Atividade 2

Node.js, Express.js e Mongoose

REST

Universidade Federal do Ceará
Projeto de Interfaces Web
Prof. Victor Farias

- **Data de entrega**: 11/01/21
- **Peso 2**
- **Modo de entrega**
  - Código + vídeo de apresentação zipado pelo moodle
  - O vídeo também pode ser subido no YouTube e envia o link no arquivo zip
- **Instruções para o código**
  - Excluir node_modules e adicionar os outros arquivos ao zip final
- **Instruções para o vídeo de apresentação:**
  - Somente as funcionalidades que foram mostradas no vídeo receberam nota
  - Não precisa falar, basta gravar a tela
    1. Lembrar de usar uma qualidade um pouco mais baixa, caso envie o vídeo pelo moodle, pois o moodle só aceita arquivos de até 20mb
    2. Pode usar algum software como o obs ou gravar a tela do computador ou celular
  - O objetivo é demonstrar as funcionalidades pedidas usando o postman ou software similar.
  - **Roteiro do vídeo**

    Montando o banco

    1. Inserir um primeiro usuário pelo endpoint **POST** /api/usuários

2. Inserir um post para esse usuário (usar o id do usuário que o banco gerou automaticamente) com **POST** /api/posts
3. Inserir um segundo usuário com **POST** /api/usuários
4. Inserir um post para o segundo usuário **POST** /api/posts
5. Adicionar um comentário desse segundo usuário no post feito pelo primeiro usuário usando **POST** /api/comentarios

   Fazendo buscas
6. Buscar todos os usuários com **GET /**api/usuarios
7. Buscar apenas o primeiro usuário pelo id com **GET /**api/usuarios/<id_primeiro_usuario>
8. Buscar todos os posts do primeiro usuário com **GET /**api/usuarios/<id_primeiro_usuario>/posts
9. Buscar todos os posts com **GET /**api/posts
10. Buscar apenas o post do primeiro usuário com **GET /**api/posts/<id_post> (pegar id na mão)
11. Buscar todos os comentários do post do primeiro usuário com **GET** /api/posts/<id_post>/comentarios
12. Buscar todos os comentários com **GET** /api/comentarios
13. Mostrar as 3 coleções no MongoDB usando o MongoShell ou alguma aplicação parecida

    Removendo dados
14. Remover o comentário feito usando **DELETE** /api/comentarios/<id_comentario>
15. Remover o post do primeiro usuário como **DELETE** /api/posts/<id_post>
16. Remover primeiro usuário com **DELETE** /api/usuarios/<id_usuario>
17. Mostrar as coleções com os dados removidos

- Dica: Deixa todas as requisições prontas no postman e, depois, é só gravar executando-as.
- Tempo máximo do vídeo: 5 minutos

Questões

1. Crie os modelos segundo o seguinte esquema (1 ponto)
   Usuário (id, nome, email, senha)
   Post (id, texto, likes, id_usuario)
   Comentário (id, texto, id_post, id_usuario)
2. Crie as views para renomear _id para id em todas as entidades e para evitar de enviar senha do usuário para o cliente (1 ponto)

3. **Endpoints.** Construa os endpoints que se recebe, respeitando **exatamente** as estruturas do JSON de exemplo e **exatamente** nessas rotas. Agora os dados serão armazenados no MongoDB.

   Usuários

   a. **POST** /api/usuarios (0,5 ponto)
      i. Recebe usuário e armazena em banco
      ii. Exemplo do json de requisição:

```
{
   “id”: “1”,
   “nome”: “Victor”,
   “email”: “victor.aefarias@gmail.com”,
   “senha”: “123”
}
```

      iii. Retorna o mesmo usuário

   b. **GET** /api/usuarios (0,5 pontos)
      i. Retorna todos os usuários
      ii. Exemplo do json de resposta:

```
[
   {
      “id”: “1”,
      “nome”: “Victor”,
      “email”: “victor.aefarias@gmail.com”,
      “senha”: “123”
   },
   {
      “id”: “5”,
      “nome”: “João”,
      “email”: “joao @gmail.com”,
      “senha”: “456”
```

},
...
]

c. **GET** /api/usuarios/:id (0,5 ponto)
   i. Retorna usuário com um dado id
   ii. Exemplo json de resposta para /api/usuarios/1:

```
{
   “id”: “1”,
   “nome”: “Victor”,
   “email”: “victor.aefarias@gmail.com”,
   “senha”: “123”
}
```

   iii. Caso usuário não exista, retorna erro 404

d. **GET** /api/usuarios/:id/posts (1 ponto)
   i. Retorna todos os posts do usuário com o id dado.
   ii. Exemplo json de resposta /api/usuarios/1/posts:

```
[
   {
      “id”: “1”,
      “texto”: “Oi, tudo bem?”,
      “likes”: “6”,
      “id_usuario”: 1

   },
   {
      “id”: “8”,
      “texto”: “Olá?”,
      “likes”: “8”,
      “id_usuario”: 1
   },
   ...
]
```

e. **DELETE** /api/usuarios/:id (0,5 ponto)
   i. Remove usuário com id dado

Posts

f. **POST** /api/posts (0,5 ponto)

i. Recebe post e armazena em bancp
ii. Exemplo json da requisição:

```
{
   “id”: “1”,
   “texto”: “Oi, tudo bem?”,
   “likes”: “6”,
   “id_usuario”: 1
}
```

iii. Retorna o mesmo post.

g. **GET** /api/posts (0,5 ponto)
   i. Retorna todos os posts
   ii. Exemplo json de resposta:

```
[
   {
      “id”: “1”,
      “texto”: “Oi, tudo bem?”,
      “likes”: “6”,
      “id_usuario”: 1

   },
   {
      “id”: “5”,
      “texto”: “Olá amigos!”,
      “likes”: “4”,
      “id_usuario”: 2
   },
   ...
]
```

h. **GET** /api/posts/:id (0,5 ponto)
   i. Retorna post com um dado id
   ii. Exemplo json de retorno para /api/posts/1

```
{
   “id”: “1”,
   “texto”: “Oi, tudo bem?”,
   “likes”: “6”
}
```

i. **GET** /api/posts/:id/comentarios (1 ponto)

i. Retorna todos os comentários do post com id dado
ii. Exemplo json de resposta para /api/posts/1/comentarios

```
[
  {
    “id”:6,
    “texto”:”Tudo certo e contigo?”,
    “id_post”: 1,
    “id_usuario”: 2
  },
  {
    “id”:6,
    “texto”: ”Tudo blz ma!”,
    “id_post”: 1,
    “id_usuario”: 3
  }
  ...
]
```

j. **DELETE** /api/posts/:id (0,5 ponto)
   i. Remove post com id dado

Comentarios

k. **POST /**api/comentarios (1 ponto)
   i. Recebe comentário e armazena em banco
   ii. Exemplo json de requisição:

```
{
    “id”:6,
    “texto”:”Tudo certo e contigo?”,
    “id_post”: 1,
    “id_usuario”: 2
}
```

l. **GET /**api/comentarios (0,5 ponto)
   i. Retorna todos os posts
   ii. Exemplo json de retorno:

```
[
  {
    “id”:6,
    “texto”:”Ótimo, amiga!”,
```

```
        “id_post”: 1,
        “id_usuario”: 2
    },
    {
        “id”:6,
        “texto”: ”Casal lindo!”,
        “id_post”: 2,
        “id_usuario”: 1
    }
    ...
]
```


m. **DELETE** /api/comentarios/:id (0,5 ponto)
    i. Exclui comentário com id dado.