



# **Token Authentifizierung**

im Django Restframework



# Was ist Token-Authentifizierung?

- Eine Methode zur **Authentifizierung von Benutzern** in Web-APIs.
- Jeder Benutzer erhält einen **eindeutigen Authentifizierungs-Token**.
- Der Token wird bei jeder Anfrage im **HTTP-Header** gesendet.

## Vorteile

- Einfach zu implementieren und zu verwenden.
- Sicherer als einfache Benutzername/Passwort-Authentifizierung.
- Ideal für REST APIs, bei denen keine Sessions erforderlich sind.

## Ablauf

1. Benutzer gibt Anmeldeinformationen (z. B. Benutzername und Passwort) an.
2. Bei erfolgreicher Authentifizierung wird ein Token generiert und zurückgegeben.
3. Der Benutzer sendet den Token bei folgenden Anfragen im Header mit.

# Einrichten der Token-Authentifizierung

Stellen Sie sicher, dass `rest_framework.authtoken` in `INSTALLED_APPS` enthalten ist:

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

## Token-Modell Migration

Führen Sie Migrationen aus, um das Token-Modell in der Datenbank zu erstellen:

**`python manage.py migrate`**

# obtain\_auth\_token

**obtain\_auth\_token** ist eine View in Django Rest Framework (DRF), die im Rahmen des Token-Authentifizierungssystems verwendet wird.

Diese View ermöglicht es Benutzern, ein Authentifizierungs-Token zu erhalten, indem sie ihre Benutzername- und Passwort-Credentials in einem POST-Request senden. Das Token, das von dieser Ansicht zurückgegeben wird, kann dann für die Authentifizierung bei zukünftigen HTTP-Anfragen an die API verwendet werden.

Beispiel eines Urlpatterns:

```
from rest_framework.auth_token.views import obtain_auth_token

urlpatterns = [
    ...
    path('api-token-auth/', obtain_auth_token, name='api_token_auth'),
]
```

# POST Aufruf an obtain\_auth\_token

Um einen Auth-Token zu erhalten, muss ein POST - Request an die URL gesendet werden, die in den urlpatterns festgelegt wurde:

```
POST /api-token-auth/  
{  
  "username": "benutzername",  
  "password": "passwort"  
}
```

**Die Antwort ist ein Token:**

```
{  
  "token": "1234567890abcdef "  
}
```

# token mitsenden. Curl Beispiel

Der erhaltene Token muss jetzt bei allen nötigen Requests im Anfrage-Header mitgesendet werden. Im Beispiel einer curl-Request mit dem Flag -H (Header) und dem Key-Value-paar **Authorization** und Token.

```
curl -X GET http://example.com/api/protected-resource/ -H  
"Authorization: Token 1234567890abcdef"
```