

MFA/2FA



Sommaire

- Pourquoi la choisir?
- Les différentes solutions
- Un exemple simple

1

Pourquoi ?

173 000



19%



4 avantages

Sécurité des données

Conformité et normalisation

Outil marketing

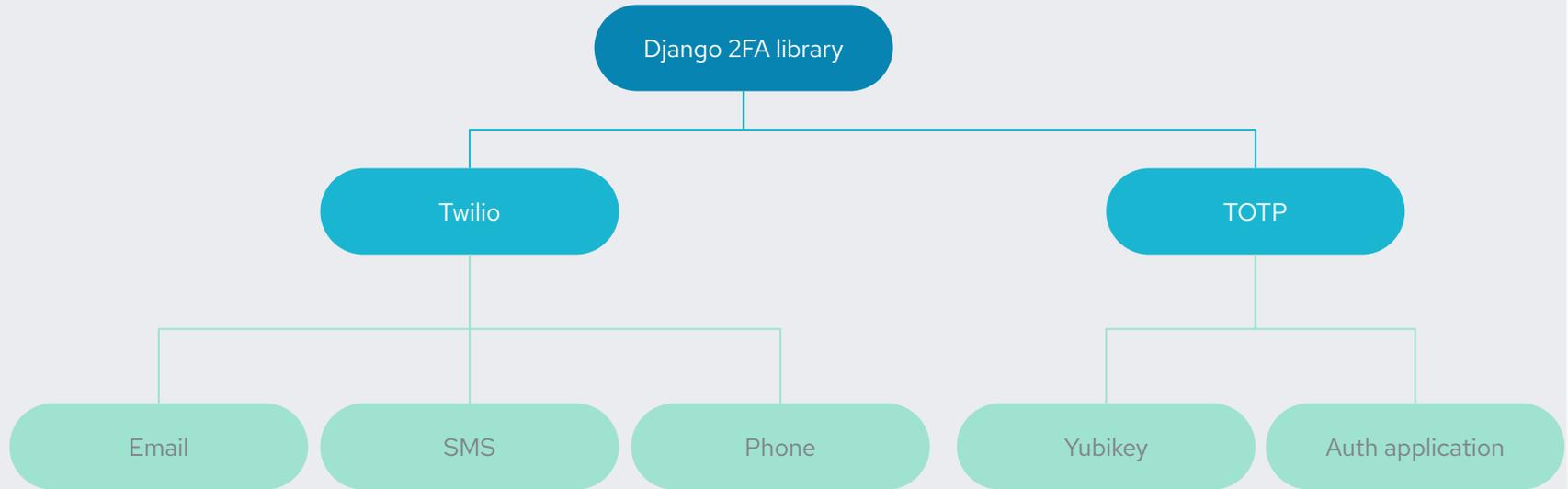
Fluidité et facilité

2

Les différentes manières de faire



Solution

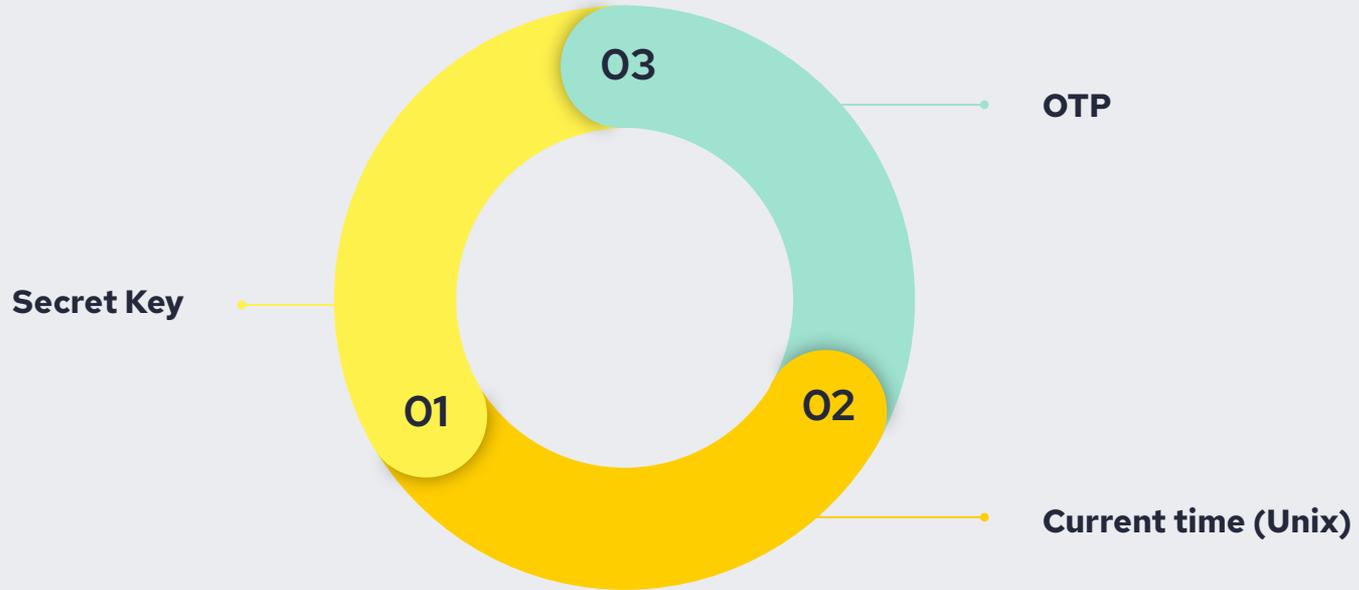


3

Un exemple de TOTP



Le Time-based One Time Password





Exemple en code

```
class UserModel(AbstractUser):
    id = models.UUIDField(primary_key=True, default=uuid.uuid4, editable=False)
    name = models.CharField(max_length=50)
    email = models.EmailField(max_length=100, unique=True)
    password = models.CharField(max_length=32)
    otp_enabled = models.BooleanField(default=False)
    otp_verified = models.BooleanField(default=False)
    otp_base32 = models.CharField(max_length=255, null=True)
    otp_auth_url = models.CharField(max_length=255, null=True)
    username = None
```

La première étape consiste à créer un modèle avec les attributs nécessaires



Exemple en code

Génération du QR Code qui contient l'URL à relier à l'authenticator (GoogleAuth par exemple)

```
class GenerateOTP(generics.GenericAPIView):
    serializer_class = UserSerializer
    queryset = UserModel.objects.all()

    def post(self, request):
        data = request.data
        user_id = data.get('user_id', None)
        email = data.get('email', None)

        user = UserModel.objects.filter(id=user_id).first()
        if user == None:
            return Response({"status": "fail", "message": f"No user with Id: {user_id} found"}, status=status.HTTP_404_NOT_FOUND)

        otp_base32 = pyotp.random_base32()
        otp_auth_url = pyotp.totp.TOTP(otp_base32).provisioning_uri(
            name=email.lower(), issuer_name="anis2FA.test")

        user.otp_auth_url = otp_auth_url
        user.otp_base32 = otp_base32
        user.save()
        qrcode.make(otp_auth_url).save("qr_auth.png")

        return Response({'base32': otp_base32, "otppath_url": otp_auth_url})
```



Exemple en code

```
POST http://127.0.0.1:8000/api/auth/otp/generate | Send | 200 OK | 53.7 ms | 167 B
JSON | Auth | Query | Headers | Docs | Preview | Headers | Cookies | Timeline
1 {
2   "user_id": "76fea22c-83ad-4751-9636-c7e30ca6a228",
3   "email": "hello@python.io"
4 }
1 {
2   "base32": "ZH2PQBYOQ3RCS5TMU4YEHELCC4CB305E",
3   "otpauth_url": "otpauth://totp/anis2FA.test:hello%40python.io?secret=ZH2PQBYOQ3RCS5TMU4YEHELCC4CB305E&issuer=anis2FA.test"
4 }
```

Exemple d'API Request afin d'obtenir l'url à ajouter dans l'application d'authentification



Exemple en code

Première request avec le token valide et la seconde 30sec plus tard qui est invalide

```
POST http://127.0.0.1:8000/api/auth/otp/verify 200 OK 9.84 ms 326 B
JSON Auth Query Headers 1 Docs Preview Headers 10 Cookies Timeline
1 {
2   "user_id": "76fea22c-83ad-4751-9636-c7e30ca6a228",
3   "token": "528625"
4 }
```

```
1 {
2   "otp_verified": true,
3   "user": {
4     "id": "76fea22c-83ad-4751-9636-c7e30ca6a228",
5     "name": "Python",
6     "email": "hello@python.io",
7     "otp_enabled": true,
8     "otp_verified": true,
9     "otp_base32": "ZH2PQBYOQ3RCS5TMU4YEHELCC4CB305E",
10    "otp_auth_url": "otppath://otp/anis2FA.test:hello%40python.io?secret=ZH2PQBYOQ3RCS5TMU4YEHELCC4CB305E&issuer=anis2FA.test"
11  }
12 }
```

```
POST http://127.0.0.1:8000/api/auth/otp/verify 400 Bad Request 6.22 ms 68 B
JSON Auth Query Headers 1 Docs Preview Headers 10 Cookies Timeline
1 {
2   "user_id": "76fea22c-83ad-4751-9636-c7e30ca6a228",
3   "token": "528625"
4 }
```

```
1 {
2   "status": "fail",
3   "message": "Token is invalid or user doesn't exist"
4 }
```

Thanks!

Sources:

- cybermalveillance.gouv.fr/
- [wpcodevo/Django_2FA_Project](https://github.com/wpcodevo/Django_2FA_Project)
- [django-two-factor-auth.readthedocs](https://django-two-factor-auth.readthedocs.io/)

