



## AULA 05 – Vídeo 05

### Desenvolvimento

Para adicionarmos novas entidades em nosso código, precisamos criar um arquivo na camada de rotas, e outro na camada de *services*.

Agora, começaremos a implementar as notificações.

No diretório de rotas, criaremos um arquivo chamado 'NotificacaoRoute.js'.

No diretório de *services*, criaremos um arquivo chamado 'NotificacaoService.js'.

No arquivo 'server.js', precisamos registrar a nova rota. Para isso, devemos inserir as seguintes duas linhas de código, seguindo nosso padrão de documentação:

```
const notificacaoRoute = require("../routes/NotificacaoRoute");  
api.use('/notificacao', notificacaoRoute);
```

O arquivo de rotas ficará semelhante ao arquivo de rotas para '/perfil', com algumas alterações. Notavelmente, devemos indicar o arquivo correto da camada de *services*. Além disso, teremos apenas quatro rotas, como definido na documentação.

Nosso arquivo de rotas ficará assim:

```
const express = require('express');  
const router = express.Router();  
const notificacaoService = require("../services/NotificacaoService");  
router.get("/:id", notificacaoService.buscarPorId);
```

```

router.get("/perfil/:id",
notificacaoService.buscarPorPerfilId);
router.post("", notificacaoService.cadastrar);
router.put("/lida/:id", notificacaoService.marcarLida);

module.exports = router;

```

O próximo passo é configurar a camada de *services*. Assim como fizemos com os perfis, como ainda não estamos trabalhando com banco de dados, precisamos de um *array* com notificações. Cada notificação terá um ID, um tipo, um título, uma descrição, o status de lida, e os IDs de remetente e destinatário. Como as notificações têm IDs, também é necessária a variável 'geradorID', que será inicializada em 4, pois temos três notificações. Você encontra o array de notificações em anexo, ao fim deste documento.

Em 'NotificacaoService.js', precisamos excluir as funções não utilizamos e declarar 'buscarPorPerfilId' e 'marcarLida'. Podemos manter 'buscarPorId' e 'cadastrar', pois iremos modificá-las e utilizá-las. Quanto às novas, por enquanto, basta isto:

```

buscarPorPerfilId: (req, res) => {},
marcarLida: (req, res) => {},

```

Nosso primeiro *endpoint* busca uma notificação por ID. Nossa função fica assim:

```

buscarPorId: (req, res) => {
    let notificacaoID = req.params.id;
    let notificacaoEncontrada =
notificacoes.find((notificacao) => notificacao.id ==
notificacaoID);

    if (notificacaoEncontrada)
res.json(notificacaoEncontrada);
    else res.json({
        message: "Erro ao buscar notificação : Objeto não
encontrado"
    })
},

```

Trata-se da mesma função implementada para busca de um perfil por ID, feitas as devidas alterações para "notificação". A lógica é exatamente a mesma. O teste pode ser feito por meio do Thunder Client.

## ANEXO.

```
let notificacoes = [  
  {  
    id: 1,  
    tipo: "Contato",  
    titulo: "Notificação 1",  
    descricao: "Esta é a notificação de número 1",  
    lida: false,  
    remetente: 2,  
    destinatario: 3  
  },  
  {  
    id: 2,  
    tipo: "Solicitação de amizade",  
    titulo: "Notificação 2",  
    descricao: "Esta é a notificação de número 2",  
    lida: false,  
    remetente: 3,  
    destinatario: 4  
  },  
  {  
    id: 3,  
    tipo: "Contato",  
    titulo: "Notificação 3",  
    descricao: "Esta é a notificação de número 3",  
    lida: false,  
    remetente: 4,  
    destinatario: 5  
  },  
]
```

```
];
```