

SPRING BOOT

ANNOTATIONS

Annotation	Détails
@SpringBootApplication	Principale annotation de l'application de démarrage de Spring Boot. Utilisé une seule fois dans l'application, contient une méthode principale et sert de paquet principal pour le scan des packages
@SpringBootConfiguration	Indique qu'une classe fournit une application Spring Boot. Ne devrait être déclaré qu'une seule fois dans un projet, habituellement automatiquement en utilisant @SpringBootApplication
@ComponentScan	Utilisé pour déclencher le scan d'un package automatiquement sur un certain paquet et ses enfants ou pour configurer le scan de package personnalisé
@Configuration	Utilisé pour déclarer une ou plusieurs méthodes @Bean. Peut être sélectionné par balayage automatique de paquets afin de déclarer une ou plusieurs méthodes @Bean au lieu de la configuration xml traditionnelle.
@Bean	Indique qu'une méthode produit un Bean à gérer par le conteneur Spring. Habituellement, les méthodes annotées @Bean seront placées dans les classes annotées @Configuration qui seront sélectionnées par balayage de paquetage pour créer des beans basés sur la configuration java.
@Component	En déclarant une classe en tant que @Component, elle devient candidate à la détection automatique lors de la configuration basée sur l'annotation et la recherche dans le classpath. Habituellement, une classe annotée avec @Component deviendra un Bean dans l'application.
@Repository	Initialement défini par Domain-Driven Design comme "un mécanisme pour encapsuler le stockage. Il est habituellement utilisé pour indiquer un référentiel pour les données de Spring.
@Service	Très similaire en pratique à @Component. Initialement défini par Domain-Driven Design comme "une opération offerte en tant qu'interface qui se trouve seule dans le modèle, sans état encapsulé".
@Controller	Indique qu'une classe annotée est un "Contrôleur" (par exemple, un contrôleur Web)
@RestController	Une annotation de commodité qui est elle-même

Annotation	Détails
	annotée avec @Controller et @ResponseBody. Sera automatiquement sélectionné par défaut car elle contient l'annotation @Controller sélectionnée par défaut.
@ResponseBody	L'annotation signifie que la valeur retournée de la méthode constituera le corps de la réponse HTTP.
@RequestMapping	L'URI est exprimée par l'attribut value<. La méthode d'appel est définie dans le paramètre method. Ce dernier peut prendre soit une seule valeur, soit plusieurs valeurs
@RequestParam	L'annotation @RequestParam est utilisée pour accéder aux valeurs des paramètres de la requête.
@PathVariable	@PathVariable identifie le modèle qui est utilisé dans l'URI pour la demande entrante.
@Value	Permet d'accéder et mapper les valeurs d'un fichier de configuration properties avec une variable de votre application.
@Autowired	L'annotation @Autowired vous permet de sauter des configurations ailleurs de ce qu'il faut injecter et le fait simplement pour vous.
@ConfigurationProperties (prefix = "pref")	permet de charger toutes les propriétés avec ce préfixe dans cette classe
@ConditionalOnExpression	Annotation de configuration pour un élément conditionnel qui dépend de la valeur d'une expression de SpEL.
@Id	Démarque un identifiant.
@EnableWebMvc	Cette annotation demande à Spring Boot de configurer une application Web de type MVC.
@ConfigurationProperties ("hello")	Réalise le mapping de valeur d'un fichier de configuration avec un objet de votre application, on donnera le préfixe des valeurs à mapper.
@GeneratedValue	Cette annotation indique que la clé primaire est générée de façon automatique lors de l'insertion en base. Sans cette annotation, la valeur de l'identifiant de la clé primaire doit être affecté avant l'insertion en base.
@RequestBody	L'annotation indiquant un paramètre de méthode doit être liée au corps de la requête Web. Le corps de la requête est passé par un HttpMessageConverter pour résoudre l'argument de la méthode en fonction du type de contenu de la requête.
@ServerEndpoint	Une classe annotée par @ServerEndpoint est enregistrée sur un point d'extrémité WebSocket côté serveur et chaque fois que le WebSocket du point final correspondant est connecté au serveur.