Lab 6.1

Posting data to a service

- **Add the following method to the interface StudentDao**

```
void add(Student student)
```

- **Implement this method in the class StudentDaoImpl**

```
@Override
public void add(Student student) {
        long id = students.keySet().stream().count();
        id++;
        student.setId(id);
        students.put(id, student);
}
```

Lab 6.1

Posting data to a service

- **Add the following method to the interface StudentService**

```
void add(Student student)
```

- **Implement this method in the class StudentServiceImpl**

```
@Override
public void add(Student student) {
	if(student.getFirstName() != null && student.getSurname() != null
			&& student.getDept() != null) {
		studentDao.add(student);
	}
}
```

Lab 6.1

Posting data to a service

- **We will now create an endpoint tied to an Http POST**
- **In the class StudentController create a new method as below;**

```
public ResponseEntity<String> addStudent(Student student) {
    studentService.add(student);
    if(student.getId() > 0) {
        URI add Student = URI.create("/college/student/" + student.getId());
        return ResponseEntity.accepted().location(uri).build();
    } else {
        return ResponseEntity.badRequest().build();
    }
}
```

- **As you can see, we account for both a positive and negative response**
- **Annotate the method with @PostMapping, tied to the url of the class itself (no concatentation) and the methods student argument with @RequestBody**