**Tabelas criadas com “migration”**

NAVERS

|  |  |  |  |
| --- | --- | --- | --- |
| Coluna | Tipo | Tamanho | Observações |
| <id> | Chave Primária |  |  |
| <name> | CharField | 48 |  |
| <birthdate> | DateTimeField |  |  |
| <admission\_date> | DateTimeField |  |  |
| <job\_role> | CharField | 48 |  |
| <user\_id> | IntegerField |  | Preenchido automaticamente conforme usuário autenciado |

PROJETOS

|  |  |  |  |
| --- | --- | --- | --- |
| Coluna | Tipo | Tamanho | Observações |
| <id> | Chave Primária |  |  |
| <name> | CharField | 48 |  |
| <user\_id> | IntegerField |  | Preenchido automaticamente conforme usuário autenciado |

NAVERPROJETOS

|  |  |  |  |
| --- | --- | --- | --- |
| Coluna | Tipo | Tamanho | Observações |
| <id> | Chave Primária |  |  |
| <naver\_id> | ForeignKey |  | Tabela Navers |
| <projeto\_id> | ForeignKey |  | Tabela Projetos |

**Rotas Disponíveis**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Rota | Descrição | Parâmetros |
| GET | /admin/ | Acessar módulo administrativo |  |
| POST | /login/ | Fazer login de usuário existente e receber o token de acesso do mesmo | POST “username” e “password” |
| POST | /signup/ | Criar um novo usuário | POST “username” e “password” |
| GET | /navers/ | Listar todos os “navers” e seus respectivos projetos | Header “Authentication” conforme recebido no login |
| GET | /navers/<id>/ | Listar um “naver” específico por ID | Header “Authentication” conforme recebido no login |
| POST | /navers/ | Criar um “naver” | Header “Authentication” conforme recebido no login e dados POST “name”, “birthdate” (data), “admission\_date” (data) e “job\_role” |
| PUT | /navers/<id>/ | Atualizar os dados de um “naver” | Header “Authentication” conforme recebido no login e dados POST (todos opcionais) “name”, “birthdate” (data), “admission\_date” (data) e “job\_role” |
| DELETE | /navers/<id>/ | Excluir um “naver” | Header “Authentication” conforme recebido no login |
| GET | /projetos/ | Listar projetos e “navers” que participam de cada um | Header “Authentication” conforme recebido no login |
| POST | /projetos/ | Criar um novo projeto | Header “Authentication” conforme recebido no login e dado POST “name” (nome do projeto) |
| PUT | /projetos/<id>/ | Atualizar os dados de um projeto | Header “Authentication” conforme recebido no login e dado POST “name” |
| GET | /projetos/<id>/ | Listar dados de um projeto específico por ID | Header “Authentication” conforme recebido no login |
| DELETE | /projetos/<id>/ | Excluir um projeto | Header “Authentication” conforme recebido no login |
| POST | /navers/add-projeto/ | Adicionar um naver a um projeto | Header “Authentication” conforme recebido no login e dados POST “naver” (id do naver) e “projeto” (id do projeto) |

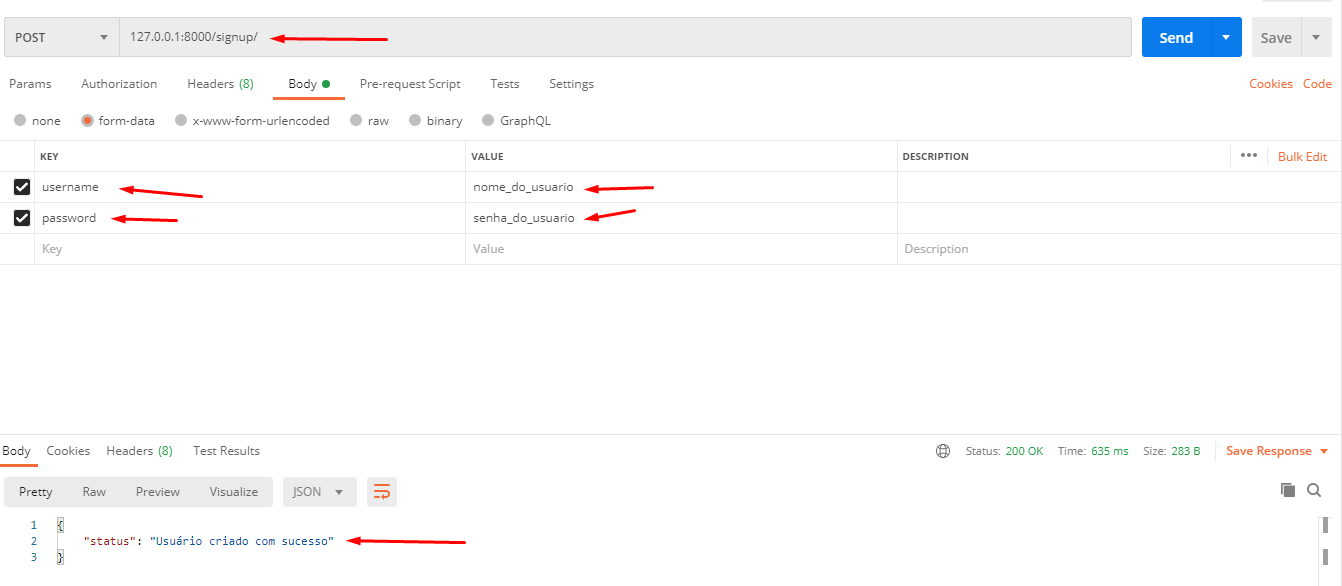
**Instruções para testes com POSTMAN**

Uma vez clonado o projeto e realizadas as migrações, subir o servidor da forma padrão para poder começar a realizar os testes no POSTMAN.

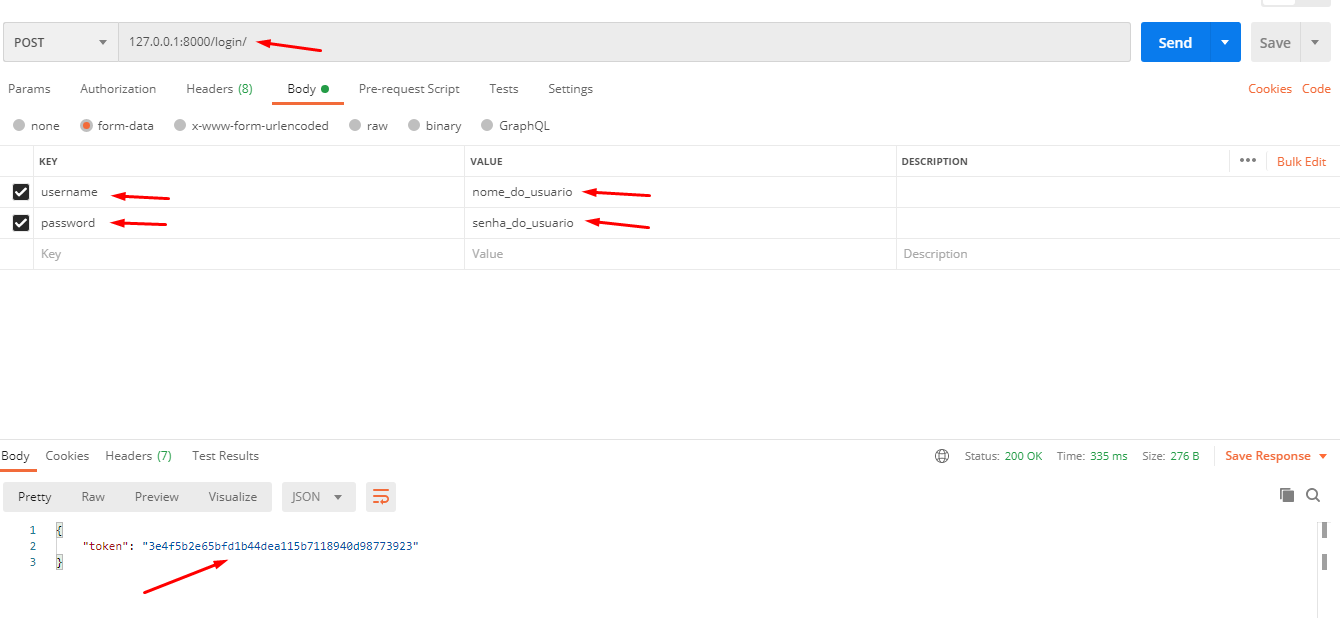
Comando: python manage.py runserver

OBS: Por padrão, a porta de acesso à API será 8000

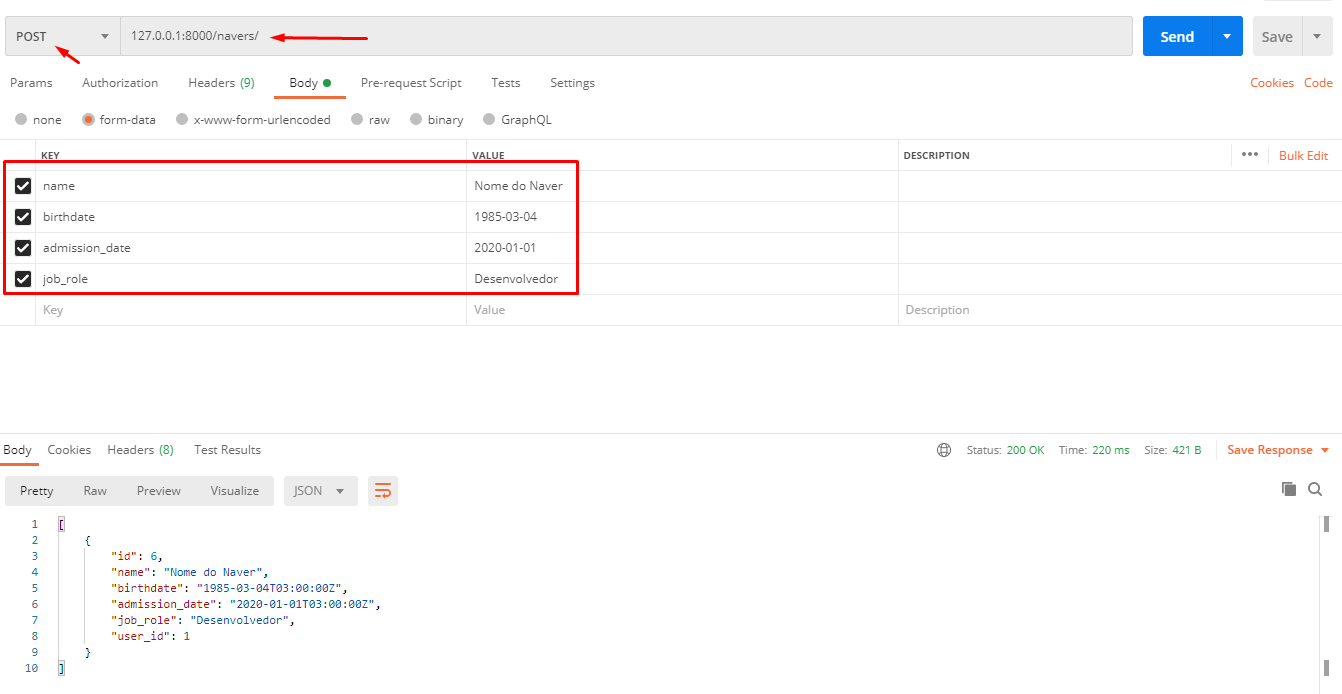
Para criar um novo usuário no POSTMAN:



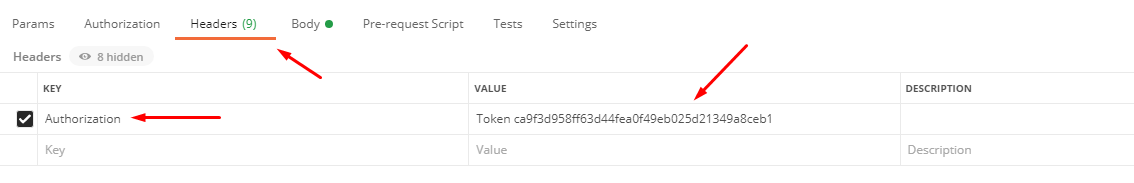
Para fazer login com o usuário criado e, assim, receber seu token de acesso:



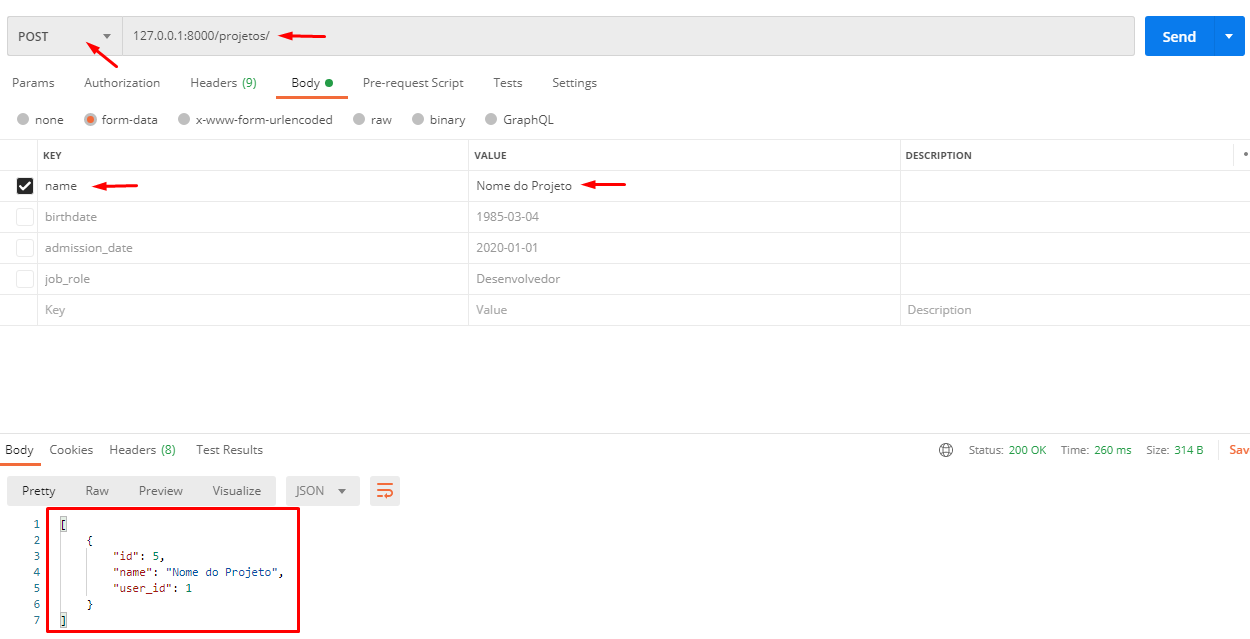
Para criar um novo “naver” (vinculado ao usuário conectado)



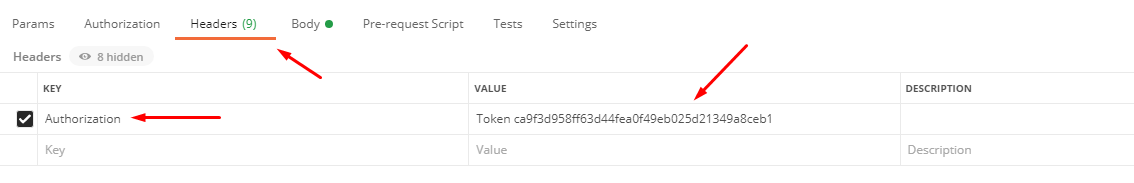
OBS: Não esquecer de fornecer o token de autenticação recebido no login



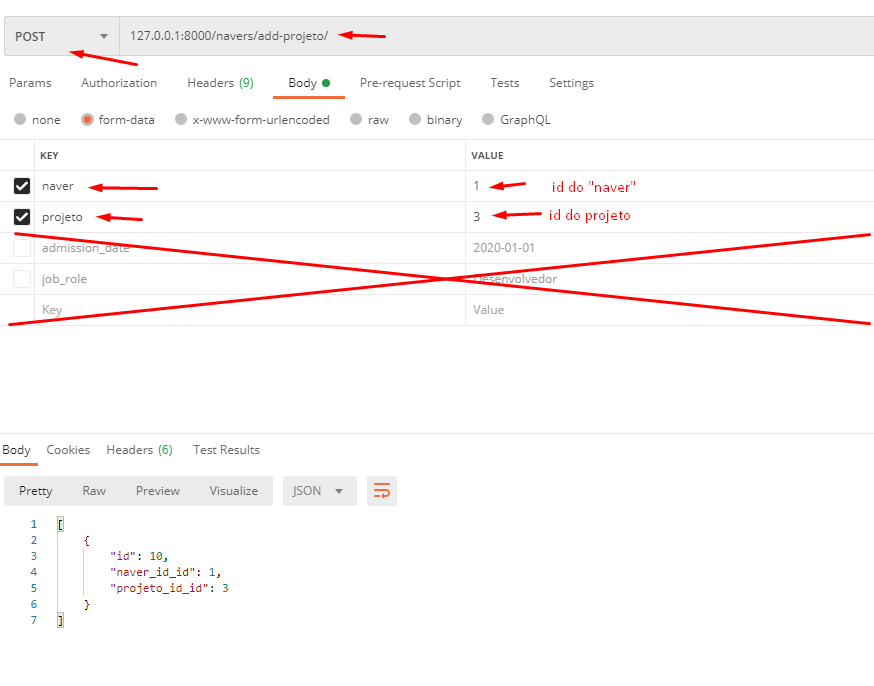
Para criar um novo projeto:



OBS: Não esquecer de fornecer o token de autenticação recebido no login



Para vincular um naver a um projeto:



OBS: Não esquecer de fornecer o token de autenticação recebido no login

