



Integrando Swagger com Java LTS e Spring Boot

O **Swagger** é essencial para documentar e testar APIs de forma eficiente. No ecossistema Java, sua integração com **Spring Boot** proporciona uma solução robusta e automatizada para o desenvolvimento de serviços web.

Com as diferentes versões **LTS do Java** (8, 11 e 17), a evolução do Swagger acompanha as inovações das plataformas Java, garantindo não apenas compatibilidade, mas também aproveitando ao máximo os novos recursos de cada versão.

Swagger com Java 8 e Spring Boot 2.1.x

- **Java 8** oferece um ambiente estável e bem conhecido para sistemas legados. A principal vantagem de usar Swagger com Java 8 é a sua base de código madura, com milhões de projetos em produção.
- **Spring Boot 2.1.x** é uma escolha excelente para esse cenário, pois foi desenvolvido com foco em estabilidade e performance.

```
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger2</artifactId>
  <version>2.9.2</version>
</dependency>
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger-ui</artifactId>
  <version>2.9.2</version>
</dependency>
```

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import springfox.documentation.swagger2.annotations.EnableSwagger2;
```

```
@RestController
@EnableSwagger2
public class ApiController {

    @GetMapping("/api/message")
    public String getMessage() {
        return "API rodando em Java 8 com Swagger!";
    }
}
```



Swagger com Java 11 e Spring Boot 2.3.x

- **Java 11** oferece melhorias significativas em relação ao Java 8, incluindo var e um novo HttpClient. Isso possibilita a criação de APIs mais modernas.
- **Spring Boot 2.3.x** traz melhorias de performance e segurança, sendo ideal para Java 11.

```
<dependency>
<groupId>org.springdoc</groupId>
<artifactId>springdoc-openapi-ui</artifactId>
<version>1.4.3</version>
</dependency>
```

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import io.swagger.v3.oas.annotations.Operation;
```

```
@RestController
public class ApiController {

    @Operation(summary = "Obter mensagem", description = "Retorna uma mensagem de boas-vindas")
    @GetMapping("/api/message")
    public String getMessage() {
        return "API rodando em Java 11 com Swagger!";
    }
}
```

Swagger com Java 17 e Spring Boot 2.6.x

- **Java 17** é a versão LTS mais recente e traz inovações como Pattern Matching e Records, que permitem construir APIs mais seguras e legíveis.
- **Spring Boot 2.6.x** aproveita todas essas melhorias, proporcionando mais eficiência e segurança para APIs RESTful.

```
<dependency>
<groupId>org.springdoc</groupId>
<artifactId>springdoc-openapi-ui</artifactId>
<version>1.6.13</version>
</dependency>
```

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import io.swagger.v3.oas.annotations.Operation;
```

```
@RestController
public class ApiController {

    @Operation(summary = "Obter mensagem com Java 17", description = "Mensagem usando novos recursos de Java 17")
    @GetMapping("/api/message")
    public String getMessage() {
        return "API rodando em Java 17 com Swagger!";
    }
}
```



Swagger com a Última Versão do Java e Spring Boot 3.x

- A última versão do **Java (JDK 23)**, com o **Project Loom**, proporciona inovações no gerenciamento de threads e outras melhorias de performance.
- **Spring Boot 3.x** é ideal para novos projetos que desejam utilizar o que há de mais moderno em termos de tecnologia Java.

```
<dependency>  
  <groupId>org.springdoc</groupId>  
  <artifactId>springdoc-openapi-ui</artifactId>  
  <version>1.7.0</version>  
</dependency>
```

```
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
import io.swagger.v3.oas.annotations.Operation;
```

```
@RestController  
public class ApiController {  
  
    @Operation(summary = "Exemplo com Project Loom", description = "Usa novas  
funcionalidades de threads com Swagger")  
    @GetMapping("/api/message")  
    public String getMessage() {  
        return "API rodando com Project Loom e Swagger!";  
    }  
}
```

Conclusão

A integração do **Swagger** com diferentes versões do **Java LTS** e **Spring Boot** facilita o desenvolvimento de APIs RESTful, promovendo documentação automatizada, eficiência e segurança. A escolha da versão do Java e Spring Boot depende das necessidades do projeto: estabilidade e compatibilidade para sistemas legados (Java 8), modernidade e performance para novos projetos (Java 17), ou a adoção de inovações (Java 23).

EducaCiência FastCode para a comunidade