🔹 Qu'est-ce que `lookup_field` dans une vue générique ?

Dans Django REST Framework (DRF), `lookup_field` est un attribut utilisé dans les **vues génériques** (`GenericAPIView` et ses dérivés) pour spécifier **quel champ** de l'objet doit être utilisé pour récupérer une ressource spécifique.

Par défaut, DRF utilise `pk` (**primary key** ou `id`) comme champ de recherche, mais `lookup_field` permet de le modifier pour utiliser un autre champ, comme `slug`, `username`, `email`, etc.

🔹 Utilisation de `lookup_field`

Si on veut récupérer un objet basé sur un champ **autre que la clé primaire**, on définit `lookup_field`.

🔹 Exemple avec `lookup_field="slug"`

Si notre modèle `Article` contient un champ `slug`, on peut l'utiliser au lieu de `pk`.

```
from rest_framework.generics import RetrieveAPIView
from myapp.models import Article
from myapp.serializers import ArticleSerializer

class ArticleDetailView(RetrieveAPIView):
    queryset = Article.objects.all()
    serializer_class = ArticleSerializer
    lookup_field = "slug"  # Utiliser le slug au lieu de l'ID
```

🔹 Explication

- **Requête standard (avec `pk`)** :

  ```
  GET /articles/1/  # Recherche par ID
  ```

- **Requête avec `lookup_field="slug"`** :

  ```
  GET /articles/my-first-article/  # Recherche par slug
  ```

🔹 Exemples avec d'autres champs

On peut aussi utiliser `lookup_field` avec d'autres attributs.

Exemple avec `username` (pour un utilisateur)

```
from rest_framework.generics import RetrieveAPIView
from myapp.models import User
from myapp.serializers import UserSerializer

class UserDetailView(RetrieveAPIView):
    queryset = User.objects.all()
    serializer_class = UserSerializer
    lookup_field = "username"
```

➡ Requête possible :

```
GET /users/johndoe/
```

🔹 `lookup_field` avec `lookup_url_kwarg`

Si l'URL utilise un paramètre différent du `lookup_field`, on peut utiliser `lookup_url_kwarg`.

Exemple :

```
from rest_framework.generics import RetrieveAPIView
from myapp.models import Article
from myapp.serializers import ArticleSerializer

class ArticleDetailView(RetrieveAPIView):
    queryset = Article.objects.all()
    serializer_class = ArticleSerializer
    lookup_field = "slug"
    lookup_url_kwarg = "article_slug"
```

🔹 **URLs :**

```
from django.urls import path
from myapp.views import ArticleDetailView

urlpatterns = [
    path("articles/<str:article_slug>/", ArticleDetailView.as_view(),
name="article-detail"),
]
```

➡ Requête possible :

```
GET /articles/my-first-article/
```

🔹 Résumé

Propriété	Description
`lookup_field`	Spécifie le champ à utiliser pour récupérer un objet au lieu de `pk`
`lookup_url_kwarg`	Spécifie le nom du paramètre dans l'URL (optionnel)

✔ **Par défaut**, DRF utilise `pk` pour retrouver un objet.
✔ **Avec `lookup_field`**, on peut utiliser un autre champ (`slug`, `username`, `email`, etc.).
✔ **Si le paramètre dans l'URL est différent de `lookup_field`**, on utilise `lookup_url_kwarg`.

📌 **Utile pour les URLs plus lisibles et SEO-friendly ! 🚀**