

SENAC

Campus Santo Amaro



TADS – Análise Desenvolvimento de Sistemas

PW – Programação Web

Desenvolvimento de programas API

Orientada

ADO #3

Nome: João Victor Matulis

Professor: Veríssimo – carlos.hvpereira@sp.senac.br

26/10/2022

Contents

1 Especificação da ADO	1
1.0.1 Objetivos desta ADO	1
1.0.2 Construção de API	1
1.0.3 O que deve ser produzido	2
2 Código API	2
3 Demonstração de teste do código	4

Chapter 1

Objetivo desta ADO

1.0.1 Objetivos desta ADO

O objetivo principal é proporcionar ao aluno praticar o desenvolvimento de programas, no lado do servidor (API), utilizando o conjunto de boas práticas API Rest. O objetivo subjacente é proporcionar ao aluno condições de desenvolver soluções robustas e com qualidade.

1.0.2 Construção de API

A produção desta ADO deverá ser embasada nos elementos estudados em sala de aula:

- Node JS
- Protocolo HTTP
- Metodos (verbos)
 - GET (Recuperar)
 - POST (Criar)
 - PUT (Editar)
 - DELETE (Remover)
- Requisições
 - Query PARAMS
 - Router PARAMS
 - Body

1.0.3 O que deve ser produzido

Deverá ser desenvolvido uma API para validação de um login, tendo como valores de validação um código de usuário e uma senha. A API deverá estar preparada para receber um endpoint com um usuário (User ID) e uma senha.

Chapter 2

Código API

```
const express = require('express');
const server = express();

server.use(express.json())
//criação do array simulando o banco de dados
const userDB = [
  {
    idUser : "Jonas",
    senhaUser : "123"
  },
  {
    idUser : "Rodolpho",
    senhaUser : "carrinho"
  }
]

server.get("/signin", (req,res) => {
  const {user, senha} = req.body;
  let verificar = acessaUserDB(user, senha);
  if(verificar == 1){
    res.json("Usuario não informado")
  }
  else if(verificar == 2){
    res.json("Senha não informada")
  }
}
```

```

    }
    else if(verificar == 3){
        res.json("Usuario não encontrado no banco de
dados")
    }
    else if(verificar == 4){
        res.json("Senha não bate com a senha do BD")
    }
    else if(verificar == 5){
        res.json("Login ok")
    }
})

server.listen(3000);

function acessaUserDB(usuario, senha){
    existeUser = false;
    //caso o usuario não seja informado
    if(usuario == null || usuario == "" || usuario ==
''){
        return 1;
    }
    //caso a senha não seja informada
    if(senha == null || senha == "" || senha == ''){
        return 2;
    }
    for(var key in userDB){
        if(usuario === userDB[key].idUser){
            if(senha === userDB[key].senhaUser){
                //caso o login seja efetuado com
sucesso
                existeUser = true
                break
            }
        }
    }
}

```

```

        //caso o usuario esteja correto porém a
senha não
        if(senha != userDB[key].senhaUser){
            return 4
        }
    }

    if(existeUser){
        return 5
    }
    //caso o usuario não exista no banco de dados
    else{
        return 3
    }
}

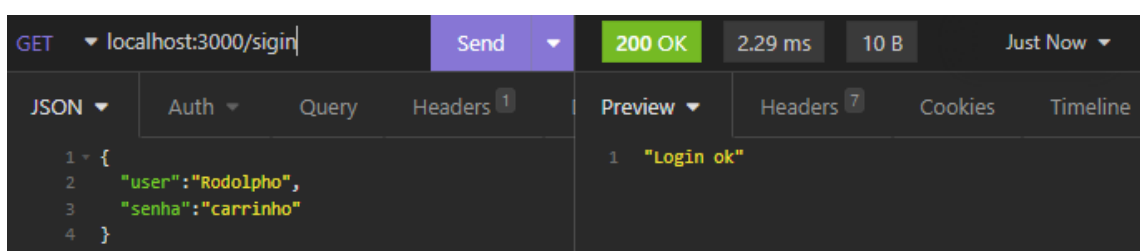
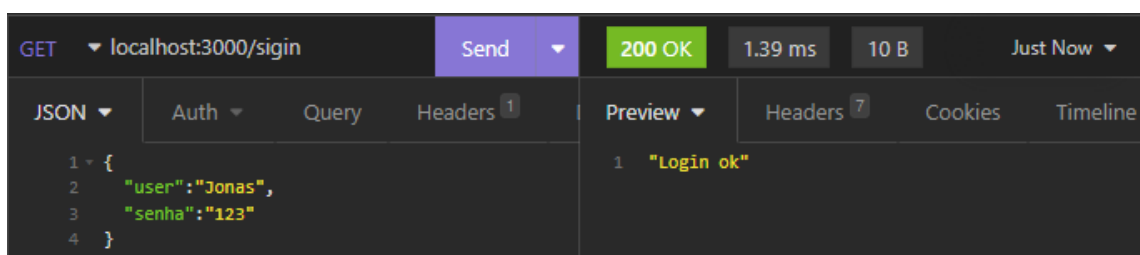
```

Chapter 3

Demonstração de teste do código

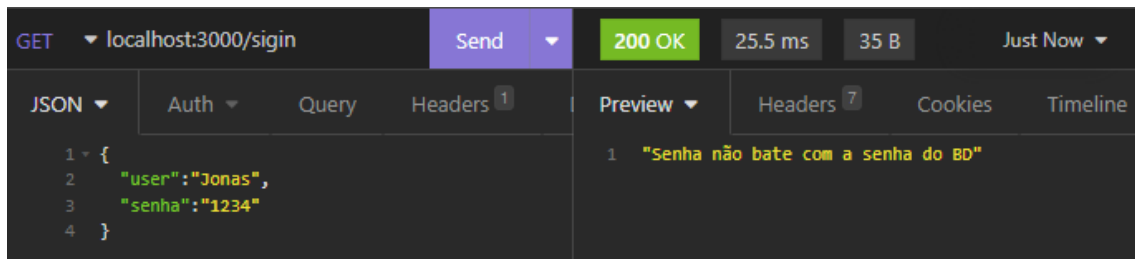
Caso 1:

-usuário e senha estejam corretos:

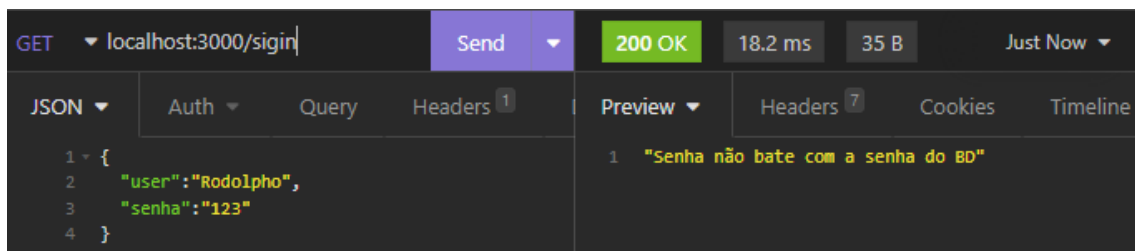


Caso 2:

-usuário correto porém senha errada:



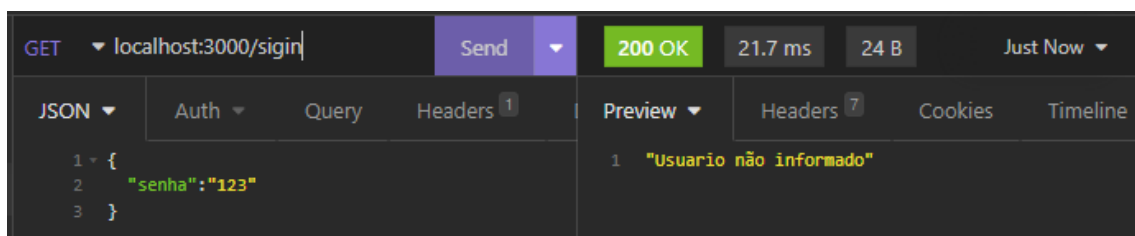
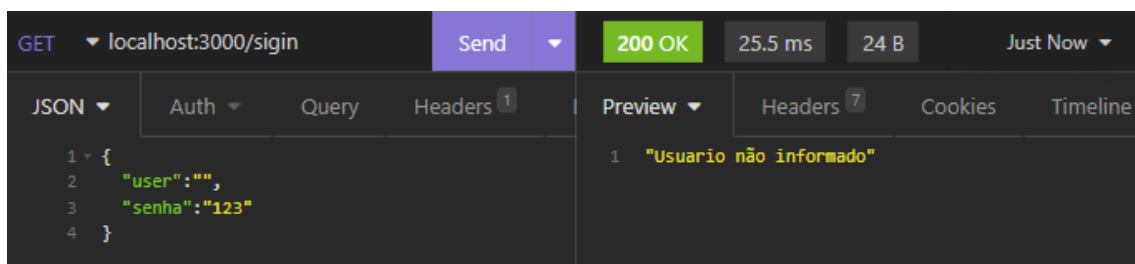
A senha da conta Jonas é 123



A senha da conta Rodolpho é carrinho

Caso 3:

-usuário não informado:



Caso 4:

-Senha não informada:

A screenshot of the Postman application showing a GET request to `localhost:3000/signin`. The request body is a JSON object: `{ "user": "Rodolpho", "senha": "" }`. The response status is `200 OK` with a response time of `2.93 ms` and a response size of `22 B`. The response body is a plain text message: `"Senha não informada"`.

A second screenshot of the Postman application for the same request. The response body is `"Senha não informada"`.

Caso 5:

-Usuário não encontrado no banco de dados:

As únicas contas guardadas são: Jonas e Rodolpho

A screenshot of the Postman application showing a GET request to `localhost:3000/signin`. The request body is a JSON object: `{ "user": "Julia", "senha": "123" }`. The response status is `200 OK` with a response time of `2.77 ms` and a response size of `43 B`. The response body is a plain text message: `"Usuario não encontrado no banco de dados"`.

A second screenshot of the Postman application for the same request. The response body is `"Usuario não encontrado no banco de dados"`.