# Java Development for API

Mastering the Art of API Creation with Java



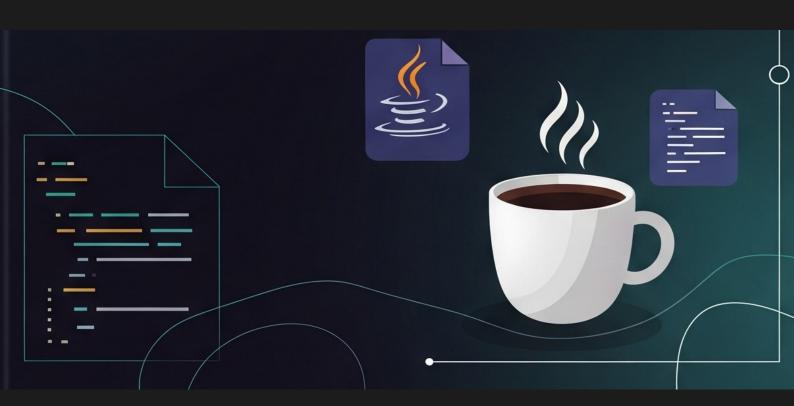






## Por Que Usar Spring no Desenvolvimento de APIs

O Spring Framework, especialmente com o módulo Spring Boot, simplifica drasticamente o desenvolvimento de APIs REST em Java. Abaixo, listamos suas principais vantagens com exemplos práticos.



### Início Rápido com Spring Boot

### Início Rápido com Spring Boot

O Spring Boot permite criar APIs com mínima configuração. Ele elimina a necessidade de configurar servidores e arquivos XML complexos.

Exemplo de aplicação REST básica:

Com apenas essas anotações, a aplicação já responde a requisições HTTP.

### Injeção de Dependência Simplificada

### Injeção de Dependência Simplificada

Com Spring, a injeção de dependência é nativa, promovendo código limpo e testável.



Exemplo com @Autowired:

```
Spring Framework
    @Service
    public class ProdutoService {
        public List<String> listar() {
            return List.of("Teclado", "Mouse");
        }
   }
    @RestController
    public class ProdutoController {
        @Autowired
        private ProdutoService produtoService;
12
        @GetMapping("/produtos")
        public List<String> listarProdutos() {
15
            return produtoService.listar();
        }
18
    }
```

Com Spring, a injeção de dependência é nativa, promovendo código limpo e testável.

# Criação Facilitada de Endpoints REST

### Criação Facilitada de Endpoints REST

Spring simplifica o mapeamento de URLs para métodos Java, usando anotações como @GetMapping, @PostMapping, etc.

#### Exemplo POST:

```
Spring Framework
   @PostMapping("/produtos")
   public String salvarProduto(@RequestBody String nome) {
       return "Produto salvo: " + nome;
   }
```

### Integração com Banco de Dados (Spring Data JPA)

### Integração com Banco de Dados (Spring Data JPA)

Você cria repositórios com poucas linhas de código e sem SQL explícito.

#### Exemplo com JPA:

```
Spring Framework
@Entity
   public class Produto {
       @Id @GeneratedValue
       private Long id;
       private String nome;
   public interface ProdutoRepository extends JpaRepository<Produto, Long> {}
```

Você cria repositórios com poucas linhas de código e sem SQL explícito.

# Tratamento de Erros com @ExceptionHandler

### Tratamento de Erros com @ExceptionHandler

Você pode personalizar respostas de erro de forma elegante.



#### Exemplo de tratamento de erro:

```
•••
                                 Spring Framework
   @RestControllerAdvice
   public class ErroHandler {
       @ExceptionHandler(Exception.class)
       public ResponseEntity<String> handleErro(Exception e) {
           return ResponseEntity.status(500).body("Erro: " + e.getMessage());
   }
```

O Spring Framework economiza tempo, reduz código repetitivo e oferece uma arquitetura sólida para APIs REST. Se você produtividade com robustez, Spring é o caminho certo.