

# Authentication

\*\*\*\*\*

## - Définition

- les méthodes de l'auth
- les types d'authentification
- l'authentification en django

\*auth de base

\*auth par jeton

# définition

## **Qu'est ce que l'authentification ?**

L'authentification est le processus par lequel un système vérifie l'identité d'un utilisateur qui souhaite accéder à des ressources, des données ou des services spécifiques. Cela garantit que la personne prétendant être un utilisateur particulier est réellement cette personne.

L'authentification requiert souvent nom d'utilisateur, mot de passe, parfois méthodes biométriques, jetons ou clés pour une sécurité renforcée.

## les méthodes de l'authentification

**Nom d'utilisateur et mot de passe** : Combinaison courante, nécessitant un identifiant et un mot de passe.

**Authentification à deux facteurs** : Utilisation de deux méthodes distinctes pour vérifier l'identité, comme un mot de passe et un code envoyé sur un appareil personnel.

**Authentification multi-facteurs** : Utilisation de plusieurs méthodes pour vérifier l'identité, comme empreintes digitales, codes PIN, ou jetons d'accès.

## les méthodes de l'authentification

**Authentification par jeton** : Attribution de jetons uniques pour accéder à des systèmes ou des services.

**Biométrie** : Utilisation de caractéristiques physiques uniques, telles que les empreintes digitales, la reconnaissance faciale ou vocale.

**Certificats numériques** : Utilisation de certificats numériques pour authentifier l'utilisateur ou l'appareil.

## Quels sont les différents types d'authentification ?

**L'authentification unique** permet une seule connexion pour accéder à diverses applications, offrant aux utilisateurs un accès transparent à leurs données sans devoir ressaisir un nouveau nom d'utilisateur et mot de passe.

**L'authentification Forte** Nécessite l'enchaînement d'au moins deux facteurs d'authentification, et plus on cumule de facteurs, plus on sécurise l'accès en réduisant considérablement les risques d'usurpation d'identité.

**L'authentification Faible:** Les systèmes basés sur des identifiants et mots de passe sont souvent considérés comme peu sûrs car les mots de passe choisis par les utilisateurs sont souvent faciles à deviner. Cela expose les comptes à des risques de piratage.

# L'authentification en Django

- Authentification de base.
- Authentification par jeton.

## **Authentication de base.**

L'authentification de base en Django utilise le protocole HTTP pour vérifier les informations d'identification des utilisateurs. Elle implique l'envoi encodé du nom d'utilisateur et du mot de passe, mais est moins sécurisée que d'autres méthodes comme JWT. Elle est simple à mettre en place mais moins recommandée pour des environnements de production où la sécurité est cruciale.



## **Authentification de base.**

voici les étapes simplifiées pour mettre en place l'authentification de base en Django :

**Configuration** : Régler `django.contrib.auth.backends.ModelBackend` dans `settings.py`.

**Utilisation des vues** : Intégrer les vues Django telles que `LoginView` et `LogoutView` dans votre système.

**Gestion des utilisateurs** : Utilisez les outils Django pour créer et gérer les comptes.

## Authentification par jeton

Ce système d'authentification repose sur un schéma HTTP simple basé sur des jetons. Il convient aux configurations client-serveur comme les applications mobiles et de bureau. Pour utiliser TokenAuthentication, configurez les classes d'authentification en ajoutant TokenAuthentication et assurez-vous d'avoir ajouté 'rest\_framework.authtoken' dans votre paramètre INSTALLED\_APPS.

```
INSTALLED_APPS = [  
    'rest_framework.authtoken',  
]
```

Après avoir ajusté les paramètres, lancez `manage.py migrate`. L'application `rest_framework.authtoken` intègre des migrations pour votre base de données Django. Ensuite, générez des jetons pour les utilisateurs, soit manuellement soit via une API lors de la création des comptes, puis utilisez les vues de Django REST Framework, telles que `ObtainAuthToken`, pour gérer l'authentification basée sur ces jeton

En cas d'authentification réussie avec `TokenAuthentication`, vous obtiendrez ces informations : `Request.user` représentera un utilisateur Django sous forme d'instance `User`, et `Request.auth` représentera un Token de `rest_framework.authtoken.models`.

.







