

# 復習 : DjangoによるWebアプリケーション開発

[Bookmark](#)

⚠ 現在回答を受け付けていません。

## Django

1.

Djangoに関する記述の中で、正しいものを全て選びなさい。

- (1) MVCのうち、情報の表示・出力に対応するのは、Cである。
- (2) オブジェクト関係マップがあると、Webアプリケーションからデータベースが必要なくなる。
- (3) URLディスパッチャは、URLとリクエストを処理する関数を関連付けるための仕組みである。
- (4) Webアプリケーションの機能の更新時に、データベースを更新する仕組みを、マイグレーションと呼ぶ。

☐ (1) ☐ (2) ☐ (3) ☐ (4)

2.

Djangoのプログラムについて、以下の問いに答えよ。

a)

このプログラムには、noteというアプリケーションがあり、以下に示すURLパターンが定義されている。

```
from django.contrib import admin
from django.urls import path
from note import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name='index'),
    path('<int:event_id>/', views.info, name='info'),
    path('<int:event_id>/update', views.update, name='update'),
    path('<int:event_id>/delete', views.delete, name='delete'),
    path('<int:event_id>/important', views.mark, name='important'),
]
```

このプログラムをPC上で動かした時、以下のURLにアクセスすると、それぞれの関数が呼び出されるか？

- `http://127.0.0.1:8000/`: `views.[ A ]` が呼ばれる
- `http://127.0.0.1:8000/10/`: `views.[ B ]` が呼ばれる
- `http://127.0.0.1:8000/10/update`: `views.[ C ]` が呼ばれる
- `http://127.0.0.1:8000/10/important`: `views.[ D ]` が呼ばれる

- (A)

- (B)

- (C)

- (D)

b)

このプログラムの `note/models.py` に、以下のようなモデルが定義されている。

```
from django.db import models
from django.utils import timezone

class Event(models.Model):
    title = models.CharField(max_length=50)
    description = models.TextField()
    memo = models.TextField()
    created_at = models.DateTimeField(default=timezone.now)
    importance = models.IntegerField(default=0)
```

データベースにレコードが1件も登録されていない状態で、Djangoのシェル上でコマンドを実行したところ、以下のようになった。

```
>>> from note.models import Event
>>> e = Event(title='Post 1', description='This is the first post.', memo='This is the first memo.')
>>> e.save()
>>> e = Event(title='TEST', description='This is the test post.', memo='This is the test memo.')
>>> e.save()
>>> e = Event(title='Post 2', description='This is the second post.', memo='This is the second memo.')
>>> e.save()
>>> e = Event(title='TEST', description='This is another test post.', memo='This is another test memo.')
>>> e.save()
>>> for e in Event.objects.all():
...     print(e.id, e.description)
...
8 This is the first post.
9 This is the test post.
10 This is the second post.
11 This is another test post.
>>> e = Event.objects.get(pk=[ E ])
>>> e.memo
'This is the second memo.'
>>> e = Event.objects.get(pk=11)
>>> e.description = 'This post will be removed.'
>>>
>>> for e in Event.objects.[ F ](title='TEST'):
...     print(e.title, e.description)
...
TEST This is the test post.
TEST [ G ]
>>> e = Event.objects.order_by('-created_at')[1]
>>> e.title
'[ H ]'
```

空欄を埋めなさい。

- (E)

- (F)

- (G)

- (H)

c)

このプログラムのトップ画面 (<http://127.0.0.1:8000/>) は、以下のような機能を持つ。

- タイトル(Title) と 説明(Description) と メモ(Memo) を投稿することができる。
- イベントには、重要度(Importance)を設定することができる。重要度は、0(Normal)、1(Important)、2(Very important)の3段階である。

- 投稿されたイベントの一覧が表示されている。このページ内のリンクから、イベントの表示順序を、以下の順序で並べ替えることができる。
  - Date : 投稿日が新しい順
  - Importance : 重要度が高い順
- イベント毎に以下が表示される。
  - タイトル (詳細画面へのリンクとなっている)
  - 投稿日
  - 説明
  - メモ
  - 重要度
  - 重要度を設定するリンク

この時、note/views.pyの index 関数、 mark 関数、テンプレートファイル index.html は以下の通りである。空欄を埋めよ。

```
def index(request):
    if request.method == 'POST':
        event = Event(title=request.POST['title'], description=request.POST['description'], memo=request.POST['memo'])
        event.save()

    if '[ I ]' in request.GET:
        if request.GET['[ I ]'] == 'importance':
            events = Event.objects.[ ]('-importance')
        else:
            events = Event.objects.[ ]('-created_at')
    else:
        events = Event.objects.[ ]('-created_at')

    context = {
        'events': events
    }
    return render(request, 'note/index.html', context)
```

```
def mark(request, event_id):
    try:
        event = Event.objects.get(pk=event_id)
    except Event.DoesNotExist:
        raise Http404("Event does not exist")

    if 'importance' in request.GET:
        event.importance = request.GET['importance']
        event.save()

    return redirect(index)
```

```
<!DOCTYPE html>
<html lang="en">

  <head>
    <title>Midterm Exam Application</title>
    <meta charset="UTF-8">
  </head>

  <body>

    <h1>Midterm Exam Application</h1>
    <form action="{% url 'index' %}" method="post">
      {% csrf_token %}
      <div>
        <label for="titleInput">Title</label>
        <input id="titleInput" name="title" type="text" placeholder="Input title">
      </div>
      <div>
        <label for="descriptionInput">Description</label>
        <textarea id="descriptionInput" name="description" rows="3"></textarea>
      </div>
      <div>
        <label for="memoInput">Memo</label>
        <textarea id="memoInput" name="memo" rows="2"></textarea>
      </div>
      <div>
        <button type="submit">Submit</button>
      </div>
    </form>

    <h2>Events</h2>
    <div>Sort by:
      <a href="{% url 'index' %}?order=date">Date</a> |
      <a href="{% url 'index' %}?order=importance">Importance</a>
    </div>
    {% for [ K ] %}
    <div>
      <h3><a href="{% url 'info' e.id %}">{{ [ L ] }}</a></h3>
      <p>
        <small>[{{ e.created_at }}] Importance : {{ e.importance }}</small>
      </p>
      <h4>Description</h4>
      <p>{{ e.description | linebreaksbr }}</p>
      <h4>Memo</h4>
      <p>{{ e.memo | linebreaksbr }}</p>
      <div>
        <a href="{% url 'important' e.id %}?[ M ]=2">Very important</a>
        <a href="{% url 'important' e.id %}?[ M ]=1">Important</a>
        <a href="{% url 'important' e.id %}?[ M ]=0">Normal</a>
      </div>
    </div>
    {% endfor %}
  </body>

</html>
```

- (I)

- (J)

- (K)

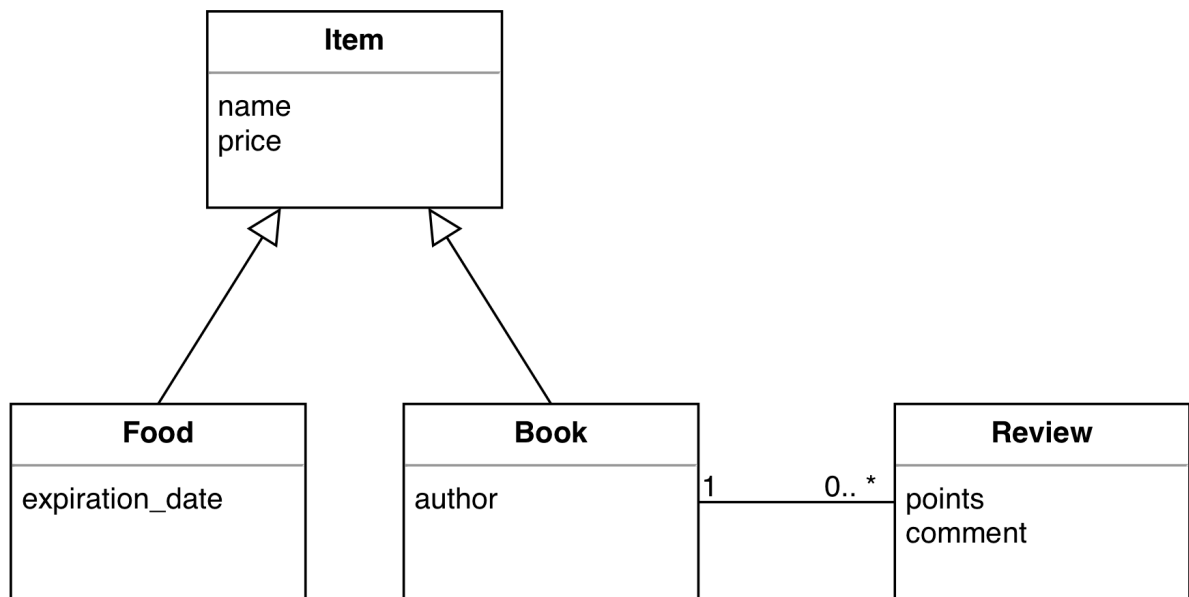
- (L)

- (M)

### 3.

以下の文章が、下に示す図の説明になるように、空欄に適切な語句を埋めよ。

- この図は、UMLにおける[ A ]というダイアグラムである。
- このダイアグラムには、[ B ]個のクラスが含まれている。
- [ C ]クラスおよび[ D ]クラスは、Itemクラスの一環である。このような関係のことを[ E ]と呼ぶ。
- BookクラスとReviewクラスの間に関連は、1つの本に複数のレビューがあることを示している。この時、関連の両端に記載された数値のことを[ F ]と呼ぶ。



- (A)

- (B)

- (C)

- (D)

- (E)

- (F)

---

\* 回答は自動的に記録されますが、最後に「提出」ボタンをクリックし提出を確認してください。

 提出

« Previous (/courses/2024/COT102/14-2/14-2-2)

Next » (/courses/2024/COT102/14-2/14-2-5)

---

Copyright © 2017-2023 INIAD (<https://www.iniad.org>). All rights reserved.