

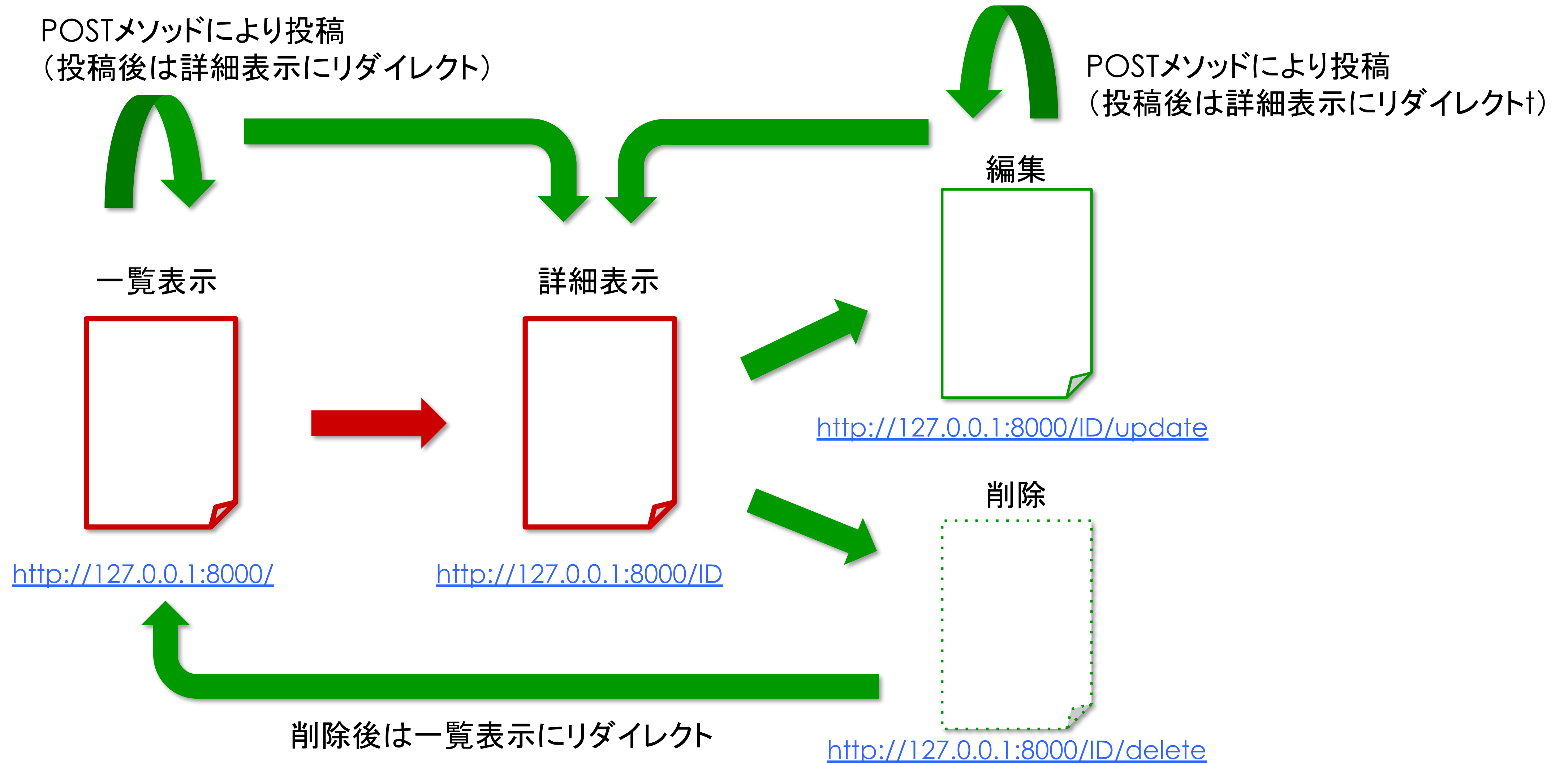
# ③

## 詳細表示の連携

---

続けて、詳細表示ページにもデータベースの内容を反映させましょう。

# ステップ②



# (再掲) レコードの取得

- 次の書き方を覚えましょう
  - クラス名.objects.get(pk=ID)
    - 指定したIDのレコードを取得します
    - オブジェクトが見つからなかった場合は、クラス名.models.DoesNotExistという例外が発生します
- レコードの各カラムの値は、オブジェクトのインスタンス変数としてアクセスできます

# views.py の編集

- **detail** 関数を、先ほどと同様に以下のように編集します

```
from django.shortcuts import render, redirect
from django.http import HttpResponseRedirect
from django.http import Http404
from django.utils import timezone
import random
from blog.models import Article
...
```

} 例外 **Http404** のインポート

```
...
def detail(request, article_id):
    try:
        article = Article.objects.get(pk=article_id)
    except Article.DoesNotExist:
        raise Http404("Article does not exist")
    context = {
        'article': article
    }
    return render(request, "blog/detail.html", context)
```

} 与えられたIDのArticleレコードを取得し、なければ例外を送出

# detail.html の追加

- テンプレートに、**blog/templates/blog/detail.html** を追加

```
<!DOCTYPE html>
<html>
  <head>
    <title>Detail</title>
    <meta charset="UTF-8"></head>
  <body>
    <h1>Django Sample Application</h1>
    <h2>Title: {{ article.title }}</h2>
    <small>Posted at:{{ article.posted_at }}</small>
    <div>
      {{ article.body | linebreaksbr }}
    </div>
    <hr>
    <p>
      <a href="{% url 'index' %}">Top</a>
    </p>
  </body>
</html>
```

Article レコードのtitle, posted\_at, bodyが、テンプレートに挿入される

トップページへのリンク

# 動作確認

- DjangoサーバをCtrl+Cで終了してから再度起動し、詳細画面を確認しましょう
  - 一覧画面から詳細画面に遷移できるようになるはずです

← → ↻ ⓘ 127.0.0.1:8000

## Django Sample Application

You can submit articles from the following form.

Title

Text

[編集](#) [2021年11月29日17:52]  
1つ目の本文

[3つ目](#) [2021年11月29日17:56]  
3つ目の本文

← → ↻ ⓘ 127.0.0.1:8000/1/

## Django Sample Application

### Title: 編集

Posted at:2021年11月29日17:52

1つ目の本文

[Top](#)

<http://127.0.0.1:8000/1/>

# ControllerからViewへの変数の受渡し

## ● views.detail

```
def detail(request, article_id):
    try:
        article = Article.objects.get(pk=article_id)
    except Article.DoesNotExist:
        raise Http404("Article does not exist")
    context = {
        'article': article
    }
    return render(request, 'blog/detail.html', context)
```

右のテンプレート内で出現する赤字のarticleは、左のController内の赤字のarticleを指しており、その値は青字のarticleとなる。

つまりArticle.objects.get(pk=article\_id)の結果得られたArticleオブジェクトが受け渡されている。

## ● blog/templates/blog/detail.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Detail</title>
    <meta charset="UTF-8"></head>
  <body>
    <h1>Django Sample Application</h1>
    <h2>Title: {{ article.title }}</h2>
    <small>Posted at:{{ article.posted_at }}</small>
    <div>
      {{ article.body | linebreaksbr }}
    </div>
    <hr>
    <p>
      <a href="{% url 'index' %}">Top</a>
    </p>
  </body>
</html>
```