### **Задание 1: Создание простого контроллера**

**Цель:** Научиться создавать контроллеры и обрабатывать запросы в Spring MVC.

**Задание:**

1. Создайте новый проект Spring Boot.
2. Создайте контроллер с маппингом на путь GET /hello.
3. Реализуйте метод, который возвращает строку "Hello, World!" при GET-запросе на путь /hello.

### **Задание 2: Обработка POST-запросов**

**Цель:** Научиться обрабатывать POST-запросы и работать с телом запроса.

**Задание:**

1. Создайте контроллер с маппингом на путь /greet.
2. Реализуйте метод, который принимает JSON-объект с полем name и возвращает приветственное сообщение, используя значение этого поля.

### **Задание 3: Валидация данных**

**Цель:** Научиться валидировать данные, поступающие в контроллер.

**Задание:**

1. Создайте класс User с полями name и email.
2. Используйте аннотации для валидации полей на непустое содержимое, длину и соответствие формату email.
3. Реализуйте метод контроллера, который принимает объект User и возвращает сообщение об успешной регистрации.

### **Задание 4: Использование @RequestParam и @PathVariable**

**Цель:** Научиться использовать параметры запроса и переменные пути в контроллерах.

**Задание:**

1. Создайте метод контроллера, который принимает параметр id из запроса и переменную name из пути.
2. Метод должен возвращать строку, объединяющую значения этих параметров.

### **Задание 5: Обработка исключений с помощью @ExceptionHandler**

**Цель:** Научиться обрабатывать исключения в контроллерах.

**Задание:**

1. Создайте контроллер с методом, который выбрасывает исключение UserNotFoundException при определенных условиях.
2. Создайте метод с аннотацией @ExceptionHandler, который обрабатывает UserNotFoundException и возвращает сообщение об ошибке.

### **Задание 6: Использование @RequestHeader**

**Цель:** Научиться работать с заголовками HTTP-запросов.

**Задание:**

1. Создайте метод контроллера, который принимает заголовок User-Agent.
2. Метод должен возвращать строку, содержащую значение этого заголовка.

### **Задание 7: Использование интерцепторов**

**Цель:** Научиться перехватывать и обрабатывать запросы до и после их выполнения контроллером.

**Задание:**

1. Создайте класс, реализующий интерфейс HandlerInterceptor.
2. Реализуйте методы preHandle, postHandle, и afterCompletion.
3. Зарегистрируйте интерцептор в конфигурации Spring.

### **Задание 8: Работа с @ResponseStatus**

**Цель:** Научиться устанавливать HTTP-статус ответа.

**Задание:**

1. Создайте метод контроллера, который выбрасывает исключение CustomException.
2. Создайте класс исключения CustomException и пометьте его аннотацией @ResponseStatus с кодом 404.
3. Создайте метод с аннотацией @ExceptionHandler, который обрабатывает CustomException и возвращает сообщение об ошибке.

### **Задание 9: Использование @PathVariable с регулярными выражениями**

**Цель:** Научиться использовать регулярные выражения в @PathVariable.

**Задание:**

1. Создайте метод контроллера, который принимает переменную пути date в формате yyyy-MM-dd.
2. Метод должен возвращать сообщение с датой, если формат верный, и сообщение об ошибке, если формат неверный.

### **Задание 10: Использование @RequestParam с параметрами по умолчанию**

**Цель:** Научиться работать с параметрами запроса и устанавливать значения по умолчанию.

**Задание:**

1. Создайте метод контроллера, который принимает параметр запроса name с значением по умолчанию "Guest".
2. Метод должен возвращать приветственное сообщение с именем пользователя.