

Criando os objetos update e delete da API

Objeto update

O objeto update quando utilizamos para realizar requisições em API é chamado de **PUT**.

A especificação para APIs HTTP REST define que para realizar uma atualização, a API deve esperar receber o id do objeto que queremos atualizar na URL no formato /coleção/id, enquanto que no corpo da requisição devemos enviar o novo objeto que vai substituir o original na coleção.

Sendo assim, vamos configurar mais uma função no nosso controller funcionário. No arquivo funcionarioController.js, crie a codificação para a função update.

```
//UPDATE
static async FuncionarioUpdate(req,res) {
  const matricula = req.params.id;
  let nome = req.body.nome;
  let endereco = req.body.endereco;
  let telefone = req.body.telefone;
  let email = req.body.email;
  let nascimento = req.body.nascimento;
  const funcionario = {
    nome: nome,
    endereco: endereco,
    telefone: telefone,
    email: email,
    nascimento: nascimento
  };
  await Funcionario.update(funcionario, { where: { matricula:matricula } })
  res.json({message: "Cadastro atualizado com sucesso!"});
}
```

Ela é bem parecida com a função create.

As diferenças são o **req.params.id** no início da função e o **Funcionario.update** que realiza atualização do objeto que possui a matricula igual ao req.params.id. Vamos agora no arquivo index.js para riar nossa rota.

```
//PUT - UPDATE
app.put("/Funcionarios/:id",funcionarioController.FuncionarioUpdate);
```

Nossa rota ficou com o mesmo caminho da rota get. Porém, já vimos antes que podemos ter caminhos de rotas iguais, mas com requisições diferentes. Na aplicação uma requisição é get e outra é put.

Objeto delete

Finalizando o CRUD da nossa API, vamos criar uma rota que espere uma requisição delete com a matricula do funcionário a ser excluído na URL.

Vamos criar nossa função no arquivo funcionarioController.js.

```
//DELETE
static async FuncionarioDelete(req,res) {
  const matricula = req.params.id;
  await Funcionario.destroy({ where: { matricula: matricula }});
  res.json({message: "Funcionário excluído com sucesso!"});
}
```

Solicitamos uma exclusão do objeto que possui a matricula igual a recebida via json.

Retornamos para o solicitante a exclusão realizada com sucesso!

Vamos criar nossa rota no arquivo index.js.

A rota terá o mesmo caminho da requisição get e put, porém teremos uma requisição com o tipo delete.

```
//DELETE - Excluir
app.delete("/Funcionarios/:id",funcionarioController.FuncionarioDelete);
```

Finalizamos a realização dos serviços de put/update e delete da nossa API.

Na próxima aula, você verá como testar nossa API utilizando o postman.

Até lá...