



AULA 04 – Vídeo 07

Conclusão

Nesta aula, implementaremos o último recurso necessário, aquele responsável pela conexão entre perfis. A requisição será feita pelo método POST para '/perfil/conexao'. O objeto enviado na requisição conterá o ID do perfil remetente e o ID do perfil destinatário. Devemos nos lembrar que em cada perfil há um *array* de conexões, que deverá conter os IDs dos perfis com os quais o perfil tem conexões. Ao final, nosso código ficará assim:

```
app.post("/perfil/conexao", (req, res) => {
  let info = req.body;

  if (info.remetente && info.destinatario) {
    let remetenteID = info.remetente;
    let destinatarioID = info.destinatario;

    let remetente = perfis.find(perfil => perfil.id ==
remetenteID);
    let destinatario = perfis.find(perfil => perfil.id ==
destinatarioID);

    if (remetente && destinatario) {
      remetente.conexoes.push(destinatarioID);
      destinatario.conexoes.push(remetenteID);

      res.json({
```

```

        message: "Conexão estabelecida com sucesso"
    });
} else {
    res.json({
        message: "Erro ao definir conexão : Perfil não
encontrado"
    });
}
} else {
    res.status(400).json({
        message: "Erro ao editar perfil : Dados
incompletos"
    });
}
});

```

Em primeiro lugar, colocamos o corpo da requisição na variável 'info'. Devemos testar se o corpo da requisição foi preenchido. Podemos fazê-lo com uma decisão condicional, verificando se ambos, remetente e destinatário, existem. Em seguida, caso o corpo da requisição tenha sido enviado corretamente, extraímos os IDs para as variáveis 'remetenteID' e 'destinatarioID':

```

let info = req.body;

if (info.remetente && info.destinatario
) {
    let remetenteID = info.remetente;
    let destinatarioID = info.destinatario;
}

```

Se os dados não forem passados, podemos enviar o status de erro 400 com uma mensagem indicando dados incompletos:

```

} else {
    res.status(400).json({

```

```

        message: "Erro ao editar perfil : Dados
incompletos"
    });
}

```

De posse dos IDs, devemos acrescentar os dados de conexões aos perfis. Para isso, o primeiro passo é garantir que os IDs passados se referem a perfis existentes. Outros tratamentos seriam necessários (por exemplo, o ID remetente não poderia ser igual ao ID destinatário), mas não o faremos neste momento.

Para verificarmos a referência dos IDs, podemos utilizar o método 'find()'. Colocamos a informação retornada por 'find()' nas variáveis 'remetente' e 'destinatario':

```

    let remetente = perfis.find(perfil => perfil.id ==
remetenteID);
    let destinatario = perfis.find(perfil => perfil.id ==
destinatarioID);

```

Quando é passado um ID correto, o método 'find()' retorna o perfil vinculado ao ID. Quando a requisição contém um ID inexistente, o método retorna 'undefined'. Podemos, portanto, utilizar uma decisão condicional para verificar se foram encontrados os perfis de remetente e destinatário. Caso algum deles não tenha sido encontrado, a resposta manterá o código 200, mas será uma mensagem de erro. Caso os dois perfis tenham sido localizados, devemos alterar os *arrays* de conexões contidos neles e enviar uma mensagem de sucesso:

```

if (remetente && destinatario) {
    remetente.conexoes.push(destinatarioID);
    destinatario.conexoes.push(remetenteID);

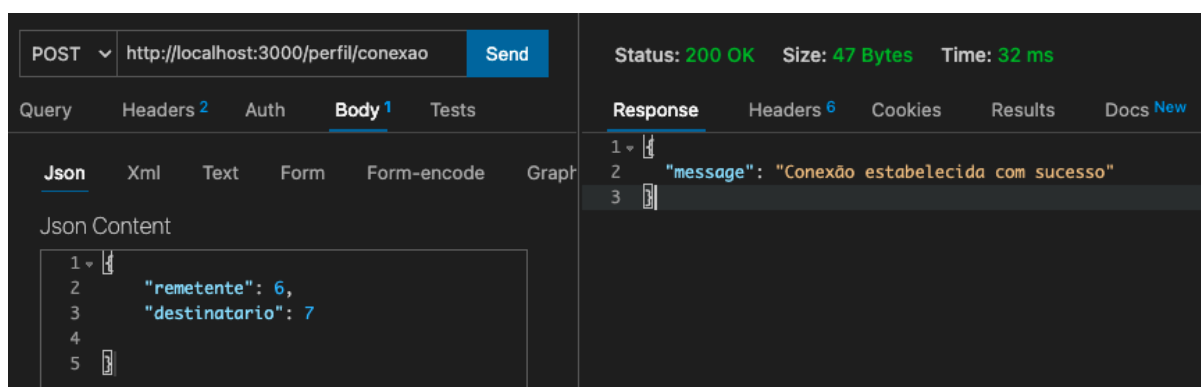
    res.json({
        message: "Conexão estabelecida com sucesso"
    });
} else {
    res.json({
        message: "Erro ao definir conexão : Perfil não
encontrado"
    });
}

```

```
    }) ;  
  }  
}
```

Feita essa implementação, podemos utilizar o Thunder Client para testar o recurso. Se a implementação estiver correta, podemos fazer uma requisição para '/perfil/conexao' passando os IDs de dois perfis a serem conectados. Também podemos fazer outra requisição para '/perfil' pelo método GET, para verificar se os *arrays* de conexão dos perfis foram atualizados.

No primeiro caso, o resultado será algo como isto:



Finalmente, podemos ver os perfis atualizados:



Uma última observação. As variáveis 'remetente' e 'destinatario' contêm aquilo que foi retornado pela função 'find()'. Somente podemos atribuir os IDs aos *arrays* de conexões presentes nelas porque a função 'find()' retorna uma referência aos perfis, um ponteiro, e não uma cópia.