Spring overview

* @Component: mark a class as a **spring bean**
* @Bean: make an external class (such as when using class from AWS) to a **spring bean**
* @AutoWired: use for dependency injection, can be placed in constructor or setter
* @Qualifier: when having multiple bean that inherited from a super bean, use this to assign a specified bean that will use for dependency injection
* @Primary: the same context from above, but assign a bean as default when using dependency injection
* @Lazy: just create singleton object when need
* @PostConstruct, @PreDestroy: execute after constructor or before destructor

JPA

* @Entity: a class that map to a table in database
* @Transactional: use this when a function can modify database (C, U, D)
* @Repository: mark a class as DAO (or this class will become DAO implementation)

REST API

* @GetMapping, @PostMapping, @PutMapping, @DeleteMapping: define a route
* @RequestMapping: catch all HTTP method
* @RequestBody: take the data from request body
* @PathVariable: take parameter from dynamic route
* @ControllerAdvice: an interceptor/filter (pre-process request, post-process response) -> perfect for global exception handling
* @Service: Spring will automatically register the Service implementation, used for Service layer (service contains many DAO to communicate with DB)

REST security

* @Configuration: mark a class that this is a configuration class – class contains all config about security/management - > everything relevant to security coded in here

MVC

* @RequestParam: read the parameter from request – define in function parameters
* Validation

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* @InitBinder: preprocess each web request before going to controller

Note:

* **spring bean:** a class that managed by spring. Spring will scan bean class, create a instance (singleton) and manage it in IoC container. When need dependency injection, spring will point to this singleton object