Arthur Silva

Softex Recife

**ATIVIDADE 5 – MÓDULO WEB SERVICES**

Demonstre como fazer as quatro operações de CRUD e suas rotas para obter o recurso de um livro que está no estoque de uma livraria. Um livro possui informações como nome, autor e ISBN.

Para simular um CRUD de uma base de livros, usaremos o domínio: [www.biblioteca.com](http://www.biblioteca.com), com os dados abaixo:

Var livros = [

{Nome: livro1,

Autor: autor1,

ISBN: 123456-123},

{Nome: livro2,

Autor: autor2,

ISBN: 123456-456},

]

Para fazer o **CRUD** (Created, Read, Update, Delete) é necessário utilizar os métodos HTTP POST, GET, PUT e DELETE e cada um deles é configurado em uma URL diferente, além de que cada um necessitará de dados para realizar a operação.

Para fazer a **C**riação de um novo registro, ou inserir um livro novo, utiliza o método POST passando o novo livro no corpo da requisição, da seguinte forma:

POST | www.biblioteca.com/

Corpo da requisição um objeto livro dentro de um json:

{

{ “nome”: “livro3”,

“Autor”: “autor3”

“isbn”: 123456-789}

}

Programação no servidor:

metodoPost(‘/’, request, response) =>{

livros.push(request.body)

return response(“Livro adicionado com Sucesso”)

}

Para fazer o **R**ead (leitura) de um registro, utiliza o método GET passando o que se deseja buscar na url da requisição, da seguinte forma:

GET | www.biblioteca.com/livros/3

Livros se refere ao recurso e 3 ao identificador do livro

Programação no servidor:

//o 3 se refere ao terceiro livro, mas no array o terceiro tem index 2

metodoGet(‘/livros/{id}’, request, response) =>{

index = request.id - 1 //3 – 1 = index 2

if(index < 0){

index = 0}

return response(livros[index])

}

Para fazer o **U**pdate (atualização) de um registro, utiliza o método PUT passando os dados atualizados no corpo da requisição, da seguinte forma:

PUT | www.biblioteca.com/livros/3

Livros se refere ao recurso e 3 ao identificador do livro

Corpo da requisição um objeto livro dentro de um json:

{

{ “nome”: “livro3.123”, //atualização do nome

“Autor”: “autor3”

“isbn”: 123456-789}

}

Programação no servidor:

//o 3 se refere ao terceiro livro, mas no array o terceiro tem index 2

metodoPut(‘/livros/{id}’, request, response) =>{

index = request.id - 1 //3 – 1 = index 2

if(index < 0){

index = 0

}

livros[index] = request.body

return response(“Livro atualizado”)

}

Para fazer o **D**elete de um registro, utiliza o método Delete passando o index do livro para ser deletado na url, da seguinte forma:

DELETE | www.biblioteca.com/livros/3

Livros se refere ao recurso e 3 ao identificador do livro

Programação no servidor:

//o 3 se refere ao terceiro livro, mas no array o terceiro tem index 2

metodoDelete(‘/livros/{id}’, request, response) =>{

index = request.id - 1 //3 – 1 = index 2

if(index < 0){

index = 0

}

Livros.splice(index , 1) //remove o item do array

return response(“Livro Removido”)

}