

CETEJ35 - Java Web - JAVA_XXX (2024_01)

[Meus cursos](#) / [CETEJ35 - Web \(2024_01\)](#) / [Semana 07: 14/10 a 20/10](#) / [API Reativa](#)

API Reativa

✓ **Feito:** Ver

✓ **Feito:** Gastar pelo menos 20 minutos na atividade

A fazer: Passar pela atividade até o fim

Fecha: segunda-feira, 2 dez. 2024, 00:00

Nesta aula vamos construir uma nova aplicação que funciona como uma API, recebendo dados usando a arquitetura REST. Para fazer isso, vamos usar uma nova tecnologia do Spring - o Spring WebFlux.

EXECUTANDO O PROJETO

Nesse ponto, o código necessário para nossa API de gerenciamento de tarefas está pronto. Esse é um projeto baseado no Spring Boot, assim como nosso projeto anterior. Por isso, para executar o projeto basta abrir o terminal na pasta do projeto e digitar: `mvn clean spring-boot:run`. Contudo, existe alguns detalhes que precisamos nos atentar.

Primeiro, esse projeto usa o MongoDB. Diferente do que fizemos anteriormente, não temos um MongoDB embutido no projeto. Por isso, antes de executar esse projeto, você precisa garantir que você tem o MongoDB rodando localmente em seu computador. Se você nunca teve contato com o MongoDB, minha sugestão é que você use uma imagem Docker, conforme orientado na aba "Gerenciador de Tarefas". Se quiser testar seu ambiente para garantir que está tudo funcionando, você pode rodar os testes unitários disponíveis no [projeto no Github](#). Basta clonar o projeto localmente, abrir a pasta do projeto no terminal, e executar: `mvn test`

Segundo, esse projeto **não tem** interface gráfica. Isso é esperado, uma vez que esse projeto consiste de uma API. Dessa forma, para ver as solicitações funcionando, você precisa de um cliente HTTP. Se precisar de sugestões, veja as informações deixadas na aba "Gerenciador de Tarefas".

Finalmente, execute o projeto usando `mvn clean spring-boot:run` e use seu cliente HTTP para fazer solicitações.

```
sh-3.2$ http get localhost:8080/todos
HTTP/1.1 200 OK
Content-Type: application/json
transfer-encoding: chunked

[]

sh-3.2$ http post localhost:8080/todos id=9a87sdf98 titulo=Hello descricao=World feito=false
HTTP/1.1 200 OK
Content-Length: 69
Content-Type: application/json

{
  "descricao": "World",
  "feito": false,
  "id": "9a87sdf98",
  "titulo": "Hello"
}

sh-3.2$ http post localhost:8080/todos id=9a87sdf98 titulo=Hello descricao=World feito=false
```


Retroceder

Avançar

◀ Verificação de aprendizado - Integração

Seguir para...

Atividade II WebConf ▶

✉ Contate o suporte do site 





Você acessou como RAFAEL ROCHA DA SILVA PROENCA (Sair)
CETEJ35 - Web (2024_01)

- Tema
- Adaptable
- Boost
- Clássico
- Campus
- Apucarana
- Campo Mourão
- Cornélio Procópio
- Curitiba
- Dois Vizinhos
- Francisco Beltrão
- Guarapuava
- Londrina
- Medianeira
- Pato Branco
- Ponta Grossa
- Reitoria
- Santa Helena
- Toledo
- UTFPR
- Ajuda
- Chat UTFPR
- Calendário Acadêmico
- Biblioteca
- e-Mail
- Nuvem (OwnCloud)
- Produção Acadêmica

Secretaria Acadêmica
Sistemas Corporativos
Sistema Eletrônico de Informação - SEI
Suporte ao usuário
Criação de curso
Comunidade
Português - Brasil (pt_br)
Deutsch (de)
English (en)
Português - Brasil (pt_br)

Resumo de retenção de dados

Baixar o aplicativo móvel.

 Dê um feedback sobre este software 

Universidade Tecnológica Federal do Paraná - UTFPR

Suporte ao usuário

