CONTROLLERS:

1. Can be either @Controller or @RestController
2. For a REST application , one can either use
   1. @Controller and @ResponseBody
   2. @RestController(combination of 2 annotations above)
3. @ResponseBody is used to serialise the response object into HttpResponse
   1. Eg:

| @Controller  @RequestMapping("books")  public class **SimpleBookController** {  @GetMapping("/{id}", produces = "application/json")  public @ResponseBody Book **getBook**(@PathVariable **int** id) {  return findBookById(id);  } |
| --- |

1. @RequestMapping is used to define the common servlet path to that particular controller and is a class level annotation
   1. Eg:

| @RestController  @RequestMapping("books-rest")  public class **SimpleBookRestController** {    @GetMapping("/{id}", produces = "application/json")  public Book **getBook**(@PathVariable **int** id) {  return findBookById(id);  } |
| --- |

1. @GetMapping,@PostMapping,@PutMapping,@DeleteMapping are annotated at method level to map the incoming request with the appropriate method
2. They can also mention the servlet path
   1. Eg: @GetMapping(“/{id}”);
3. @RequestBody is an annotation which validates the incoming request body is similar to the expected request body
4. @PathVariable is used to get the path variable from the request URL
   1. Eg:

| @GetMapping("/area/{city}")  **public** List<Area> getAllAreasByCity(@PathVariable(value="city") String city){  **return** areaRepo.findByCity(city);  } |
| --- |

1. Both @RequestBody and @PathVariabel can be made optional by setting ‘required’ attribute as ‘false’
2. @Autowired is used to inject the dependency during runtime by spring application context
3. General/Good rules:
   1. Separate POST,GET,PUT,DELETE requests
   2. The RequestBody should be a DTO and not an entity
   3. Use a mapper to convert the DTO to entity
   4. GET should return list of objects/ 1 entity with response code
   5. DELETE should return a custom response body stating the status code,timestamp,successful,message
   6. Same for update/PUT request(sometimes send the updated entity)
   7. POST should return status code with the posted entity or custom response
4. Don’t throw errors in controllers