Objective – make a nice ui skin and display the parsed data from the given repository.

Using springboot

Dependencies—

* Thymeleaf(Templating engine)
* Spring web(because it is a web application)
* dev tools

import the project using the maven model

we would be taking data from CSSEGISanddata repository on github

CoronaVirusTrackerApplication – Main Method()

Create a new package services and add a class CoronaVirusDataServices

{this will take all the data from the repositories}

Create a string VIRUS\_DATA\_URL with the value of url of the data

**Create a method fetchVirusData inside** CoronaVirusDataServices

We make all the http call in java using http client and http request.

And convert the string into URI.

Then we are client.sending a request and deciding what happens with the body(here we are saying that just take the body and pass it as a string).

Then we are printing the body

Mark this as a service using @Service

Then we tell spring to execute this when the application starts using @postconstruct -this will show that when the construction of CoronaVirusDataServices is done call **fetchVirusData.**

We use java csv libraries to parse csv

for reading the data...csvBodyReader reader is created from the string instance

@Scheduled(cron = "\* \* 1 \* \* \*")//this will make the method run every day...

**Now we will create model to store data**

Its Class\_name will be LocationStats.

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 +  
 '}';  
 }  
}

we create an instance of LocationStats named all stats outside fetch data

and create a an instance of LocationStats named newStats inside fetchdata() so that as many people are accessing the data if at a given point of time a person requests data and the data is being constructed he will get the last updated data in that fraction of a second..

locationStat.setState(record.get("Province/State"));  
 locationStat.setCountry(record.get("Country/Region"));  
// locationStat.setLatestTotalCases(Integer.parseInt(record.get(record.size()-1)));//here since we have a string we convert it into an integer and since we have to find the last column we do size -1...

**Now we will create Controller to store data**

It is the way of accessing a page or a URL

public String home(Model model)

this will make the data available when we will be rendering the HTML

@Autowired//this is one of the feature that a service can be autowired to controller...  
CoronaVirusDataServices coronaVirusDataService;

We are autowiring this model so that we can use this(Services can be autowired)

**Now we will create home.html to display data**

<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>  
  
<body>  
<h1>CoronaVirus Tracker Application</h1>  
<p>This application shows the total number of case that a country and state has till date.</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>  
  
<div class="container">  
 <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}">Onions</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>

We calculate the total cases using linked list

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

to show change from last day

int latestCases = Integer.*parseInt*(record.get(record.size() - 1));  
int prevDayCases = Integer.*parseInt*(record.get(record.size()-2));  
locationStat.setLatestTotalCases(latestCases);  
locationStat.setDiffFromPrevDay(latestCases - prevDayCases);