011 Section 011 Controller Annotations

Spring Boot Annotations and Their Usage

1. @RestController

Usage: This annotation is a combination of `@Controller` and `@ResponseBody`. It is used to create RESTful web services in Spring Boot.

Example:

```
@RestController
@RequestMapping("/journal")
public class JournalController {
    // Controller methods
}
```

Explanation: `@RestController` tells Spring Boot that this class handles HTTP requests and returns data directly in JSON format.

---

2. @RequestMapping

Usage: This annotation is used to map web requests onto specific handler classes or methods.

Example:

```
@RequestMapping("/journal")
```

Explanation: All endpoints inside `JournalController` will be prefixed with `/journal`.

---

3. @GetMapping

Usage: This annotation is used to handle HTTP GET requests.

Example:

```
@GetMapping
public List<JournalEntries> getAll(){
    return new ArrayList<>(journalEntries.values());
}
```

Explanation: When a GET request is made to `/journal`, this method returns a list of all journal entries.

---

4. @PostMapping

Usage: This annotation is used to handle HTTP POST requests.

Example:

```
@PostMapping
public boolean createEntry(@RequestBody JournalEntries journalEntries){
    this.journalEntries.put(journalEntries.getId(), journalEntries);
    return true;
}
```

Explanation: When a POST request is made to `/journal`, it adds a new journal entry to the `journalEntries` map.

---

5. @RequestBody

Usage: This annotation is used to convert the JSON body of a request into a Java object.

Example:

```
public boolean createEntry(@RequestBody JournalEntries journalEntries)
```

Explanation: It takes the incoming JSON request body and converts it into a `JournalEntries` object.

---

6. @GetMapping("/id/{myid}")

Usage: This annotation maps the GET request with a path variable.

Example:

```
@GetMapping("/id/{myid}")
public JournalEntries getJournalById(@PathVariable long myid){
    return journalEntries.get(myid);
}
```

Explanation: This method retrieves a journal entry by its ID. The `{myid}` in the URL is replaced by the actual ID provided by the user.

---

7. @PathVariable

Usage: This annotation binds a method parameter to a URI variable.

Example:

```
public JournalEntries getJournalById(@PathVariable long myid)
```

Explanation: The `myid` parameter is automatically assigned the value from the URL.

---

8. @PutMapping

Usage: This annotation handles HTTP PUT requests used for updating resources.

Example:

```
@PutMapping
public JournalEntries updateJournalById(@PathVariable long id,@RequestBody JournalEntries myEntry){
    return journalEntries.put(id, myEntry);
}
```

Explanation: It updates the existing journal entry with the given `id` using data from the request body.

Issue in Code: `@PathVariable long id` is missing from the method URL path. It should be:

```
@PutMapping("/id/{id}")
public JournalEntries updateJournalById(@PathVariable long id, @RequestBody JournalEntries myEntry){
    return journalEntries.put(id, myEntry);
}
```

---

9. @DeleteMapping

Usage: This annotation handles HTTP DELETE requests.

Example:

```
@DeleteMapping("/id/{id}")
public boolean deleteJournal(@PathVariable long id){
    journalEntries.remove(id);
    return true;
}
```

Explanation: When a DELETE request is made to `/journal/id/{id}`, it removes the journal entry with the given ID.

Issue in Code: The mapping is incorrect: `@DeleteMapping("/id{id}")` should be `@DeleteMapping("/id/{id}")`.

Version of `JournalController`

```
@RestController
@RequestMapping("/journal")
public class JournalController {

    private Map<Long, JournalEntries> journalEntries = new HashMap<>();

    @GetMapping
    public List<JournalEntries> getAll(){
        return new ArrayList<>(journalEntries.values());
    }

    @PostMapping
    public boolean createEntry(@RequestBody JournalEntries journalEntries){
        this.journalEntries.put(journalEntries.getId(), journalEntries);
        return true;
    }

    @GetMapping("/id/{myid}")
    public JournalEntries getJournalById(@PathVariable long myid){
        return journalEntries.get(myid);
    }

    @PutMapping("/id/{id}")
    public JournalEntries updateJournalById(@PathVariable long id, @RequestBody JournalEntries myEntry){
        return journalEntries.put(id, myEntry);
    }

    @DeleteMapping("/id/{id}")
    public boolean deleteJournal(@PathVariable long id){
        journalEntries.remove(id);
        return true;
    }
}
```