

Roteiro Didático de Aula Aula 04 – React Native 13/11/2020

Assuntos que serão abordados:

Consumo de API

Referências

Partindo do pressuposto que o roteiro da aula 03 foi finalizado, continuaremos com nosso CRUD.

Na aula passada fizemos o GET e o POST

Agora vamos para o delete:

```
routes.delete('/deletecliente/:id', async (request, response)=>{
    //única diferença é eu agora vamos usar o parmas vindo junto do link, o id do cliente
que desejamos deletar
    const dataBase = await db;
    dataBase.run(`DELETE FROM clientes WHERE rowid=?`, [`${request.params.id}`], functio
n(err: { message: any; }) {
    //ao invés de body, usar params
    if (err) {
        return console.error(err.message);
    }
        return response.send('ok')
    });
});
```

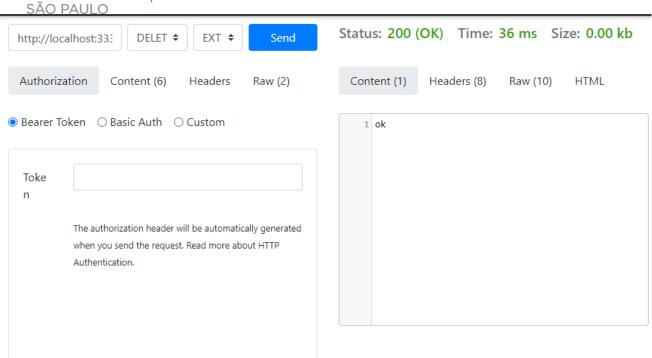
Vamos testar!!

Então digitaremos no endereço:

http://localhost:3334/deletecliente/12

no caso estaremos deletando o cliente com id 12

Depois de digitar o endereço você deve selecionar a opção "delete" e pressionar "send"



Agora para testar vamos fazer um get e ver o resultado

```
{
    id: 11,
    name: 'vivian',
    telefone: 113698524,
    cep: 3694910,
    n_casa: 789
}
{
    id: 13,
    name: 'Caique',
    telefone: 11000000000,
    cep: 96325874,
    n_casa: 200
}
{
    id: 14,
    name: 'fulano',
    telefone: 12343543,
    cep: 963222274,
    n_casa: 200
}
```

Nós deletamos o usuário com id 12, para deletar outro usuário basta colocar o id desejado no final da URL

Agora para rota PUT, de edição, vamos fazer uma rota para cada dado, assim podemos editar apenas um dado do cliente sem mexer nos demais dados dele, como para corrigir um erro de digitação ou para editar o endereço ou telefone

Vamos escrever esse codigo:

```
SÃO PAULO
routes.put('/editcliente/name', async (request, response)=>{
const dataBase = await db;
let sql = `UPDATE clientes
          SET name = ?
          WHERE id = ?`;
dataBase.run(sql, [`${request.body.name}`, `${request.body.id}`], function(err: { message: any
; }) {
if (err) {
 return console.error(err.message);
return response.send(200)
});
});
routes.put('/editcliente/telefone', async (request, response)=>{
 const dataBase = await db;
  let sql = `UPDATE clientes
            SET telefone = ?
            WHERE id = ?`;
  dataBase.run(sql, [`${request.body.telefone}`, `${request.body.id}`], function(err: { messag
e: any; }) {
  if (err) {
    return console.error(err.message);
  return response.send(200)
  });
});
  routes.put('/editcliente/cep', async (request, response)=>{
    const dataBase = await db;
    let sql = `UPDATE clientes
              SET cep = ?
              WHERE id = ?`;
    dataBase.run(sql, [`${request.body.cep}`, `${request.body.id}`], function(err: { message:
any; }) {
    if (err) {
      return console.error(err.message);
    return response.send(200)
    });
    });
    routes.put('/editcliente/n-casa', async (request, response)=>{
      const dataBase = await db;
      let sql = `UPDATE clientes
                SET n_casa = ?
                WHERE id = ?`;
      dataBase.run(sql, [`${request.body.n_casa}`, `${request.body.id}`], function(err: { mess
age: any; }) {
      if (err) {
        return console.error(err.message);
      return response.send(200)
      });
```

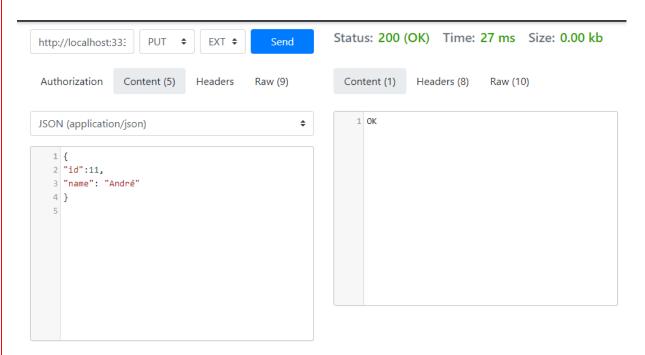


Vamos ver se todos deram certo?

Primeiro o name:

No link: http://localhost:3334/editcliente/name

Vamos ter esse retorno:



Vamos dar um GET em http://localhost:3334/clientes para testar

```
express deprecated res.send(status): Use res.sendStatus(status) instead src\routes.ts:50:25
express deprecated res.send(status): Use res.sendStatus(status) instead src\routes.ts:19:25
{
   id: 11,
    name: 'André',
    telefone: 113698524,
    cep: 3694910,
    n_casa: 789
}
{
   id: 13,
   name: 'Caique',
   telefone: 11000000000,
   cep: 96325874,
   n_casa: 200
}
```

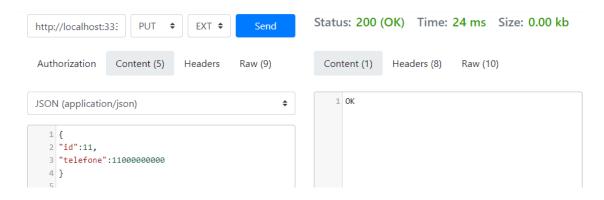
Sucesso!



Agora telefone...

No link: http://localhost:3334/editcliente/telefone

Teremos esse resultado:



Vamos dar um GET em http://localhost:3334/clientes para testar

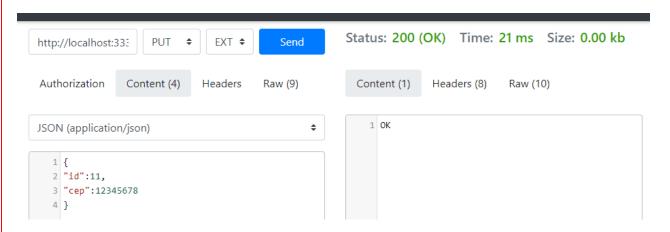
```
id: 11,
    name: 'André',
    telefone: 11000000000,
    cep: 3694910,
    n_casa: 789
}

id: 13,
    name: 'Caique',
    telefone: 1100000000,
    cep: 96325874,
    n_casa: 200
}
```

Agora cep:

No link: http://localhost:3333/editcliente/cep

Tendo esse retorno:



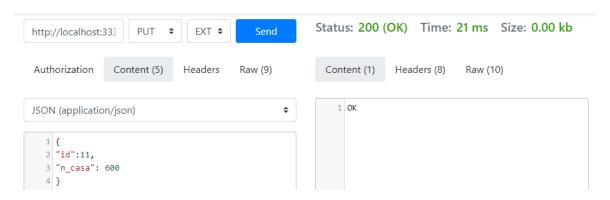
Vamos dar um GET em http://localhost:3334/clientes para testar

```
SÃO PAULO
}
express deprecated res.send(status): Use res.sendStatus(status) instead src\routes.ts:74:25
{
   id: 11,
   name: 'André',
   telefone: 11000000000,
   cep: 12345678,
   n_casa: 789
}
{
   id: 13,
   name: 'Caique',
   telefone: 1100000000,
   cep: 96325874,
   n_casa: 200
}
```

Agora por último o n_casa:

No link: http://localhost:3333/editcliente/n-casa

Retorno parecido com esse:



Vamos ver se mudou?

```
h_casa: 200
}
express deprecated res.send(status): Use res.sendStatus(status) instead src\routes.ts:86:25
{
    id: 11,
        name: 'André',
        telefone: 11000000000,
        cep: 12345678,
        n_casa: 600
}
{
    id: 13,
        name: 'Caique',
        telefone: 1100000000,
        cep: 96325874,
        n_casa: 200
}
```

Referências

Rocketseat. (s.d.). *Expo.* Fonte: https://www.youtube.com/watch?v=ZaDpDIPr25M Yarn. (s.d.). Fonte: https://yarnpkg.com/getting-started/install