

Social Login

La funzionalità di Social login è garantita generando un token temporaneo dopo che l'utente ha effettuato il login tramite Social. Questo tipo di implementazione permette di implementare il social login con un qualsiasi captive portal che supporta il protocollo RADIUS perchè è totalmente trasparente ad esso.

Installazione di django-allauth:

```
pip install django-allauth
```

Assicurarsi che il file settings.py e come l'esempio seguente (Sarà spiegato utilizzando facebook come esempio)

```
INSTALLED_APPS = [ # ... other apps .. # apps needed for social login
'rest_framework.authtoken', 'django.contrib.sites', 'allauth', 'allauth.account',
'allauth.socialaccount', # showing facebook as an example # to configure social login with other
social networks # refer to the django-allauth documentation
'allauth.socialaccount.providers.facebook', ]
```

```
SITE_ID = 1
```

Al file urls.py va aggiunto anche il la seguente linea tra gli url

```
urlpatterns = [ # .. other urls ... url(r'^accounts/', include('allauth.urls')), ]
```

Facebook supporta sia OAuth2 che il Facebook Connect Javascript SDK.

Un vantaggio dell'SDK di Javascript potrebbe essere un'esperienza utente più ottimizzata poiché non si esce dal sito. Inoltre, non è necessario preoccuparsi di adattare la finestra di dialogo di accesso a seconda che si stia utilizzando o meno un dispositivo mobile. Tuttavia, fare affidamento su Javascript potrebbe non essere sempre la soluzione migliore.

The following Facebook settings are available:

```
SOCIALACCOUNT_PROVIDERS = { 'facebook': { 'METHOD': 'oauth2', 'SDK_URL': '//
connect.facebook.net/{locale}/sdk.js', 'SCOPE': ['email', 'public_profile', 'user_friends'],
'AUTH_PARAMS': {'auth_type': 'reauthenticate'}, 'INIT_PARAMS': {'cookie': True}, 'FIELDS': [
'id', 'email', 'name', 'first_name', 'last_name', 'verified', 'locale', 'timezone', 'link', 'gender',
'updated_time', ], 'EXCHANGE_TOKEN': True, 'LOCALE_FUNC': 'path.to.callable',
'VERIFIED_EMAIL': False, 'VERSION': 'v2.12', }}
```