

Projeto Integrador IV-B

Nome: Diego Cardoso Gressler





Proposta do trabalho

Criação de uma API em Node.js e sua implantação com HTML salvando os dados em um arquivo Json.



Tecnologias utilizadas

- HTML, CSS (Bootstrap) e Node.js
- Módulos Node utilizados:
 - EJS (utilização do node com HTML)
 - JS (manipulação de arquivos)
 - Method-Override (utilização de métodos PUT e DELETE no html)
 - EXPRESS (criação do servidor)



Código Node - Rotas

```
// Busca todos os dados
router.get('/', function(req, res){
  const content = readFile();
  // Renderiza a página index com a lista de contatos
  res.render('./index',{ lista : content });
});
```



Código Node - Rotas

```
// Carrega o formulário de adicionar  
router.post('/', function(req, res){  
  res.render('./form_add');  
});
```



Código Node - Rotas

```
// Insere o contato ao arquivo
router.post('/adicionar', function(req, res){
  const currentContent = readFile();
  const {nome, endereco, cep, data nascimento, telefone} = req.body;
  // Gera o id aleatório para a pessoa
  const cliente_id = Math.random().toString(32).substr(2,9);
  // Adiciona no array para ser adicionado no arquivo json
  currentContent.push({cliente_id, nome, endereco, cep,
data nascimento, telefone});
  writeFile(currentContent);
  const content = readFile();
  // Começo de testes com o modulo ejs
  res.render('./index',{ lista : content });
});
```



Código Node - Rotas

```
// Rota de busca de id específico
router.get('/edit/:cliente_id', function(req, res){
  const {cliente_id} = req.params;
  //res.send(cliente_id);
  const currentContent = readFile();
  // Busca elemento com a id específica
  const selectedItem = currentContent.find((item) =>
item.cliente_id === cliente_id);
  res.render('./edit',{ contato : selectedItem});
});
```



Código Node - Rotas

```
// Edita o contato selecionado
router.put('/:cliente_id', function(req, res){
  const {cliente_id} = req.params;
  const {nome, endereco, cep, data_nascimento, telefone} =
req.body;
  const currentContent = readFile();
  const selectedItem = currentContent.findIndex((item) =>
item.cliente_id === cliente_id);
  const {cliente_id: cId, nome: cNome, endereco: cEndereco, cep:
cCep, data_nascimento: cData, telefone: cTelefone} =
currentContent[selectedItem];
```




Código Node - Rotas

```
// Cria objeto para ser atualizado no arquivo
const newContato = {
  cliente id: cId,
  nome: nome ? nome : cNome,
  endereco: endereco ? endereco : cEndereco,
  cep: cep ? cep : cCep,
  data nascimento: data nascimento ? data nascimento : cData,
  telefone: telefone ? telefone : cTelefone
}

currentContent[selectedItem] = newContato;
writeFile(currentContent);
const content = readFile();

// Começo de testes com o modulo ejs
res.render('./index',{ lista : content });
});
```



Código Node - Rotas

```
// Exclui o contato selecionado
router.delete('/:cliente_id', function(req, res){
  const {cliente_id} = req.params;
  const currentContent = readFile();
  // Busca elemento com a id especifica
  const selectedItem = currentContent.findIndex((item) => item.cliente_id
=== cliente_id);
  // Remove do array o item com a id
  currentContent.splice(selectedItem, 1);
  writeFile(currentContent);
  // Le novamente o arquivo
  const content = readFile();
  res.render('./index',{ lista : content });
});
```



Exemplo do uso do EJS

```
<%for(var i=0;i<lista.length;i++) {%>  
  <tr>  
    <td><%=lista[i].cliente id%></td>  
    <td><%=lista[i].nome%></td>  
    <td><%=lista[i].endereco%></td>  
    <td><%=lista[i].cep%></td>  
    <td><%=lista[i].data nascimento%></td>  
    <td><%=lista[i].telefone%></td>  
  <%}%>
```



Exemplo do uso MethodOverride

- DELETE

```
<form name="delete" action="/<%=lista[i].cliente_id%>?_method=DELETE"
method="post">
  <input type="hidden" name="_method" value="DELETE">
```

- PUT

```
<form action="/<%=contato.cliente_id%>?_method=PUT" method="post">
  <input type="hidden" name="_method" value="PUT">
```



OBRIGADO !!!