecord record : records) {

LocationStats locationStat = new LocationStats();

locationStat.setState(record.get("Province/State"));

locationStat.setCountry(record.get("Country/Region"));

int latestCase = Integer.parseInt(record.get(record.size() - 1));

int prevDayCase = Integer.parseInt(record.get(record.size() - 2));

locationStat.setLatestTotalCases(latestCase);

locationStat.setDiffFromPrevDay(latestCase - prevDayCase);


System.out.println(locationStat);

newStats.add(locationStat);


}

this.allStats = newStats;

}

}
```

Models and Controllers

In Spring MVC, the model works a container that contains the data of the application. Here, a data can be in any form such as objects, strings, information from the database, etc.

It is required to place the Model interface in the controller part of the application. The object of HttpServletRequest reads the information provided by the user and pass it to the Model interface. Now, a view page easily accesses the data from the model part.

```
}



//LocationStats.java

package com.example.CoronaVirus.models;


public class LocationStats {

private String state;

private String country;

private int latestTotalCases;

private int diffFromPrevDay;


public int getDiffFromPrevDay() {

return diffFromPrevDay;

}


public void setDiffFromPrevDay(int diffFromPrevDay) {

this.diffFromPrevDay = diffFromPrevDay;

}


public String getState() {

return state;

}


public void setState(String state) {

this.state = state;

}


public String getCountry() {
```

```
return country;
}

public void setCountry(String country) {
this.country = country;
}

public int getLatestTotalCases() {
return latestTotalCases;
}

public void setLatestTotalCases(int latestTotalCases) {
this.latestTotalCases = latestTotalCases;
}

@Override
public String toString() {
return "LocationStats{" +
"state='" + state + '\'' +
", country='" + country + '\'' +
", latestTotalCases=" + latestTotalCases +
'}';
}
}
```

Classic controllers can be annotated with the *@Controller* annotation. This is simply a specialization of the *@Component* class and allows implementation classes to be autodetected through the classpath scanning.

@Controller is typically used in combination with a *@GetMapping* annotation used on request handling methods.

```
//HomeController.java

package com.example.CoronaVirus.controllers;


import com.example.CoronaVirus.models.LocationStats;

import com.example.CoronaVirus.services.CoronaVirusDataServices;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;


import java.util.List;


@Controller

public class HomeController {


@Autowired

CoronaVirusDataServices coronaVirusDataServices;


@GetMapping("/")

public String home(Model model){

List<LocationStats> allStats = coronaVirusDataServices.getAllStats();
int totalReportedCases = allStats.stream().mapToInt(stat ->
stat.getLatestTotalCases()).sum();

int totalNewCases = allStats.stream().mapToInt(stat ->
stat.getDiffFromPrevDay()).sum();

model.addAttribute("locationStats", allStats);

model.addAttribute("totalReportedCases", totalReportedCases);

model.addAttribute("totalNewCases", totalNewCases);


return "home";

}

}
```

UI

Our Application is rendered on home.html

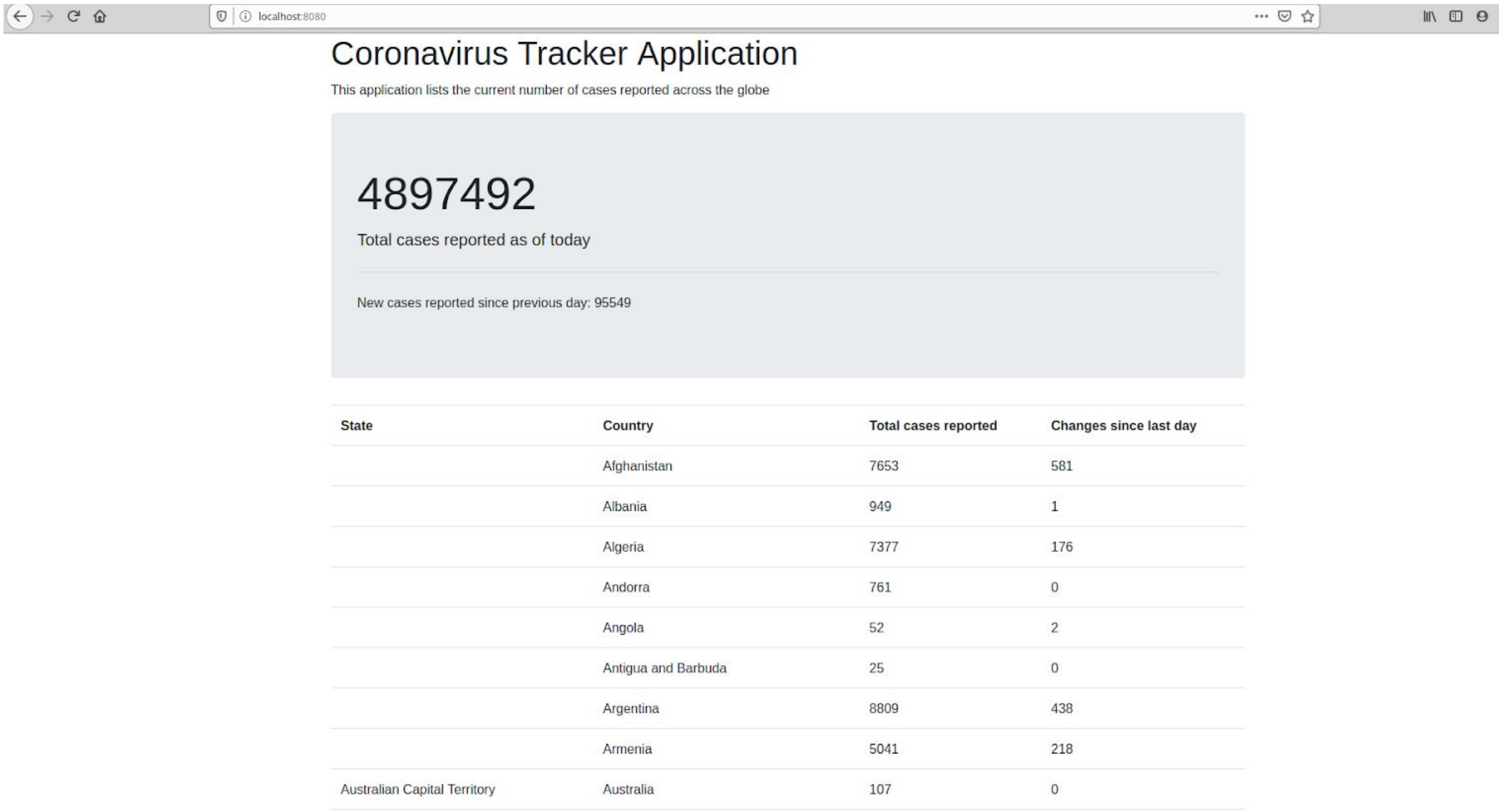
```
<!DOCTYPE html>


<html xmlns:th="http://www.thymeleaf.org">


<head>

<title>Coronavirus Tracker Application</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">

</head>

<link rel="stylesheet" href="style.css">


<body>

<div class="container">
```

```
<h1>Coronavirus Tracker Application</h1>
<p>This application lists the current number of cases reported across the globe</p>

<div class="jumbotron">
<h1 class="display-4" th:text="${totalReportedCases}"></h1>
<p class="lead">Total cases reported as of today</p>
<hr class="my-4">
<p>
<span>New cases reported since previous day:</span>
<span th:text="${totalNewCases}"></span>
</p>

</div>

<table class="table">
<tr>
<th>State</th>
<th>Country</th>
<th>Total cases reported</th>
<th>Changes since last day</th>
</tr>
<tr th:each="locationStat : ${locationStats}">
<td th:text="${locationStat.state}"></td>
<td th:text="${locationStat.country}"></td>
<td th:text="${locationStat.latestTotalCases}">0</td>
<td th:text="${locationStat.diffFromPrevDay}">0</td>
</tr>
</table>
```

```
</div>

</body>



</html>
```

Application is now hosted on localhost, can be transferred to production as JUnit tests are also performed.

Output:



Coronavirus Tracker Application

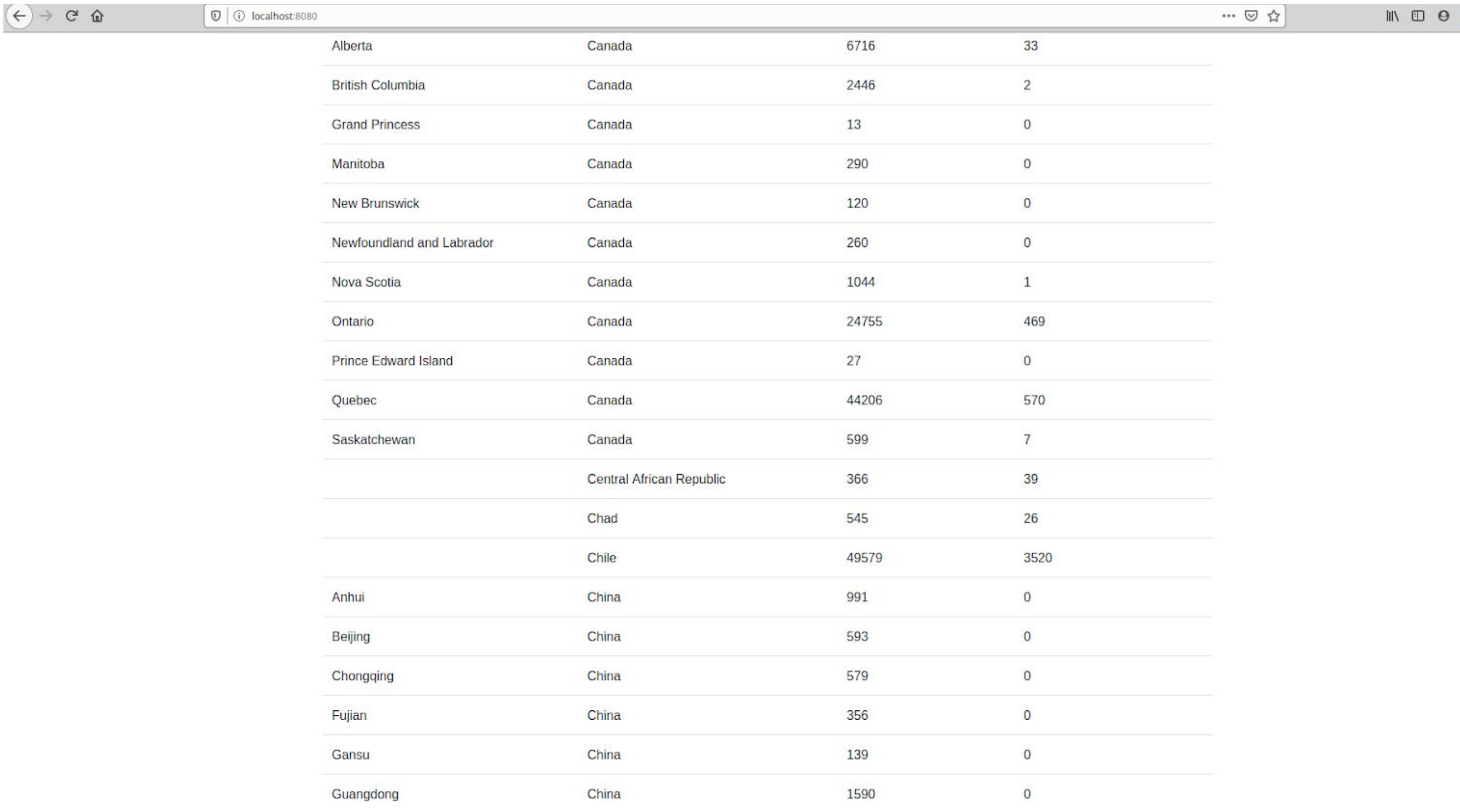
This application lists the current number of cases reported across the globe

4897492

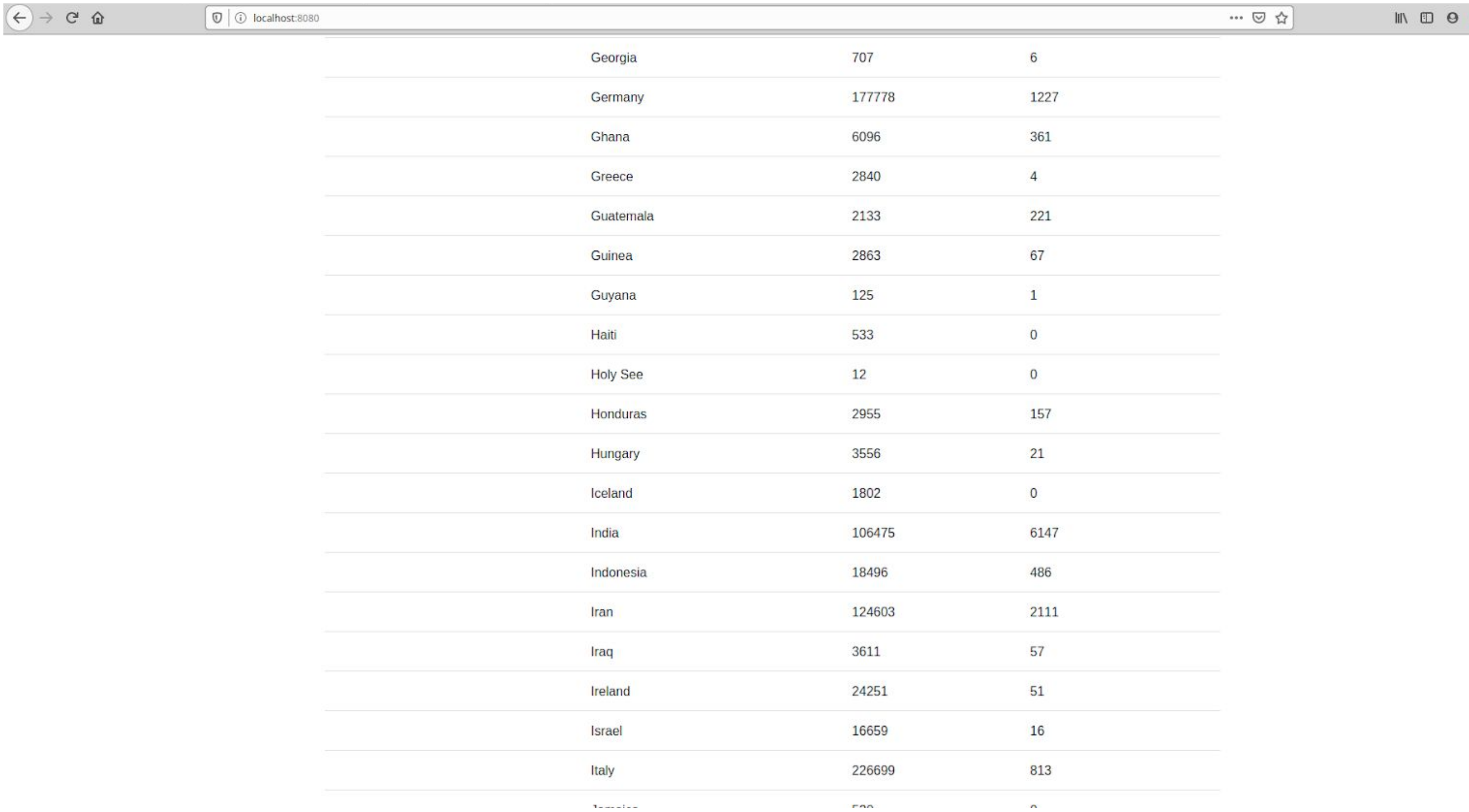
Total cases reported as of today

New cases reported since previous day: 95549

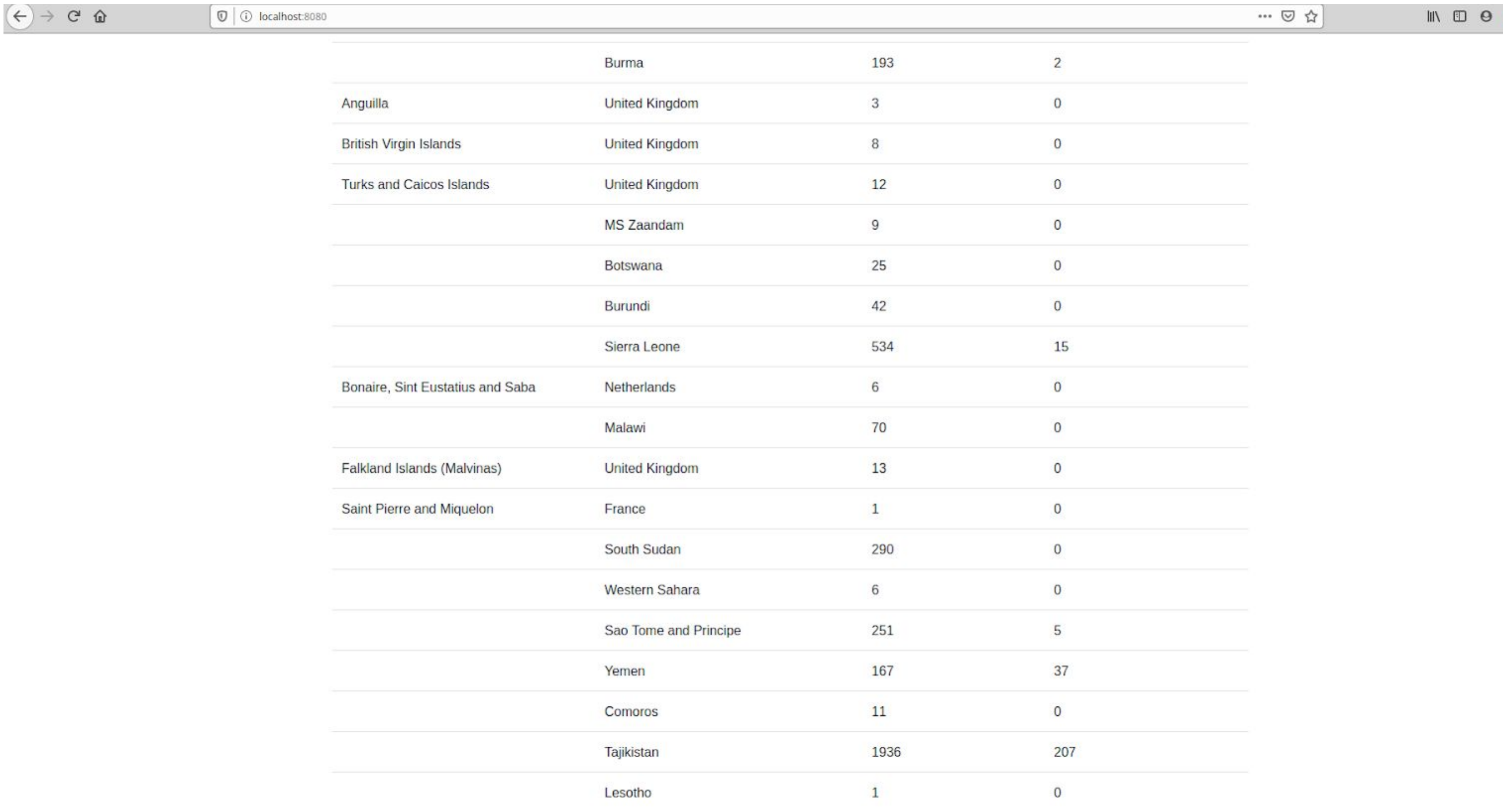
State	Country	Total cases reported	Changes since last day
	Afghanistan	7653	581
	Albania	949	1
	Algeria	7377	176
	Andorra	761	0
	Angola	52	2
	Antigua and Barbuda	25	0
	Argentina	8809	438
	Armenia	5041	218
Australian Capital Territory	Australia	107	0



Alberta	Canada	6716	33
British Columbia	Canada	2446	2
Grand Princess	Canada	13	0
Manitoba	Canada	290	0
New Brunswick	Canada	120	0
Newfoundland and Labrador	Canada	260	0
Nova Scotia	Canada	1044	1
Ontario	Canada	24755	469
Prince Edward Island	Canada	27	0
Quebec	Canada	44206	570
Saskatchewan	Canada	599	7
	Central African Republic	366	39
	Chad	545	26
	Chile	49579	3520
Anhui	China	991	0
Beijing	China	593	0
Chongqing	China	579	0
Fujian	China	356	0
Gansu	China	139	0
Guangdong	China	1590	0



	Georgia	707	6
	Germany	177778	1227
	Ghana	6096	361
	Greece	2840	4
	Guatemala	2133	221
	Guinea	2863	67
	Guyana	125	1
	Haiti	533	0
	Holy See	12	0
	Honduras	2955	157
	Hungary	3556	21
	Iceland	1802	0
	India	106475	6147
	Indonesia	18496	486
	Iran	124603	2111
	Iraq	3611	57
	Ireland	24251	51
	Israel	16659	16
	Italy	226699	813



	Burma	193	2
Anguilla	United Kingdom	3	0
British Virgin Islands	United Kingdom	8	0
Turks and Caicos Islands	United Kingdom	12	0
	MS Zaandam	9	0
	Botswana	25	0
	Burundi	42	0
	Sierra Leone	534	15
Bonaire, Sint Eustatius and Saba	Netherlands	6	0
	Malawi	70	0
Falkland Islands (Malvinas)	United Kingdom	13	0
Saint Pierre and Miquelon	France	1	0
	South Sudan	290	0
	Western Sahara	6	0
	Sao Tome and Principe	251	5
	Yemen	167	37
	Comoros	11	0
	Tajikistan	1936	207
	Lesotho	1	0