**Аннотации MVC**

**1. @SpringBootApplication** – Аннотация @SpringBootApplication включает сканирование компонентов и автоматическую конфигурацию приложения. На самом деле, за этой аннотацией скрывается целых три аннотация с настройками по умолчанию:

**@Configuration** (Spring) – помечает класс как конфигурационный (в контексте java-based конфигурации).

**@ComponentScan** (Spring) – включает сканирование компонентов, то есть классы-контроллеры и другие компоненты, которые вы создаете, будут автоматически зарегистрированы как бины в application context.

**@EnableAutoConfiguration** (Spring Boot) – самая магическая аннотация из всех возможных. Именно она под капотом запускает весь механизм автоматической конфигурации Spring Boot. Это означает, что Spring Boot ищет bean-компоненты автоконфигурации в своем пути к классам и автоматически применяет их.

**2. @RequestMapping** – используется для отображения запросов на методы контроллеров. Она имеет различные атрибуты для сопоставления по URL-адресу, HTTP-методу, параметрам запроса, заголовкам и типам данных. Вы можете использовать её на уровне класса для выражения общих отображений или на уровне метода для сужения до конкретного отображения конечной точки.

Проще говоря, аннотация используется для сопоставления веб-запросов с методами Spring Controller.

Существуют также специфические для HTTP-метода варианты сокращения аннотации:

@RequestMapping:

@GetMapping

@PostMapping  
@PutMapping  
@DeleteMapping  
@PatchMapping

Сокращения являются кастомными аннотациями, которые указываются потому, что, пожалуй, большинство методов контроллера нужно сопоставить с конкретным HTTP-методом, а не использовать аннотацию @RequestMapping, которая по умолчанию производит сопоставление со всеми HTTP-методами. Для выражения общих отображений на уровне класса по-прежнему необходима @RequestMapping.

Пример:

