Annotations

1. **@RestController**

- Combines @Controller + @ResponseBody.
- Tells Spring to treat the class as a web controller where **every method returns a response body**, typically JSON.

---

2. **@Service**

- Marks a class as a **service component**.
- Used to define business logic. Spring creates an object of this class and manages it as a **Spring Bean**.

---

3. **@Controller**

- Marks the class as a **web controller**.
- Used in MVC architecture. Methods typically return **view names**.

---

4. **@Repository**

- Marks the class as a **DAO (Data Access Object)** layer component.
- Used to handle database operations.
- Spring provides exception translation (e.g., SQL exceptions into Spring exceptions).

5. `@Autowired`

- Automatically **injects** the required dependency (like a service or repository) into a class.

6. `@RequestBody`

- Binds the **HTTP request body** to a method parameter in a controller.
- Mostly used with POST/PUT to accept JSON input and map it to a POJO.

7. `@RequestMapping`

- Maps **HTTP requests** to handler methods.
- Can be used at the class level (base URL) or method level with method type (GET, POST, etc.).

8. `@GetMapping`

- Shortcut for `@RequestMapping(method = RequestMethod.GET)`
- Maps a GET request.

9. `@PutMapping`

- Shortcut for `@RequestMapping(method = RequestMethod.PUT)`
- Maps a PUT request (used for updates).

---

10. `@PostMapping`

- Shortcut for `@RequestMapping(method = RequestMethod.POST)`
- Maps a POST request (used for creating data).

---

11. `@DeleteMapping`

- Shortcut for `@RequestMapping(method = RequestMethod.DELETE)`
- Maps a DELETE request.