

## Lista 4 de Exercício – React – Componentes – API

### Desenvolvimento Web – Lab. Profª Angelina V S Melaré

Siga as orientações abaixo:

#### Parte 1 – Continuação conceito de COMPONENTES

- 1- Vocês farão uma melhoria no projeto da aula anterior criando um componente “Exibir” que irá listar todos os clientes que fizeram pedido e estão na fila de espera, conforme imagem a seguir:

Nome:

Telefone:

- Wallace Souza - (11)9888-8888
  - Victoria Mendes - (11)9888-5676
- 2- Vá na pasta “projeto-componentes” (cd)
  - 3- Inicie o projeto para ir visualizando as mudanças realizadas  
npm start
  - 4- Altere seu arquivo APP:
    - a) Após a criação dos elementos de estado criados (nome, telefone, ...) acrescente o cliente  
**const [clientes, setClientes] = useState([]);**  
\*\*Perceba que ele será um array de objeto por isso o [] e ele conterà os dados inseridos do cliente.
    - b) Depois do código de validar os dados e confirmar o sucesso de cadastro inclua o novo cliente na lista atual de clientes.  
**let novo={nome: nome, telefone: telefone};**  
**setClientes([...clientes, novo]);**  
\*\* Perceba que a variável objeto novo é criada e tem seus valores atribuídos. O nome receberá o nome e demais dados inseridos.  
\*\*Esse novo cliente será acrescido aos clientes atuais por meio da função setClientes().

- 5- Após encerrar o formulário (/form) insira o novo componente “Exibir”, enviando para ele o array de objetos clientes

```
</form>
```

```
<Exibir conteudo={clientes} />
```

**\*\* Não se esqueça de importar o componente no início do componente APP.**

- 6- Crie o componente Exibir inserindo o código

```
export default function Exibir( {conteudo} ) {  
  let lista= conteudo.map( dados => {return <li> {dados.nome} - {dados.telefone}</li>} )  
  
  return (  
    <ul>  
      {lista}  
    </ul>  
  )  
}
```

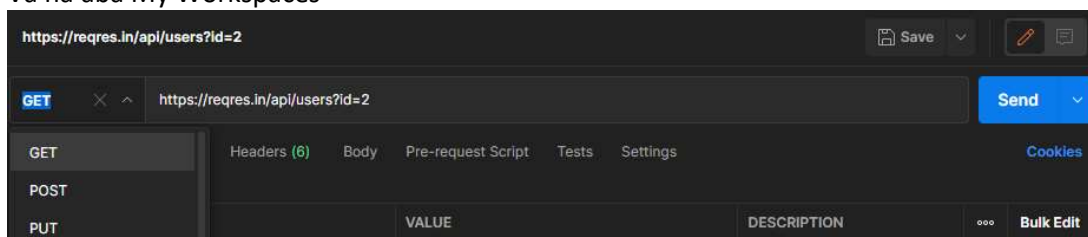
**\*\* Perceba que poderia ter sido colocado diretamente o commando map no lugar da variável lista. As chaves é que definem que não é um texto e sim um conteúdo a ser exibido (após um processo/função)**

```
{ conteudo.map( dados => {return <li> {dados.nome} - {dados.telefone}</li>} ) }
```

## Parte 2 – API

### 2-1 Postman

- 1- Instale a ferramenta de testes Postman <https://www.postman.com/downloads/>
- 2- Vá na aba My Workspaces



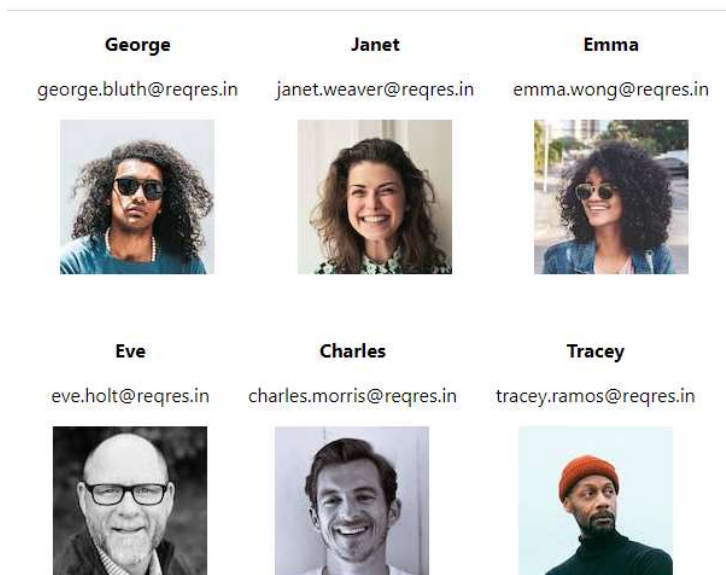
Abrirá uma janela para colocar a URL da API, escolha o método/verbo (get, post...). Depois pressione o botão “Send” para ver o retorno (conforme mostrado abaixo). Veja que tem a opções Preview e Visualize (conforme abaixo).

```
Body Cookies Headers (19) Test Results
Pretty Raw Preview Visualize
{"data":{"id":2,"email":"janet.weaver@reqres.in","first_name":"janet","last_name":"weaver","image.jpg":"https://reqres.in/img/faces/1-image.jpg","support":{"url":"https://reqres.in/#support-headin-appreciated!"}}}
```

- 3- Faça os testes no Postman e veja o retorno
  - a) Use o verbo (método) GET na API <http://viacep.com.br/ws/18020320/teste>
  - b) Use o GET em <https://reqres.in/api/users?id=2>
  - c) Use o POST em <https://reqres.in/api/users?id=2>
  - d) Use o método POST para a <https://viacep.com.br/ws/18500000/json>
  - e) Use o método GET para <https://viacep.com.br/ws/18080-200/json>
  - f) Use o método GET para <https://viacep.com.br/ws/18080-200/xml>

## 2-2 Projeto consome serviço de API

- 4- Crie um projeto no React identificado como projeto-consome-api  
O projeto irá fazer uma consulta numa API pública e retornará os dados das pessoas cadastradas.  
A interface deve ficar parecida com a imagem a seguir:



- 5- Apague o conteúdo do APP padrão.
- 6- Importe o arquivo de estilização do App (item 13)
- 7- Importe todos os componentes do React (\* as React from "react")
- 8- Crie o atributo "users" de estado do componente, conforme comando a seguir:  
const [users, setUsers] = React.useState([]);

- 9- Cria a função assíncrona que irá consumir o serviço da API buscando os dados das pessoas cadastradas. Veja que irá retornar os dados no formato “json”. Os dados do “json” serão armazenados no atributo criado anteriormente (users) por meio da função setUsers()

```
const funcaoAssync = async () => {  
  const resultado = await fetch("https://reqres.in/api/users/");  
  const json = await resultado.json();  
  setUsers(json.data);  
};
```

- 10- A função deve ser chamada após sua criação. Para isso apenas coloque sua chamada:  
funcaoAssync()

- 11- Como retorno do componente APP deve ser criada uma div com o nome da classe e configurada como “flex” (estilo que permitirá que os dados fiquem dispostos em todo a largura no dispositivo de visualização) Veja a explicação do estilo no item 13.  
Exemplo: return (<div ...> código de retorno </div>)

- 12- Dentro da div criada use o método map para percorrer o array de objeto das pessoas e trazer cada usuário.

```
{users.map((user) => { return (  
  <div key={user.id}>  
    <p> <strong> {user.first_name} </strong> </p>  
    <p> {user.email} </p>  
    <img key={user.avatar} src= {user.avatar} />  
  </div> ); }}
```

- 13- Pode-se configurar o estilo da “div” como:

```
.flex {  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: center;  
  align-items: center;  
}
```