Ask Django

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

EP 1. Overview

선행 코스

- 장고 - 기본편
  - Form/View/Template 에 대한 이해
- 웹프론트엔드 시작편
  - HTML/CSS/JavaScript에 대한 이해

장고 위젯 개발은 **웹프론트엔드**의 영역입니다.
하지만, 장고 위젯 활용은 **파이썬/장고**의 영역입니다.

잘 만든 위젯 하나,
열 *React* 안 부럽다. ;)

누리세요.

모든 웹페이지를 SPA Single Page App 으로 작성할 필요는 없습니다.
적절히 조합해서 쓰시고 누리실 건 누리세요.

장고가 지원해주는 강력한 **Form/Widget** ~ !!! :D

Django Widgets 이란? #doc

1. 지정된 Form Field에 대한 **HTML 태그** 생성을 담당
2. 각 Model Field 마다 디폴트 Form Field가 지정되어있고,
    - 각 Form Field마다 디폴트 Widget이 지정되어있습니다.

모델 필드	디폴트 폼 필드	디폴트 폼 위젯
`models.CharField`	`forms.CharField`	`forms.TextInput`
`models.TextField`	`forms.CharField`	`forms.Textarea`
`models.URLField`	`forms.CharField`	`forms.TextInput`

Example

한 Post에 대한 위치(위도/경도)를 저장하려 합니다.

그래서 모델에 다음과 같이 필드를 정의했으며,

```
class Post(models.Model):
    location = models.CharField(max_length=100)
```

다음과 같이 모델폼을 정의했습니다.
혹은 직접 Form 필드를 정의하실 수도 있습니다.

```
class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        fields = ['location']
```

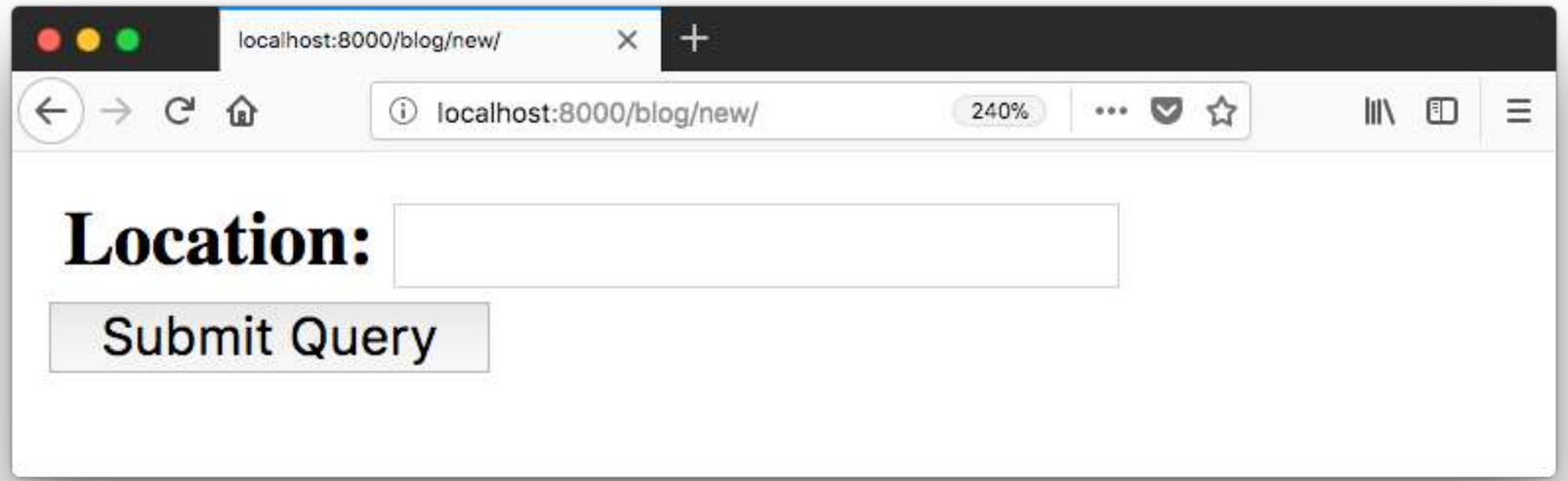
템플릿에서는 다음과 같이 폼태그를 생성하겠죠.

```
<form action="" method="post">
    {% csrf_token %}
    <table>
        {{ form.as_table }}
    </table>
    <input type="submit" />
</form>
```

이는 다음과 같은 HTML을 생성(render)합니다.

```
<form action="" method="post">
    <input type='hidden' name='csrfmiddlewaretoken'
           value='wVC7tru0QzeLBzmPXn3JzQBagZHYlhN7qf99SVCsvIZ5wTOGSRyPea78QMUwq98b' />
    <table>
        <tr>
            <th>
                <label for="id_location">Location:</label>
            </th>
            <td>
                <input type="text" name="location" maxlength="100" required id="id_location" />
            </td>
        </tr>
    </table>
    <input type="submit" />
</form>
```

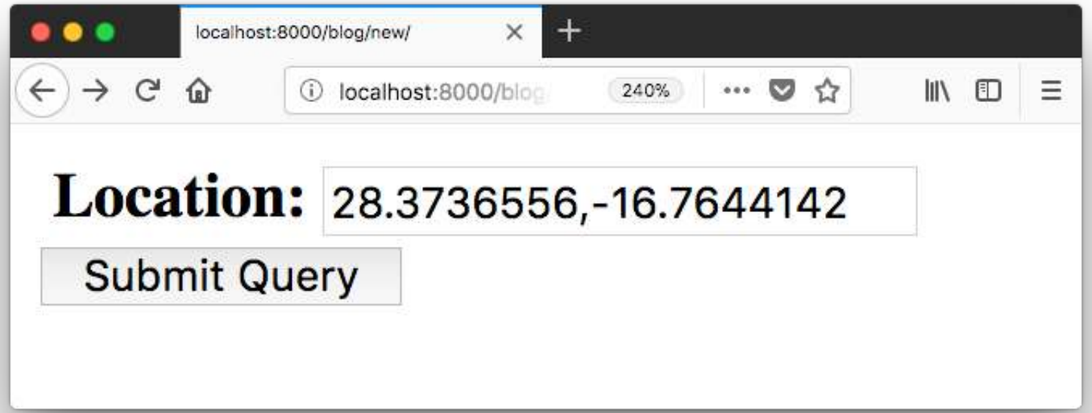
다음과 같은 UI가 보여집니다.



가라치코 여행을 다녀와서 포스팅을 하려한다면,

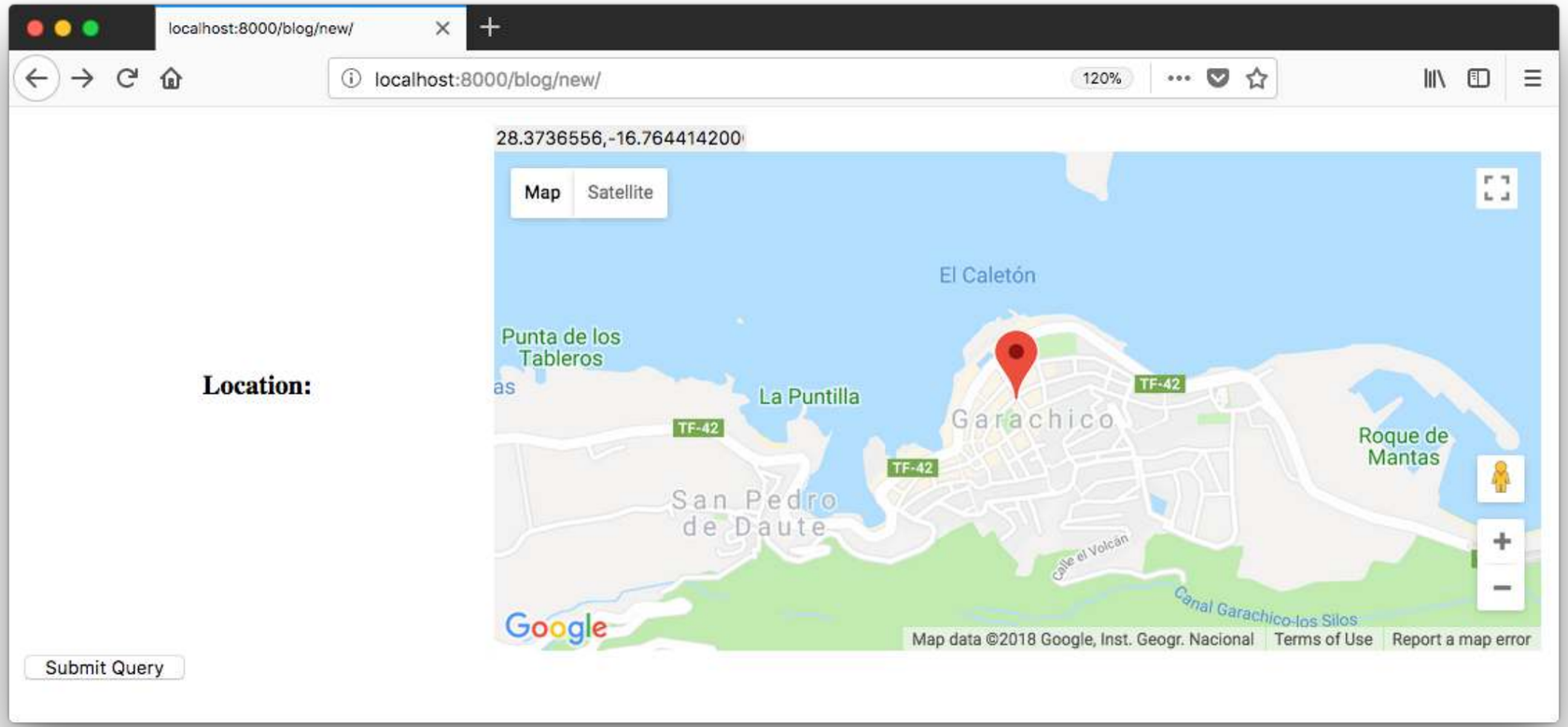
윤식당2 촬영지 위도/경도는 **28.3736556,-16.7644142** 입니다.

유저에게 다음과 같이 입력하라고 요구하는 것은 ... 너무 가혹한 일입니다.

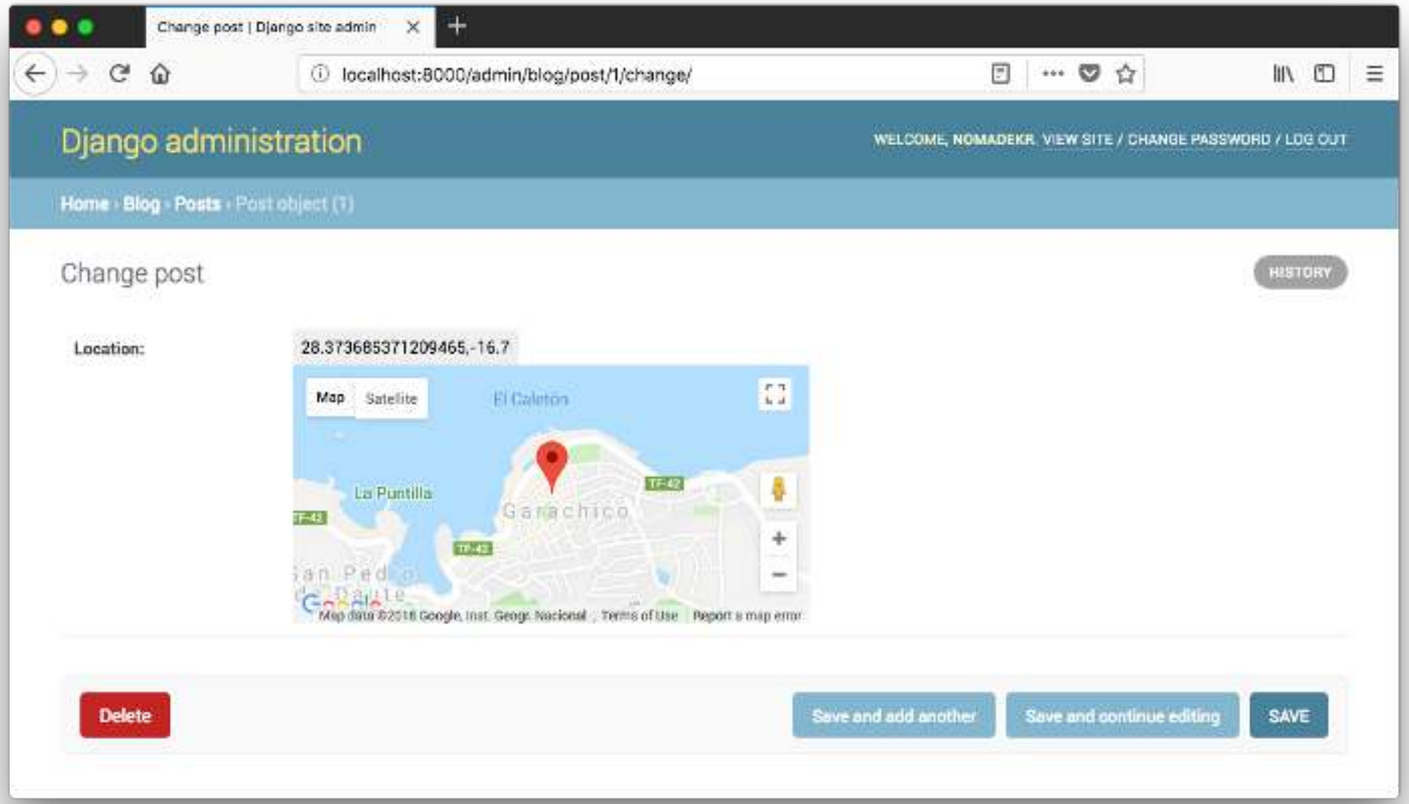


정확한 위도/경도를 알아내기도 힘들 뿐더러, 오타칠 확률도 높죠. :(

다음과 같은 UI를 제공한다면 어떤가요?
지도를 클릭하면, 클릭위치를 프로그램JavaScript을 통해 기입하기



admin 페이지에서도 이렇게 사용하실 수 있어요. ~ :D
admin은 Django Form을 사용하니깐요.



위젯을 통해
재사용성 높은 편리한 UI를 만
드실 수 있어요.

장고가 기본 제공해주는 위젯 #doc

- Text : TextInput, NumberInput, EmailInput, URLInput, PasswordInput, HiddenInput, DateInput, DateTimeInput, TimeInput, Textarea
  - 모두 생긴 건 기본 `<input type="text" />`
- Selector and checkbox : CheckboxInput, Select, NullBooleanSelect, SelectMultiple, RadioSelect, CheckboxSelectMultiple
  - 모두 생긴 건 기본 `<select></select>`
- File Upload : FileInput, ClearableFileInput
  - 모두 생긴 건 기본 `<input type="file" />`
- Composite : MultipleHiddenInput, SplitDateTimeWidget, SplitHiddenDateTimeWidget, SelectDateWidget

장고 위젯형태로 제공되는 위젯이 그리 다양하진 않아요.

- 웹프론트엔드 세상은 춘추전국시대.
  - 너무나도 빠르게 변화해갑니다.
- 다양한 UI는 HTML/CSS/JS 형태로는 쉬이 가져올 수도 있지만, 장고 위젯형태로는 공개된 것이 그리 많진 않아요.

그러니, 필요한 UI의 HTML/CSS/JS를 찾아서 위젯으로 만들어 쓰시면 OK

체험해보기

django-summernote

모델 코드가 다음과 같다면 ...

```
class Post(models.Model):
    title = models.CharField(max_length=200)
    content = models.TextField()
```

django-summernote를 적용하는 방법이 다양하지만, 다음과 같이 위젯을 지정할 수 있습니다.

```
from django_summernote.widgets import SummernoteWidget

class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        widgets = {
            'content': SummernoteWidget,
        }
```

Tip: 필히 다양한 *OPTIONS*을 확인하세요.

위젯을 지정하는 다양한 방법

1) Form Field는 그대로 두고, Widget만 변경하기

지원 : ModelForm

```
class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        widgets = {
            'content': SummernoteWidget,
        }
```

2) Form Field (재)정의하면서, Widget 지정하기

지원 : Form / ModelForm

```
class PostForm(forms.ModelForm):
    content = forms.CharField(widget=SummernoteWidget)  # 기존 Form Field는 무시

    class Meta:
        model = Post
```

3) 생성자에서 변경하기 하드코어

지원 : Form / ModelForm

```
class PostForm(forms.ModelForm):
    class Meta:
        model = Post

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['content'].widget = SummernoteWidget()
```

Widget 지정 시에는

*Class*로 지정하거나, *Instance*를 생성해서 지정해도 됩니다.

- Class로 지정하면, Instance를 생성해줍니다.
- 아래 코드 모두 OK

```
widgets = {'content': SummernoteWidget}      # Case 1
widgets = {'content': SummernoteWidget()}    # Case 2
```

attrs 속성

Instance를 생성할 때, 생성자에 공통적으로 attrs를 지정할 수 있습니다.
이는 태그의 커스텀 속성으로 활용됩니다.

다음 위젯 코드는

```
TextInput(attrs={'data-type': 'number'})
```

다음과 같이 표현됩니다.

```
<input data-type="number" />
```

위젯 종류에 따라 생성자에 추가 인자를 지원할 수도 있습니다.

이번 코스에서 다룰 위젯

- 별점 위젯
- 자동완성 위젯
- 캘린더 Datefield
- 이미지 보이는 파일 업로더
- 구글맵 HiddenInput
- 위지위그 에디터 (with Toast UI Editor)

before

```
from django import forms
from .models import Post

class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        fields = '__all__'
```

after

위젯만 잘 만들어두면, 두고 두고 편리하게 :D

```
from django import forms
from django.urls import reverse_lazy
from .models import Post
from .widgets import (
    AutoCompleteSelect, DatePickerWidget, LocationWidget, PreviewClearableFileInput,
    RateitjsWidget, TuiEditorWidget)

class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        fields = '__all__'
        widgets = {
            'country': AutoCompleteSelect(ajax_url=reverse_lazy('country_list')),
            'rating': RateitjsWidget,
            'location': LocationWidget,
            'when': DatePickerWidget,
            'message': TuiEditorWidget,
            'photo': PreviewClearableFileInput,
        }
```

차근차근
다양한 위젯을 만들어봅시다.