

メモを表示させよう

このレッスンでは、前回のレッスンで作成したテンプレートファイルやViewに修正を加えて画面上にメモを表示させていきます。

その前に、まずはAdminページからいくつかメモを作成してみましょう。models.pyで設定したように、テキストは空欄でも保存できます。作成日と更新日は自動で保存されるフィールドですので入力フォームには表示されません。

memo を追加

Title: 買うもののリスト

Text: 卵
オレンジジュース
洗剤
タオル

保存してもう一つ追加 保存して編集を続ける 保存

作成できたら、それぞれのファイルを以下のように編集してください。ここでは、トップページでメモを表示させると同時に、メモの詳細画面、新規メモ投稿画面を作っています。detail.htmlとnew_memo.htmlは新たに作成してください。

内容としては、1つ目のチュートリアルとほぼ同じですので、もしわからないところがあればそちらを参考にしてください。今回新たに学習するポイントについては、下で説明します。

~/memo/app/urls.py

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

app_name = 'app'
urlpatterns = [
    path('', views.index, name='index'),
    path('<int:memo_id>', views.detail, name='detail'),
    path('new_memo', views.new_memo, name='new_memo'),
]
```

~/memo/app/views.py

```
from django.shortcuts import render
from .models import Memo
from django.shortcuts import get_object_or_404

def index(request):
    memos = Memo.objects.all().order_by('-updated_datetime')
    return render(request, 'app/index.html', {'memos': memos })

def detail(request, memo_id):
    memo = get_object_or_404(Memo, id=memo_id)
    return render(request, 'app/detail.html', {'memo': memo})

def new_memo(request):
    return render(request, 'app/new_memo.html')
```

~/memo/app/templates/app/index.html

```
<a class="btn" href="{% url 'app:new_memo' %}">新規メモ作成</a>

{% for memo in memos %}
<a href="{% url 'app:detail' memo.id %}" class="memo-title">
  <div>
    {{ memo }}
    <span class="updated_datetime">{{ memo.updated_datetime }}</span>
  </div>
</a>
{% endfor %}
```

~/memo/app/templates/app/detail.html

```
<div>
  <a href="{% url 'app:index' %}">ホームに戻る</a>
</div>

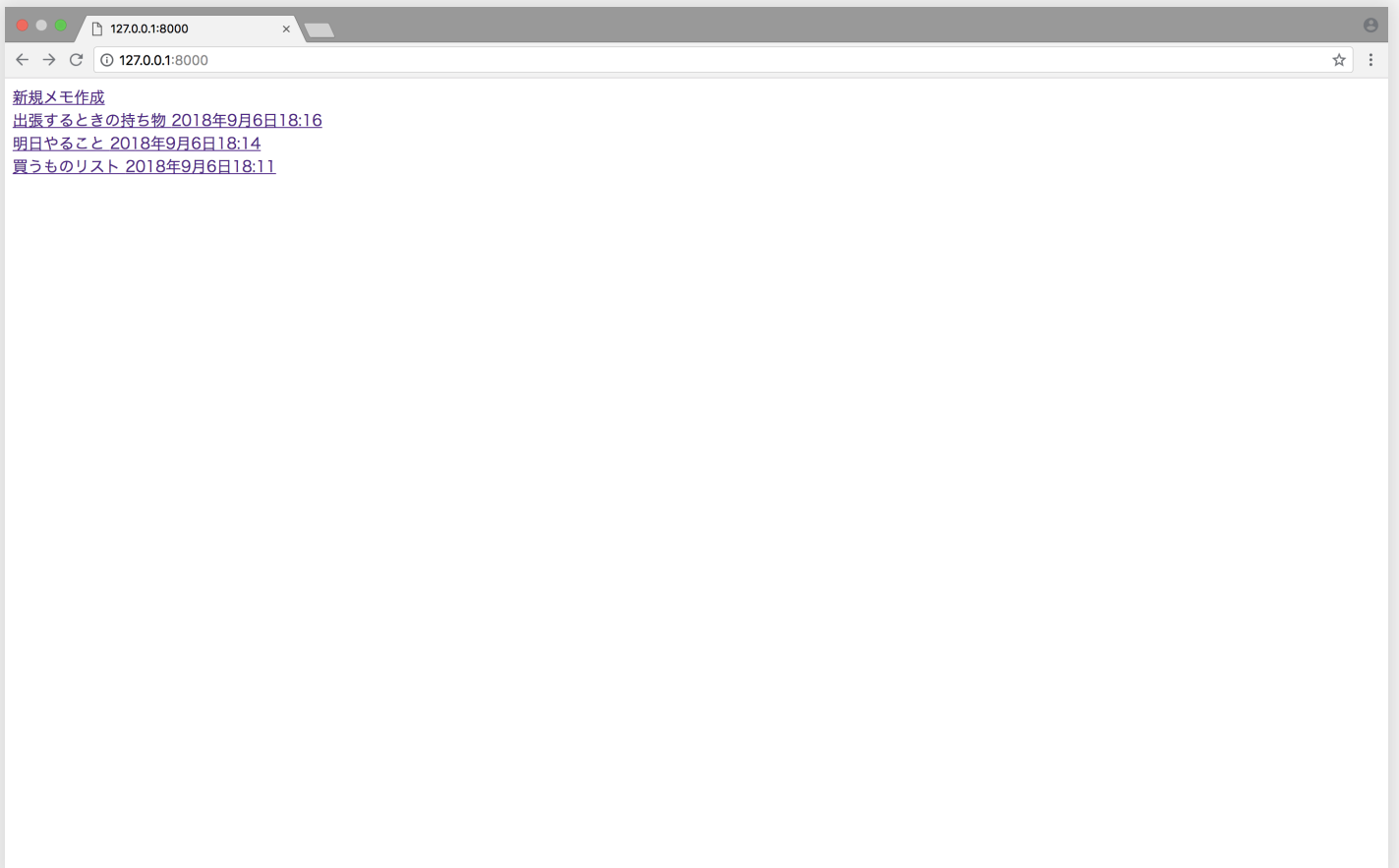
<h2>{{ memo.title }}</h2>
<div>
  {{ memo.text | linebreaks | urlize }}
</div>
```

~/memo/app/templates/app/new_memo.html

```
<div>
  <a href="{% url 'app:index' %}">ホームに戻る</a>
</div>
```

これで、それぞれのページと各ページを移動するリンクが設定できているはずです。

トップページ(index.html)



詳細ページ (detail.html)



新規メモ投稿画面(new_memo.html)



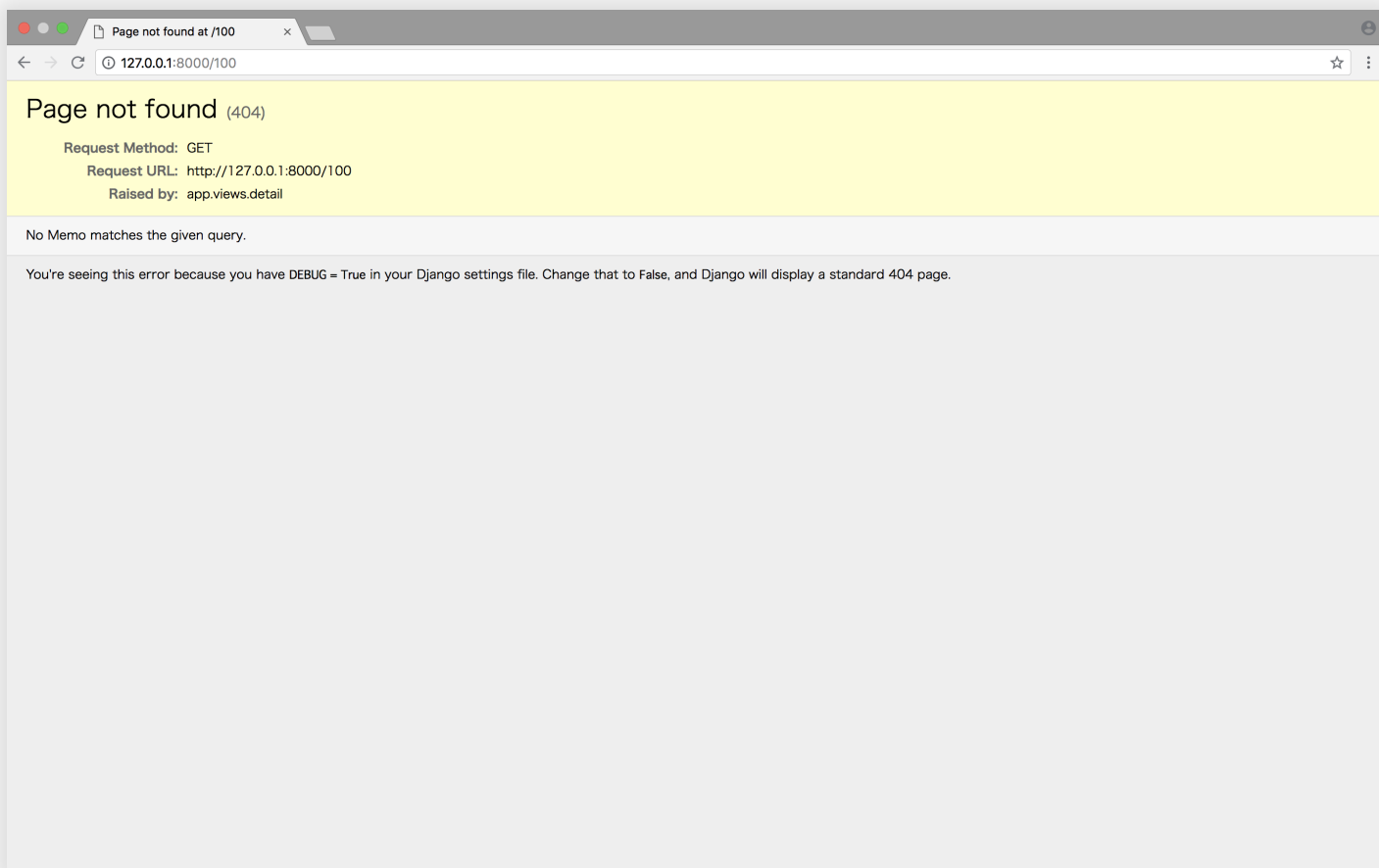
今回はじめて出てきたものとして、views.pyの `get_object_or_404` があります。

ここは、`memo = Memo.objects.get(id=memo_id)` のようにgetメソッドを使ってもよいのですが、この場合、memo_idが存在しなかった場合、エラーとなってしまいます。

つまり、id:100のメモが存在していないに関わらず、`http://127.0.0.1:8000/100` にアクセスするとエラーが発生します。

一方、`get_object_or_404` では、存在しないidが指定された場合は404ページを表示します。404ページとは、ユーザーに「ページが見つからなかったですよ」と伝えるページで

す。



何も設定しないと、デフォルトの404ページが表示されますが、[404ページは自分で編集することが可能です](#)。

`get_object_or_404` は、インポートしないと使えませんが忘れないようにしましょう。

続いて、`detail.html`の `{{ memo.text | linebreaks | urlize }}` についてです。これは、1つ目のチュートリアルでも出てきたタグフィルターを適用しています。

linebreaks:テキストの改行を反映して表示してくれるタグです。[ドキュメント](#)

urlize:URLとEmailアドレスをクリック可能な状態で表示します。[ドキュメント](#)

それぞれのタグをつけたり消したりして、それぞれがどんな役割を果たしているか実際に試してみてください。

次のレッスンでは、Webサービスを作る上で欠かせないCRUDという概念を学習していきましょう！