



Como validar parâmetros de solicitação no Spring Boot

No Spring Boot MVC, o url contém parâmetros de consulta. Esses parâmetros de consulta são identificados como parâmetros de solicitação. O URL da solicitação contém parâmetros de consulta. A inicialização da mola identifica um parâmetro de solicitação na classe do controlador. Nesta postagem, veremos “como ler o parâmetro de solicitação na inicialização da mola” e como os parâmetros da solicitação na inicialização da mola são validados.

Os parâmetros de solicitação são os valores enviados do lado do cliente. O servidor lerá os valores como parâmetros de solicitação. Esses valores são lidos apenas na natureza. Qualquer valor que deve ser enviado ao cliente é atribuído no objeto de resposta.

O Spring boot oferece suporte à validação do parâmetro de solicitação antes do processamento na classe do controlador. Aqui, veremos o procedimento passo a passo de Como configurar o parâmetro de solicitação na inicialização da mola, Como ler o parâmetro da solicitação na inicialização da mola e como validar os parâmetros da solicitação na inicialização da mola.

Etapa 1 - Adicionar dependência em pom.xml

A dependência “spring-boot-starter-validation” incluirá todos os jars necessários para a validação dos parâmetros de solicitação. O aplicativo da web precisa da dependência de “spring-boot-starter-browser”. Isso permitirá que o servidor tomcat comece como um aplicativo da web. A dependência de teste padrão “spring-boot-starter-test” será aplicada a pom.xml para o teste de unidade.

Se você criou um aplicativo spring boot, agora está tentando incorporar o código de validação e, em seguida, adicione a dependência “spring-boot-starter-validation” ao arquivo pom.xml. O arquivo pom.xml completo será conforme mostrado abaixo.

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.2.5.RELEASE</version>
    <relativePath /> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.yawintutor</groupId>
  <artifactId>Spring-Application</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>SpringBootValidation</name>
  <description>Spring Boot Project</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-validation</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>

</project>
```

Etapa 2 - Adicionar anotação @RequestParam

O parâmetro de consulta é identificado como o parâmetro de solicitação na classe do controlador com uma anotação @RequestParam. Os parâmetros de solicitação serão atribuídos ao valor do parâmetro de consulta automaticamente no início da primavera. Se o parâmetro de consulta não for definido, o parâmetro de solicitação será definido como nulo ou pelo valor padrão.

```
package com.yawintutor;

import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class TestController {

    @GetMapping("/student/")
    ResponseEntity<String> studentR(@RequestParam("age") int age) {
        return ResponseEntity.ok("Your age is " + age);
    }
}
```

Etapa 3 - Adicionar anotação @Validated para validação

A anotação @Validated é usada para especificar o controlador restante a ser validado. O aplicativo Spring Boot valida todos os métodos antes de ser chamado. @Min (5) é adicionado neste caso, para validar os parâmetros de solicitação no exemplo abaixo.

TestController.java

```
package com.yawintutor;

import javax.validation.constraints.Min;

import org.springframework.http.ResponseEntity;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
@Validated
public class TestController {

    @GetMapping("/student/")
    ResponseEntity<String> studentR(@RequestParam("age") @Min(5) int age) {
        return ResponseEntity.ok("Your age is " + age);
    }
}
```

Etapa 4 - Execute o aplicativo

Para validar o parâmetro de consulta, inicie o aplicativo na inicialização da mola. Ligue para o url abaixo que está executando sem erros.

```
http://localhost:8080/student/?age=5
```

Resultado

```
Your age is 5
```

Se a entrada inválida for passada como parte do url, a validação falhará. O url abaixo exibe o parâmetro de solicitação com dados inválidos.

```
http://localhost:8080/student/?age=2
```

Resultado

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Wed Mar 25 08:29:24 IST 2020

There was an unexpected error (type=Internal Server Error, status=500).

student.age: must be greater than or equal to 5

Exceção

A exceção abaixo aparecerá na janela do console ou no arquivo de log.

```
2020-03-25 08:29:24.577 ERROR 88142 --- [nio-8080-exec-1] o.a.c.c.C.[.[./].[dispatcherServlet] :  
Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request  
processing failed; nested exception is javax.validation.ConstraintViolationException: student.age: must be  
greater than or equal to 5] with root cause
```

```
javax.validation.ConstraintViolationException: student.age: must be greater than or equal to 5
```

Como visualizar o URL do pedido para depuração

Para entender a exceção, é importante saber como o url é chamado. Incapaz de ver ou registrar o valor real dentro do método. Porque a validação falha antes de chamar o método. A melhor maneira é permitir o registro de depuração adicionando o código abaixo no arquivo application.properties.

application.properties

```
logging.level.org.springframework.web.servlet.DispatcherServlet=debug
```

O log ficará como abaixo

```
2020-03-25 08:29:24.499 DEBUG 88142 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : GET  
"/student/?age=2", parameters={}  
2020-03-25 08:29:24.572 DEBUG 88142 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Failed  
to complete request: javax.validation.ConstraintViolationException: student.age: must be greater than or  
equal to 5
```

Outros links relacionados

[Como validar a variável de caminho no Spring Boot](#)
[Como validar os parâmetros de solicitação no Spring Boot](#)
[Como validar o corpo da solicitação no Spring Boot](#)
[Como validar os cabeçalhos de solicitação no Spring Boot](#)
[Como validar o Spring Boot Bean programaticamente ou manualmente](#)
[Como personalizar a mensagem de erro padrão no Spring Boot Validação](#)
[Como personalizar a mensagem de erro padrão usando @ControllerAdvice na validação do Spring Boot](#)