# 1 Exercício

Neste exercício, você criará um endpoint para redefinição de senha. Para acessá-lo, o aplicativo cliente enviará um objeto JSON com um campo de email apenas. O servidor deve montar uma URL que, quando acessada, permite a redefinição de senha. A seguir, ele envia um email para o usuário, instruindo-o a utilizar o link para reconfigurar a sua senha. Para isso, utilize as seguintes dicas.

- Faça as seguintes configurações no settings.py.

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = 'seu_email@gmail.com'
EMAIL_HOST_PASSWORD = 'sua_senha'
```

**Dica**: Se desejar, utilize seu Gmail. Para tal, visite o link

https://myaccount.google.com/apppasswords

e gere uma senha de aplicativo. Assim você não utilizará a sua senha pessoal. A senha gerada deve ser usada normalmente, onde você usaria a sua senha.

- Crie uma classe de modelo PasswordResetToken com

user (ForeignKey apontando para User de django.contrib.auth.models)
token (UUIDField, importe o pacote uuid para gerar)
created_at (DateTimeField)

Crie um arquivo para views chamado **reset_password_views.py**. Implemente o seguinte Endpoint:

```
@api_view(['POST'])
def request_password_reset(request):
```

Ele deve:

- Obter o email da request

- Verificar se, na base, há um usuário com aquele e-mail cadastrado (User.objects.get). Devolver 404 se não existir.

- Construir um PasswordResetToken vinculado ao usuário (reset_token = PasswordResetToken(user=user)) e salvar no banco com **save**.

- Montar um link para reset de password:

reset_link = f"https://localhost:8000/reset-password/{reset_token.token}"

- Enviar o e-mail com a função send_mail (django.core.mail import send_mail)

https://docs.djangoproject.com/en/3.2/topics/email/#send-mail

- Devolver 200 e uma mensagem dizendo que o e-mail foi enviado com sucesso.

- Ainda no arquivo reset_password_views, crie outro endpoint:

@api_view(['POST'])
def reset_password(request, token):

Ele deve

- Verificar se o token existe (reset_token = PasswordResetToken.objects.get(token=token)). Se não existir, devolver 400.

- Da requisição, pegar a senha do JSON enviado pelo cliente.

{“password”: “nova-senha” }

- Configurar a nova senha do usuário

reset_token.user.set_password(new_password)
reset_token.user.save()

Apagar o token

reset_token.delete()

Devolver 200.

- Configure as novas URLs.

```
path('request-password-reset/', request_password_reset, name='request_password_reset'),
    path('reset-password/<uuid:token>/', reset_password, name='reset_password'),
]
```

- Envie novas requisições POST a fim de

- Obter novo token para nova senha
- Configurar nova senha (token como path e password no corpo da requisição)