

=31 December 2019

Django Authentication

User model

`django.contrib.auth.models.User`

Fields

1. Username must be unique
2. Password must be provided.
3. `is_staff`, `is_active`
4. `last_login` : When the user last logged in

Attributes

1. `is_authenticated` : To check if the given user is authenticated or not.

Methods

1. `get_username()`
2. `set_password()` : Does not save the User instance
3. `check_password()`

User model has custom manager so it provides some extra functionalities like

1. `create_user` : Creates, saves and returns a user.

Authenticating a User

1. `authenticate()` : Verify username and password and return a user instance if valid

Utility functions

1. `get_user(request)` : Returns the user model instance associated with the given request's session.
2. `login(request, user)` : Logs in a user
3. `logout(request)` : Logout user

If we use `login_required` decorator or `LoginRequiredMixin`, these will search in `settings.LOGIN_URL` to access when the user is not logged in.

The `login_required` decorator does NOT check the `is_active` flag on a user, but the default *AUTHENTICATION_BACKENDS* reject inactive users.