



## AULA 05 – Vídeo 04

### Desenvolvimento

Nesta aula, implementaremos a *cama services*.

Para começarmos, criamos um novo diretório dentro de 'src' chamado 'services'. Nesse diretório, criamos um arquivo chamado 'PerfilServices.js'. Precisamos, então, trazer as funções para esse arquivo. Para isso, recortamos e colamos em 'PerfilServices.js' todas as funções de tratamento que atualmente estão em 'PerfilRoute.js'. Como utilizaremos os perfis nessas funções, também devemos jogar o *array* que os contém para 'PerfilServices.js', juntamente com a variável 'geradorID'.

Em 'PerfilServices.js', teremos as funções. Primeiramente, precisamos nomeá-las. Além disso, precisamos exportá-las para podermos, então, importá-las em 'PerfilRoute.js', para que elas possam ser utilizadas para tratamento de requisições. Para exportá-las, podemos utilizar um objeto. A função responsável pelo tratamento de requisições para '/perfil' pelo método GET, por exemplo, será nomeada como 'buscarUltimos', e ficará assim:

```
module.exports = {  
  buscarUltimos: (req, res) => {  
    res.json(perfis.length > 5 ? perfis.slice(perfis.length  
- 5) : perfis);  
  }  
}
```

Para que as funções possam ser utilizadas em 'PerfilRoute.js', precisamos importá-las. Podemos fazê-lo com a seguinte linha de código:

```
const perfilService = require("../services/PerfilServices");
```

Com isso, todas as funções ficarão disponíveis na constante 'perfilService'. Podemos, então, utilizá-la para tratar requisições:

```
router.get("/", perfilService.buscarUltimos);
```

Agora, repetimos o mesmo procedimento para as demais funções. Podemos utilizar o mesmo objeto que já está sendo exportado, bastando separar as funções por vírgula. As rotas ficarão da seguinte forma:

```
router.get("", perfilService.buscarUltimos);  
router.get("/:id", perfilService.buscarPorId);  
router.post("", perfilService.cadastrar);  
router.put("/:id", perfilService.editar);  
router.post("/conexao", perfilService.conectar);
```