

# TRAINING DAY18 REPORT

14 JULY 2025

## What is a Slug?

A **slug** is a *URL-friendly, human-readable string* used to identify a resource (like a blog post) in a website's URL.

It usually:

- Converts spaces into **hyphens (-)**
- Removes **special characters**
- Uses **lowercase** letters only

## Example:

- If your blog post title is:

"My First Django Blog!"

- Then its **slug** would be:

"my-first-django-blog"

So instead of having a URL like:

```
bash  
  
/post/12/
```

you can have a cleaner and more descriptive URL like:

```
bash  
  
/post/my-first-django-blog/
```

## Why Slugs Are Used?

### 1. SEO (Search Engine Optimization)

Search engines prefer descriptive URLs — it helps ranking and readability.

e.g.,

example.com/post/django-introduction is better than example.com/post/5.

### 2. User-Friendly Links

Easier to read and share.

You can guess what the page is about just by looking at the link.

### 3. Uniqueness and Identification

Each post gets its own unique slug (like a readable ID).

### 4. Professionalism

It makes your website URLs look clean and production-grade.

## What is a SlugField (in Django)?

In Django models, **SlugField** is a special field type designed to store these *URL-friendly strings (slugs)*.

It behaves like a CharField, but it's intended specifically for slugs.

Syntax Example:

```
blogapi > blog > models.py > ...
1  from django.db import models
2
3  class Post(models.Model):
4      title = models.CharField(max_length=200)
5      slug = models.SlugField(unique=True)
6
```

## Key Attributes of SlugField:

Attribute	Description
unique=True	Ensures that no two slugs are the same.
blank=True	Allows the field to be optional in forms.
null=True	Allows the field to store NULL in the database.
max_length	Limits the number of characters. Default: 50.
db_index=True	Optimizes lookups by slug. (Enabled by default)

## Automatically Generating Slugs

Django doesn't generate slugs automatically — you must create them manually or use a helper like `slugify()`.

Example:

```
blogapi > blog > models.py > ...
1  from django.utils.text import slugify
2
3  class Post(models.Model):
4      title = models.CharField(max_length=200)
5      slug = models.SlugField(unique=True, blank=True)
6
7      def save(self, *args, **kwargs):
8          if not self.slug:
9              self.slug = slugify(self.title)
10             super(Post, self).save(*args, **kwargs)
11
12
```

## What slugify() Does?

`slugify()` converts a string into a valid slug:

```
python

>>> from django.utils.text import slugify
>>> slugify("Hello Django World!")
'hello-django-world'
```