**. Create a Custom Annotation for Validation**

First, create a custom annotation to apply the validation logic:

java

Copy code

import javax.validation.Constraint;

import javax.validation.Payload;

import java.lang.annotation.ElementType;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

import java.lang.annotation.Target;

@Constraint(validatedBy = TestListValidator.class)

@Target({ ElementType.FIELD })

@Retention(RetentionPolicy.RUNTIME)

public @interface ValidTestList {

String message() default "Invalid entry in list: each value must be numeric, 5 digits long, and non-null";

Class<?>[] groups() default {};

Class<? extends Payload>[] payload() default {};

}

**2. Create the Validator Class**

Next, implement the ConstraintValidator to define the validation logic:

java

Copy code

import javax.validation.ConstraintValidator;

import javax.validation.ConstraintValidatorContext;

import java.util.List;

public class TestListValidator implements ConstraintValidator<ValidTestList, List<String>> {

@Override

public void initialize(ValidTestList constraintAnnotation) {

}

@Override

public boolean isValid(List<String> testList, ConstraintValidatorContext context) {

if (testList == null) {

return true; // Consider null list as valid; change if you want to enforce non-null.

}

for (String value : testList) {

if (value == null || value.length() != 5 || !value.matches("\\d+")) {

return false;

}

}

return true;

}

}

**3. Apply the Custom Annotation to Your Field**

Use the custom @ValidTestList annotation on the List<String> field in your model:

java

Copy code

import javax.validation.constraints.NotNull;

import java.util.List;

public class YourModel {

@NotNull(message = "Test list cannot be null")

@ValidTestList

private List<String> test;

// Getters and setters

}

**4. Controller and Validation Handling**

Ensure you use @Valid or @Validated in your controller method to trigger validation:

java

Copy code

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import javax.validation.Valid;

@RestController

public class TestController {

@PostMapping("/test")

public String validateTest(@Valid @RequestBody YourModel model) {

return "Validation passed!";

}

}

**Explanation:**

* The @ValidTestList annotation is used to mark the List<String> field that requires validation.
* The TestListValidator checks if each string in the list is non-null, exactly 5 digits long, and numeric.
* The @NotNull annotation ensures the list itself is not null.

This approach ensures that each string in the test list is validated according to your requirements