# Django Allauth and Django Axes Explained for Beginners

## What is Django Allauth?

Django Allauth is like a magical toolkit that helps you handle everything related to users on your website. It helps users sign up, log in, reset their password, and even use their social media (like Google or GitHub) to log in. It's already built and safe to use in production.

## What Can Django Allauth Do?

* User Registration: Users can create accounts.
* Login and Logout: Secure login/logout with session cookies.
* Email Confirmation: Confirms email addresses to prevent spam/fake accounts.
* Password Reset: Sends a reset link to the user's email.
* Email Address Management: Users can manage multiple emails.
* Social Login: Users can log in with Google, GitHub, etc.
* Rate Limiting (Basic): Prevents brute-force attempts (combine with Axes).
* Custom Templates: Customize how login/signup pages look.
* Custom Signup Fields: Collect extra data like first name.
* Redirects: Automatically redirect after login/logout.

## How to Use Django Allauth (Step-by-Step)

* Install: pip install django-allauth
* Add to INSTALLED\_APPS in settings.py, including 'django.contrib.sites'.
* Set SITE\_ID = 1 in settings.py.
* Add AUTHENTICATION\_BACKENDS in settings.py.
* Add 'accounts/' path in your project's urls.py.
* Set options like ACCOUNT\_EMAIL\_REQUIRED and ACCOUNT\_AUTHENTICATION\_METHOD.
* Run migrations: python manage.py migrate.
* Create a superuser: python manage.py createsuperuser.
* Start server and visit /accounts/signup/ or /accounts/login/.
* For email confirmation/reset, set EMAIL\_BACKEND to console for testing or SMTP for real email.

## Customizing Templates

Create a 'templates/account/' folder. Copy default templates from the Allauth GitHub repo and customize them.

## Protecting Your App with Django Axes

Django Allauth doesn’t offer full brute-force protection. That’s where Django Axes helps. It blocks repeated failed login attempts and tells the user to try again later.

### Install and Set Up Django Axes

* Install: pip install django-axes
* Add 'axes' to INSTALLED\_APPS and MIDDLEWARE in settings.py.
* Run: python manage.py migrate
* Add AXES\_ENABLED = True in settings.py
* Optionally configure timeout: AXES\_COOLOFF\_TIME = timedelta(minutes=5)
* Login failures will be tracked and blocked automatically.

### What Happens When Blocked?

If someone tries logging in too many times with wrong details, they’ll be blocked and see a message like 'Too many login attempts. Try again in 5 minutes.'

## Can Django Send Email on Login?

Yes! You can write a custom signal using Django's user\_logged\_in signal to send an email when someone logs into their account. This helps notify the user about login activity (just like Google or Facebook does).

### Example: Send Email on Successful Login

Create a file named signals.py in your app:

from django.contrib.auth.signals import user\_logged\_in  
from django.core.mail import send\_mail  
from django.dispatch import receiver  
  
@receiver(user\_logged\_in)  
def send\_login\_notification(sender, request, user, \*\*kwargs):  
 send\_mail(  
 'New Login Detected',  
 f'Hello {user.username}, someone just logged into your account. If this wasn’t you, please change your password.',  
 'no-reply@yourapp.com',  
 [user.email],  
 fail\_silently=True,  
 )

Then import this signal in your app’s apps.py or ready() function.