



AULA 02 - Vídeo 04

Conclusão

Agora que já falamos das telas, das entidades, vamos falar sobre as funcionalidades/requisições HTTP que o sistema terá. Pra facilitar, primeiramente vamos criar um arquivo chamado endpoints.txt no VSCode.

Após isso, iremos começar com o **usuário**, posto que, sem ele, nós não temos um sistema funcional, as funcionalidades do mesmo será:

Usuário:

1. /login

Método: POST

Path: /login

Corpo da Requisição:

```
{  
  email:"jao@gmail.com",  
  senha:"123"  
}
```

Resposta:

```
{  
  _id: ObjectId(*id*),  
  perfil: ObjectId (*id*),  
  email: jao@gmail.com,  
  token: "..."  
}
```

Feito isso, vamos estar atribuindo funcionalidades ao **perfil do usuário**:

Perfil:

/perfil

Realiza a busca dos últimos 5 perfis cadastrados.

Método: GET

Path: /perfil

Resposta:

```
{
  nome: "Jão da Silva",
  dataNascimento: 2000-12-08,
  skills: ["Proativo", "JavaScript", "VueJs", "NodeJs"],
  disponibilidadeDeHorario: "Integral",
  disponibilidadeMudanca: true,
},
educacao: [{
  instituicao: "E.M. Doutor Zebrinha",
  ingresso: 2009-03-10,
  conclusao: 2014-12-10,
  nivelEscolaridade: "Ensino Fundamental"
}],
certificacoes: [{
  instituicao: "High Tech Cursos",
  titulo: "Curso Fabrica De Programador",
  cargaHoraria: 80
}],
conexoes: [
  ObjectID ("id"),
  ObjectID ("id"),
  ObjectID ("id"),
]
```

/perfil

Realiza o cadastro do perfil e o cadastro do usuário associado.

O usuário cadastrado é uma entidade a parte do perfil, sendo que o usuário mantém a referência para o perfil. Após o cadastro do perfil/usuário será feito o login automaticamente.

Método: POST

Path: /perfil

Corpo da Requisição:

```
{
  usuario: {
    email: "jao@gmail.com",
    senha: "123"
  },
  {
    nome: "Jão da Silva",
```

```

dataNascimento: 2000-12-08,
skills: ["Proativo", "JavaScript", "VueJs", "NodeJs"],
disponibilidadeDeHorario: "Integral",
disponibilidadeMudanca: true,
},
educacao: {
instituicao: "E.M. Doutor Zebrinha",
ingresso: 2009-03-10,
conclusao: 2014-12-10,
nivelEscolaridade: "Ensino Fundamental"
},
certificacoes: {
instituicao: "High Tech Cursos",
titulo: "Curso Fabrica De Programador",
cargaHoraria: 80
}
}

```

Resposta:

Retorna o ID do usuário cadastrado

```

{
_id: ObjectID(id),
perfil : ObjectID(id),
email: "jao@gmail.com",
token: "..."
}

```

/perfil <id>

Realiza a busca do perfil por id.

Método: GET

Path Param: id do perfil

Cabeçalho da Requisição:

x-access-token: token

Resposta:

```

[{
  nome: "Jão da Silva",
  dataNascimento: 2000-12-08,
  skills: ["Proativo", "JavaScript", "VueJs", "NodeJs"],
  disponibilidadeDeHorario: "Integral",
  disponibilidadeMudanca: true,
},
educacao: {
instituicao: "E.M. Doutor Zebrinha",
ingresso: 2009-03-10,
conclusao: 2014-12-10,
nivelEscolaridade: "Ensino Fundamental"
},
certificacoes: {

```

```
    instituicao: "High Tech Cursos",
    titulo: "Curso Fabrica De Programador",
    cargaHoraria: 80
  }
}
```

/perfil/<id>

Realiza a atualização do perfil por id

```
Método: PUT
Path: /perfil/<id>
Path Param: id do perfil
Cabeçalho da Requisição:
x-acess-token: token
Resposta:
[{
  nome: "Jão da Silva",
  dataNascimento: 2000-12-08,
  skills: ["Proativo", "JavaScript", "VueJs", "NodeJs"],
  disponibilidadeDeHorario: "Integral",
  disponibilidadeMudanca: true,
},
educacao: {
  instituicao: "E.M. Doutor Zebrinha",
  ingresso: 2009-03-10,
  conclusao: 2014-12-10,
  nivelEscolaridade: "Ensino Fundamental"
},
certificacoes: {
  instituicao: "High Tech Cursos",
  titulo: "Curso Fabrica De Programador",
  cargaHoraria: 80
}
}]
```

Agora, vamos estar atribuindo a funcionalidade **buscar informações**:

Busca

/busca

Realiza a busca filtrada utilizando as informações da queryString.

Ex: /busca?skills=NodeJs, VueJs, Proativo&nome=Jão%20da%20Silva

Método: GET

Path: /busca

Query String:

skills: String das skills de filtragem, separadas por vírgula

nome: String do nome para filtragem

Resposta:

```
{
  nome: "Jão da Silva",
  dataNascimento: 2000-12-08,
  skills: ["Proativo", "JavaScript", "VueJs", "NodeJs"],
  disponibilidadeDeHorario: "Integral",
  disponibilidadeMudanca: true,
},
educacao: {
  instituicao: "E.M. Doutor Zebrinha",
  ingresso: 2009-03-10,
  conclusao: 2014-12-10,
  nivelEscolaridade: "Ensino Fundamental"
},
certificacoes: {
  instituicao: "High Tech Cursos",
  titulo: "Curso Fabrica De Programador",
  cargaHoraria: 80
}
}
```

Por fim, vamos estar atribuindo a funcionalidade **notificações**:
Notificações

/notificacao

Realiza o cadastro de uma notificação.

Método: POST

Path: /notificacao

Cabeçalho da Requisição:

x-access-token: token

Corpo da Requisição:

```
{
  tipo: "Solicitação de Amizade",
  titulo: "Solicitação de Amizade",
  descrição: "Olá Fulano, gostaria de fazer parte do meu network?",
  destinatario: ObjectID(id),
  remetente ObjectID(id),
}
```

/notificacao/lida/<id>

Realiza a edição da notificação indicando que a mesma foi lida.

Método: PUT

Path: /notificacao/lida/<id>

Cabeçalho da Requisição:

x-access-token: token

Resposta:

```
{
  tipo: "Solicitação de Amizade",
  titulo: "Solicitação de Amizade",
  descrição: "Olá Fulano, gostaria de fazer parte do meu network?",
  destinatario: ObjectID(id),
  remetente ObjectID(id),
  lida:true
}
```