Displaying user Feedback

MavenWrapperDownloader.java

```
/*
 * Copyright 2007-present the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
import java.net.*;
import java.io.*;
import java.nio.channels.*;
import java.util.Properties;

public class MavenWrapperDownloader {

    private static final String WRAPPER_VERSION = "0.5.6";
    /**
     * Default URL to download the maven-wrapper.jar from, if no 'downloadUrl' is provided.
     */
```

```
    private static final String DEFAULT_DOWNLOAD_URL =
"https://repo.maven.apache.org/maven2/io/takari/maven-wrapper/"
        + WRAPPER_VERSION + "/maven-wrapper-" + WRAPPER_VERSION + ".jar";

    /**
     * Path to the maven-wrapper.properties file, which might contain a downloadUrl property
to
     * use instead of the default one.
     */
    private static final String MAVEN_WRAPPER_PROPERTIES_PATH =
            ".mvn/wrapper/maven-wrapper.properties";

    /**
     * Path where the maven-wrapper.jar will be saved to.
     */
    private static final String MAVEN_WRAPPER_JAR_PATH =
            ".mvn/wrapper/maven-wrapper.jar";

    /**
     * Name of the property which should be used to override the default download url for the
wrapper.
     */
    private static final String PROPERTY_NAME_WRAPPER_URL = "wrapperUrl";

    public static void main(String args[]) {
        System.out.println("- Downloader started");
        File baseDirectory = new File(args[0]);
        System.out.println("- Using base directory: " + baseDirectory.getAbsolutePath());

        // If the maven-wrapper.properties exists, read it and check if it contains a custom
        // wrapperUrl parameter.
```

```
        File mavenWrapperPropertyFile = new File(baseDirectory,
MAVEN_WRAPPER_PROPERTIES_PATH);
        String url = DEFAULT_DOWNLOAD_URL;
        if(mavenWrapperPropertyFile.exists()) {
            FileInputStream mavenWrapperPropertyFileInputStream = null;
            try {
                mavenWrapperPropertyFileInputStream = new
FileInputStream(mavenWrapperPropertyFile);
                Properties mavenWrapperProperties = new Properties();
                mavenWrapperProperties.load(mavenWrapperPropertyFileInputStream);
                url =
mavenWrapperProperties.getProperty(PROPERTY_NAME_WRAPPER_URL, url);
            } catch (IOException e) {
                System.out.println("- ERROR loading '" +
MAVEN_WRAPPER_PROPERTIES_PATH + "'");
            } finally {
                try {
                    if(mavenWrapperPropertyFileInputStream != null) {
                        mavenWrapperPropertyFileInputStream.close();
                    }
                } catch (IOException e) {
                    // Ignore ...
                }
            }
        }
        System.out.println("- Downloading from: " + url);

        File outputFile = new File(baseDirectory.getAbsolutePath(),
MAVEN_WRAPPER_JAR_PATH);
        if(!outputFile.getParentFile().exists()) {
            if(!outputFile.getParentFile().mkdirs()) {
                System.out.println(
```

```
                    "- ERROR creating output directory '" +
outputFile.getParentFile().getAbsolutePath() + "'");
            }
        }
        System.out.println("- Downloading to: " + outputFile.getAbsolutePath());
        try {
            downloadFileFromURL(url, outputFile);
            System.out.println("Done");
            System.exit(0);
        } catch (Throwable e) {
            System.out.println("- Error downloading");
            e.printStackTrace();
            System.exit(1);
        }
    }

    private static void downloadFileFromURL(String urlString, File destination) throws
Exception {
        if (System.getenv("MVNW_USERNAME") != null &&
System.getenv("MVNW_PASSWORD") != null) {
            String username = System.getenv("MVNW_USERNAME");
            char[] password = System.getenv("MVNW_PASSWORD").toCharArray();
            Authenticator.setDefault(new Authenticator() {
                @Override
                protected PasswordAuthentication getPasswordAuthentication() {
                    return new PasswordAuthentication(username, password);
                }
            });
        }
        URL website = new URL(urlString);
        ReadableByteChannel rbc;
```

```
        rbc = Channels.newChannel(website.openStream());
        FileOutputStream fos = new FileOutputStream(destination);
        fos.getChannel().transferFrom(rbc, 0, Long.MAX_VALUE);
        fos.close();
        rbc.close();
    }

}
```

Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <parent>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-parent</artifactId>
                <version>2.3.1.RELEASE</version>
                <relativePath/> <!-- lookup parent from repository -->
        </parent>
        <groupId>com.example</groupId>
        <artifactId>springcore</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <name>springcore</name>
        <description>Demo project for Spring Boot</description>

        <properties>
                <java.version>1.8</java.version>
        </properties>
```

```
        <dependencies>
                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-data-jpa</artifactId>
                </dependency>
                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-web</artifactId>
                </dependency>

<dependency>
                        <groupId>org.apache.tomcat.embed</groupId>
                        <artifactId>tomcat-embed-jasper</artifactId>
                </dependency>

                <dependency>
   <groupId>mysql</groupId>
   <artifactId>mysql-connector-java</artifactId>
 </dependency>

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

        <build>
            <plugins>
                <plugin>
                    <groupId>org.springframework.boot</groupId>
                    <artifactId>spring-boot-maven-plugin</artifactId>
                </plugin>
            </plugins>
        </build>

</project>
```

SpringcoreApplicationTests.java

```
package com.example.springcore;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class SpringcoreApplicationTests {
```

```
	@Test
	void contextLoads() {
	}

}
```

FeedbackController.java

```
package com.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

import com.service.AppService;

@Controller
public class FeedbackController {

	@Autowired
	private AppService appService;

	@GetMapping("/feedback")
	public String feedback() {
		return "feedback";
	}
}
```

MyRestApp.java

```
package com.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import com.entity.Feedback;
import com.service.AppService;

@RestController
public class MyRestApp {

	@Autowired
	private AppService service;

	@PostMapping("/feedback")
	public String userRegister(@RequestParam("firstname") String firstname, @RequestParam("lastname") String lastname, @RequestParam("email") String email , @RequestParam("feedback1") String feedback1) {
	Feedback f = new Feedback(firstname, lastname, email,feedback1);
	boolean data= service.addFeedback(f);
	if(data) {
		return "Feedback added succesfully!";
	}
	else {
		return "Unable to add the feedback";
	}
```

```
	}
}
```

MyRepo.java

```
package com.dao;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import com.entity.Feedback;

@Repository
public interface MyRepo extends CrudRepository<Feedback, Integer>{

}
```

Feedback.java

```
package com.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity @Table(name="feedback")
public class Feedback {
	@Override
```

```
	public String toString() {
		return "Feedback [id=" + id + ", firstname=" + firstname + ", lastname=" +
lastname + ", email=" + email
				+ ", feedback1=" + feedback1 + "]";
	}

	@Id @GeneratedValue
	private int id;
	private String firstname;
	private String lastname;
	private String email;
	private String feedback1;

	public Feedback() {
		super();
		// TODO Auto-generated constructor stub
	}

	public Feedback(String firstname, String lastname, String email ,String feedback1) {
		super();
		this.firstname = firstname;
		this.lastname = lastname;
		this.email= email;
		this.feedback1 = feedback1;
	}


}
```

SpringcoreApplication.java

```
package com.example.springcore;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.domain.EntityScan;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;
import org.springframework.web.bind.annotation.GetMapping;

@SpringBootApplication
@ComponentScan(basePackages="com")

@EntityScan(basePackages="com")
@EnableJpaRepositories(basePackages="com")
public class SpringcoreApplication {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		SpringApplication.run(SpringcoreApplication.class, args);

	}

}
```

AppService.java

```
package com.service;

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

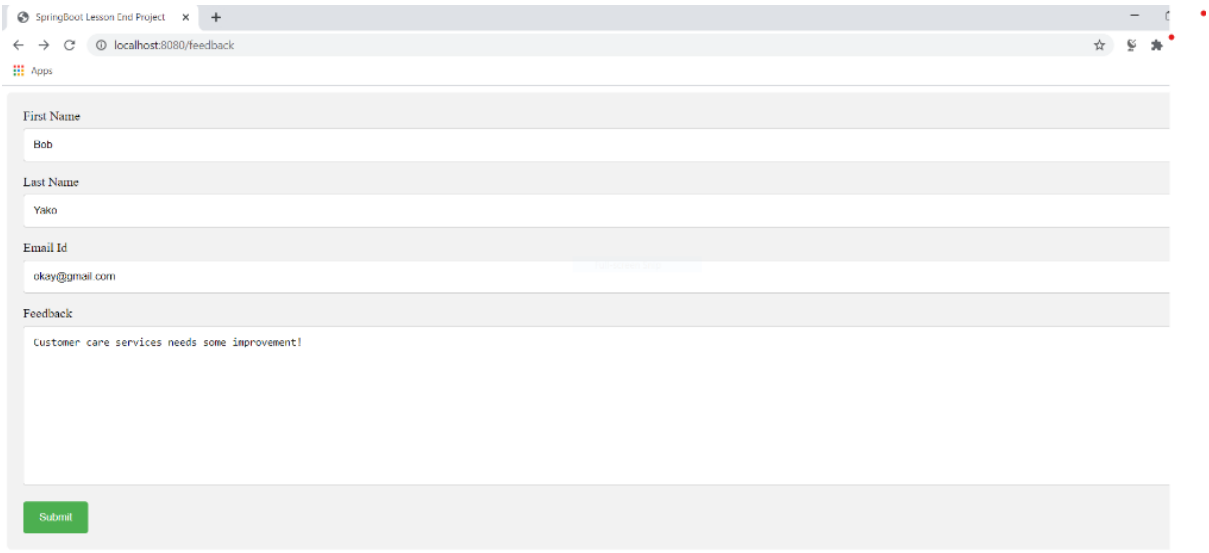
```
import com.dao.MyRepo;
import com.entity.Feedback;

@Service
public class AppService {

	@Autowired
	private MyRepo myRepo;

	public boolean addFeedback( Feedback f) {
		myRepo.save(f);
		return true;
	}
}
```

Output Screenshot




```
Hibernate:
    insert
    into
        feedback
        (email, feedback1, firstname, lastname)
    values
        (?, ?, ?, ?)
Hibernate:
    insert
    into
        feedback
        (email, feedback1, firstname, lastname)
    values
        (?, ?, ?, ?)
```

```
ysql> select * from feedback;
+----+---------------+-----------------------------------------+-----------+----------+
| id | email         | feedback1                               | firstname | lastname |
+----+---------------+-----------------------------------------+-----------+----------+
|  1 | abc@gmail.com | Delivery services could be improved a lot.     | Priyansh  | Shah     |
|  2 | okay@gmail.com | Customer care services needs some improvement! | Bob       | Yako     |
+----+---------------+-----------------------------------------+-----------+----------+
 rows in set (0.00 sec)

ysql>
```