



AULA 05 – Vídeo 03

Desenvolvimento

No momento, o arquivo que contém nossas rotas de perfil ainda está abarrotado, pois estamos executando os tratamentos no mesmo local em que definimos as rotas. O próximo passo na organização de nossos arquivos é separar os tratamentos das definições. Para isso, precisaremos criar uma nova camada.

Como sabemos, clientes fazem requisições para o servidor e recebem respostas, enviadas a eles pela aplicação *back-end*. Nossa aplicação roda na porta 3000. Cada vez que uma requisição HTTP é feita a essa porta, a requisição é direcionada para uma rota. O *express*, que tratará essa requisição, possui um roteador, uma lista de rotas disponíveis. O arquivo 'index.js' é o ponto de partida da aplicação, que ouve a porta 3000. O arquivo 'server.js' contém a configuração do servidor com a rota raiz e, por enquanto, as rotas de perfil.

Na aplicação *back-end*, a primeira camada na instanciação do *express* é a camada de rotas. Ela contém um arquivo que contém todas as rotas que podem ser tratadas. Todos os arquivos no diretório 'routes', responsável pelas rotas, integram a camada de rotas.

Após a camada de rotas, criaremos a camada *service*. Criaremos um novo diretório para ela. A função da camada *service* é conter as funções ou métodos responsáveis pelo tratamento das requisições. Com isso, as rotas serão, de fato, meras rotas, e toda a lógica de tratamento das requisições ficará na camada *service*. O termo "*service*" geralmente indica uma camada de tratamento de regras de negócio específicas à empresa que utiliza o sistema.

Após a camada *service*, pode-se ter a camada *controllers*, que executa ações depois da aplicação de regras de negócio. No entanto, se não houver muitas regras de negócio, a camada *service* pode acabar sem uso. Nesse caso, em nível de arquitetura de projeto, pode-se tomar a decisão de suprimir uma das camadas, conforme o caso.

Em nosso projeto, faremos somente a camada *service*, que ficará responsável, portanto, por executar as ações, trabalho que geralmente caberia à camada *controllers*.

Outra camada seria a camada *repository*, que tem acesso direto à estrutura de dados e que a manipula.

Em alguns casos, há também a camada de *models*, que contém as definições das entidades.

A estruturação em camadas é uma forma de bem organizar o código, o que traz várias vantagens. Uma delas é facilitar a localização de problemas, visto que, separando-se rotas, tratamentos e manipulação de estruturas de dados, erros relativos a cada um dos níveis ocorrerão na camada pertinente, e não em outra.