



AULA 05 – Vídeo 02

Introdução

Continuando a organização dos arquivos de nossa API, iremos agora criar o arquivo que conterá as rotas. Posteriormente, separaremos as rotas dos tratamentos, jogando estes para os *controllers* ou *services*.

Em primeiro lugar, criaremos uma pasta 'routes' dentro do diretório 'src'. Dentro do diretório 'routes' recém-criado, criaremos um arquivo chamado 'PerfilRoute.js', que conterá as rotas de perfil. Moveremos todas as rotas que agora estão em 'server.js' para 'PerfilRoute.js', com exceção da rota raiz, '/perfil' pelo método GET. Também moveremos os perfis que estão em um *array*, juntamente com a variável 'geradorID', para 'PerfilRoute.js'.

As rotas em 'PerfilRoute.js' precisam da instância do express, que é um objeto único e que está em 'server.js'. Para que elas possam utilizá-la, podemos importar o arquivo 'PerfilRoute.js', colocando-o na constante 'perfilRoute'. Para que o servidor utilize as rotas, devemos chamar o método 'use()' do express, passando-lhe a rota e o responsável pelo tratamento, ou seja, 'perfilRoute'. Ao final, 'server.js' ficará da seguinte forma:

```
const express = require("express");
const          perfilRoute                               =
require("./routes/PerfilRoute");

const api = express();
api.use(express.json());
api.use('/perfil', perfilRoute);

api.get("/", (req, res) => {
```

```

        res.send("Bem-vindo(a) ao Perfil Profissional
API");
    });

module.exports = api;

```

Agora, precisamos criar um roteador em 'PerfilRoute.js'. Para isso, instanciamos o express no referido arquivo e chamamos seu método 'Router()':

```

const express = require("express");
const router = express.Router();

```

Com isso, substituímos os objetos 'app' por 'router'. Além disso, como em 'server.js' chamamos o método 'use()' na rota '/perfil', agora devemos retirar todos os '/perfil' presentes em 'PerfilRoute.js', deixando somente o restante do caminho para o recurso. O recurso '/perfil' pelo método GET, por exemplo, passará a ser assim:

```

router.get("", (req, res) => {
    res.json(perfis.length > 5 ?
    perfis.slice(perfis.length - 5) : perfis);
});

```

Finalmente, devemos exportar o 'router'.

```

module.exports = router;

```

Com isso, podemos utilizar o Thunder Client para verificar que os recursos já programados estão funcionando.