Java Interview Questions

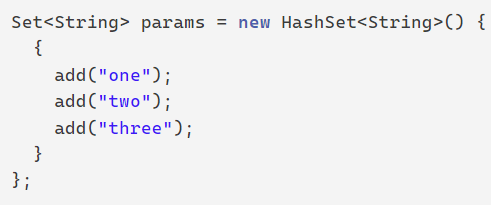
Table of Contents

[Java Specific 1](#_Toc131837330)

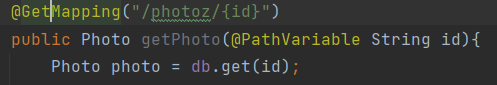
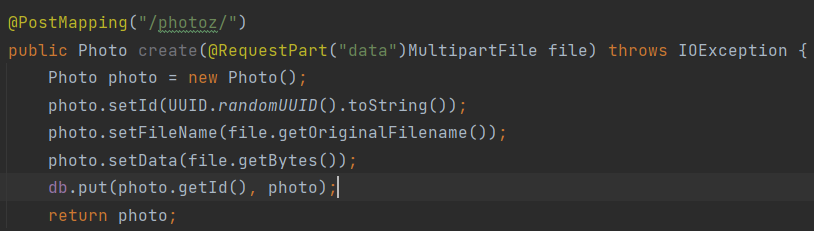
[Spring Boot 1](#_Toc131837331)

[Designs 3](#_Toc131837332)

# Java Specific

1. What are double curly braces for?
   1. Java double brace initialization is defined as creating and initializing the objects in a single step, normally done in multiple steps. They are done using double curly braces.
   2. Ex:  
      
   3. [Source](https://howtodoinjava.com/java/double-brace-initialization-in-java/)

# Spring Boot

1. Annotations:
   1. @RestController
      1. Creates RESTful service using Spring MVC
      2. Used on class
      3. [Source](https://www.digitalocean.com/community/tutorials/spring-restcontroller)
   2. Http Methods
      1. @GetMapping(“/{arg}”), @PostMapping(), etc
      2. Shortcut for @RequestMapping(method=RequestMethod.GET)
      3. Alias for @RequestMapping(value = "/{arg}", method = RequestMethod.GET)
      4. [Source](https://www.javaguides.net/2018/11/spring-getmapping-postmapping-putmapping-deletemapping-patchmapping.html)
      5. POST = Create, GET = Read, PUT = Update, DELETE = Delete == CRUD
   3. @PathVariable
      1. Allows Spring Boot to extract value of a variable in the request mapping and binds/passes/assigns it to the similarly named instance variable.
      2. Ex:   
         
   4. @RequestBody
      1. Deserializes then maps body of inbound HttpRequest object to Java Object
      2. Type we annotate with the @RequestBody annotation must correspond to the JSON sent from our client-side controller.
   5. @ResponseBody
      1. Tells the controller that the object being returned is to be serialized into JSON and passed back to HttpResponse object.
      2. [Source](https://www.baeldung.com/spring-request-response-body)
   6. @Valid, @NotNull, @NotEmpty
      1. Part of spring-boot-starter-validation dependency
      2. @Valid ensures the validation of the **whole** object
      3. @NotNull indicates a method shouldn’t return null, however it does not enforce it on its own. Can use org.hibernate.validator to enforce it, but is tedious.
         1. Instead, can use spring’s validation @Validated to a class, @NotNull to parameter
         2. [Source](https://www.baeldung.com/java-notnull-method-parameter)
      4. @NotEmpty requires a field/element to be nonempty. Otherwise returns error (404 in http; server still runs though).
      5. [Source 1](https://www.baeldung.com/spring-valid-vs-validated#:~:text=The%20%40Valid%20annotation%20ensures%20the,including%20the%20above%20partial%20validation.)
   7. @JsonIgnore
      1. Ignores a field/attribute
      2. [Source](https://www.tutorialspoint.com/jackson_annotations/jackson_annotations_jsonignore.htm)
   8. @Component or @Service
      1. See section 5.b
2. What is the difference between @RequestMapping and @GetMapping?
   1. @RequestMapping is used at the class level while @GetMapping is used to connect the methods.
3. MultipartFile
   1. Spring specific class used for mapping uploaded files
   2. To use, need the annotation @RequestPart(“<key>”) for the parameter where “key” is the key for the file in the json.
   3. Ex: Note that “data” corresponds to the key and the value will be our file. This is done in html and js.  
      

# Designs

1. Spring Application
   1. Controllers are the @RestController classes. These are meant for http methods like CRUD. i.e. they convert json to java objects and vice versa.
   2. Services are the @Component or @Service and are meant to interact with databases.
      1. Typically you inject these dependencies.
      2. Ex: in the controller class, you create a private field of the service class. Then pass this class in the constructor. Spring boot will scan, see the @Service, create a singleton of this class, then provide it to the constructor.
         1. @Autowired applied to the field takes care of this automatically and a constructor won’t be needed.
         2. Notice that the field is final and the class has @Service/@Component annotated. 
         3. The point is that all the controllers can use the same service object.