SOLUTIONS FOR GUIDED LAB 3

Temperature.java

```
package com.temperaturereviewer.model;


public class Temperature {

    private int temperature;
    private String day;

    public Temperature() {}

    public Temperature(int t, String d) {

        temperature = t;
        day = d;
    }

    public int getTemperature() {
        return temperature;
    }
    public void setTemperature(int temperature) {
        this.temperature = temperature;
    }
    public String getDay() {
        return day;
    }
    public void setDay(String day) {
        this.day = day;
    }
}
```

TemperatureApplication.java

```
package com.temperaturereviewer.config;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cache.annotation.EnableCaching;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@EnableCaching
@ComponentScan({"com.temperaturereviewer.config,com.temperaturereviewer.controller,com.temperaturereviewer.model,
com.temperaturereviewer.service,com.temperaturereviewer.exception"}) public class TemperatureApplication {


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

TemperatureController.java

```
/**
 *
 */
package com.temperaturereviewer.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.temperaturereviewer.model.Temperature;
import com.temperaturereviewer.service.TemperatureService;


@RestController
@RequestMapping("/temperatures")
public class TemperatureController {

    @Autowired
    TemperatureService temperatureService;


    /**
     * @param day
     * @return Temperature
     */
    @GetMapping(value = "/daysTemp/{day}")
    public Temperature getTemperatureOfParticularDay(@PathVariable("day") int day)
    {
        return temperatureService.getTemperatureOfParticularDay(day);
    }

    /**
     * @return Temperature
     */
    @RequestMapping("/nextDay")
    public Temperature getTemperatureOfTheDay()
    {
        return temperatureService.getTemperatureOfTheDay();
    }

}
```

TemperatureService.java

```
/**
 *
 */
package com.temperaturereviewer.service;

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

import com.temperaturereviewer.model.Temperature;
import com.temperaturereviewer.model.TemperatureBO;

@Service
public class TemperatureService {
    @Autowired
    private TemperatureBO temps;


    /**
     * @return Temperature
     */
    public Temperature getTemperatureOfTheDay() {
        return temps.getTemperatureOfTheDay();
    }


    /**
     * @param day
     * @return Temperature
     */
    public Temperature getTemperatureOfParticularDay(int day) {

        return temps.getTemperatureOfParticularDay(day);
    }
}
```

TemperatureBO.java

```
package com.temperaturereviewer.model;

import java.util.ArrayList;
import java.util.List;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.cache.annotation.Cacheable;
import org.springframework.stereotype.Component;


@Component
public class TemperatureBO {

    private List<Temperature> temps = new ArrayList<Temperature>();
    @Value("${DayOfWeek}")
    private int nextDay;

    public List<Temperature> getTemps() {
        return temps;
    }
    public void setTemps(List<Temperature> temps) {
        this.temps = temps;
    }
    public int getNextDay() {
        return nextDay;
    }
    public void setNextDay(int nextDay) {
        this.nextDay = nextDay;
    }

    public TemperatureBO()
    {
        temps.add(new Temperature(31, "Monday"));
        temps.add(new Temperature(34, "Tuesday"));
        temps.add(new Temperature(30, "Wednesday"));
        temps.add(new Temperature(36, "Thursday"));
        temps.add(new Temperature(37, "Friday"));
        temps.add(new Temperature(38, "Saturday"));
        temps.add(new Temperature(33, "Sunday"));
    }

    /**
     * @return Temperature
     */
    public Temperature getTemperatureOfTheDay() {
        Temperature temperature = temps.get(nextDay);
        nextDay = (nextDay+1) % 7;
        return temperature;
    }

    /**
     * @param day
     * @return Temperature
     */
    @Cacheable("temps")
    public Temperature getTemperatureOfParticularDay(int day) {
        try {
            Thread.sleep(1500);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        day %=7;
        return temps.get(day);
    }
}
```

TemperatureTesting.java

```
package com.temperaturereviewer.service;

import static org.hamcrest.CoreMatchers.is;
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.jsonPath;
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.http.MediaType;
import org.springframework.test.context.junit4.SpringRunner;
import org.springframework.test.web.servlet.MockMvc;

import com.temperaturereviewer.config.TemperatureApplication;

@RunWith(SpringRunner.class)
@SpringBootTest(
        classes = TemperatureApplication.class)

@AutoConfigureMockMvc
public class TemperatureTesting {

    @Autowired
    private MockMvc mvc;

    /**
     * @throws Exception
     */
    @Test
    public void getTempForWednesday() throws Exception {
        mvc.perform(get("/temperatures/daysTemp/2")
                .contentType(MediaType.APPLICATION_JSON))
                .andExpect(status().isOk())
                .andExpect(jsonPath("$.temperature", is(33)));
    }

}
```