In case of **@RestController** the parameter value depicts the component name or bean name, whereas in **@RequestMapping** the value parameter is used to specify the path.

The **@RequestMapping** annotation is used to map requests to controllers methods.

The **@CrossOrigin** annotation is a savior in such scenarios. At its core, it provides a way to overcome the same-origin policy applied by web browsers.

In Spring Boot, **@Autowired** is an annotation used for dependency injection.

The **@PostMapping** annotation is a Spring annotation that is used to map HTTP POST requests onto specific handler methods.

In Spring Boot, the **@RequestBody** annotation is used to bind the HTTP request body to a method parameter in a controller handler method.

In Spring Boot, you can use the **@GeneratedValue** annotation with the **@Id** annotation to automatically generate primary key values for your entities.

In Spring Boot, a **@repository** is an interface that provides a way to interact with a database or other data store.

In Spring Boot, a **service** is a class that holds business logic. It acts as a layer between the controller (which handles HTTP requests) and the repository (which interacts with the database).

In Spring Boot, a **bean** is a Java object that is managed by the Spring IoC (Inversion of Control) container.

ModelMapper is a Java library that helps in mapping objects from one model to another, reducing the need for manual mapping code.

Spring Boot favors Java-based **configuration**. Although it is possible to use SpringApplication with XML sources, we generally recommend that your primary source be a single @Configuration class.

JpaRepository is particularly a JPA specific extension for Repository. It has full API CrudRepository and PagingAndSortingRepository.

In Spring Boot, the **@SpringBootApplication** annotation is a powerful tool that simplifies the bootstrapping and launching of your Spring applications.