Ask Django

여러분의 파이썬/장고 페이스메이커가 되겠습니다.
EP 4. 위젯 만들기 (자동완성 Select 위젯)

기본 구현

blog/forms.py

```
from django import forms
from django.urls import reverse_lazy
from .models import Post
from .widgets import AutoCompleteSelect, RateitjsWidget

class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        fields = '__all__'
        widgets = {
            'country': AutoCompleteSelect,
            'rating': RateitjsWidget,
        }
```

blog/widgets.py

```
class AutoCompleteSelect(forms.Select):
    template_name = 'widgets/autocomplete_select.html'

    class Media:
        css = {
            'all': [
                '//cdnjs.cloudflare.com/ajax/libs/select2/4.0.6-rc.0/css/select2.min.css',
            ],
        }
        js = [
            '//code.jquery.com/jquery-2.2.4.min.js',
            '//cdnjs.cloudflare.com/ajax/libs/select2/4.0.6-rc.0/js/select2.min.js',
        ]

    def build_attrs(self, *args, **kwargs):
        context = super().build_attrs(*args, **kwargs);
        context['style'] = 'min-width: 200px;'
        return context
```

blog/templates/widgets/autocomplete_select.html

```
{% include "django/forms/widgets/select.html" %}

<script>
$(function() {
    var options = {minimumInputLength: 1};
    $('#{{ widget.attrs.id }}').select2(options);
});
</script>
```

개선 포인트

- ForeignKey를 통한 Select 위젯이기에, 매번 그려질 때마다 전체 Country Record에 대해 DB쿼리
  - django-debug-toolbar를 통해 확인해봅시다.

Ajax를 통한 자동완성

blog/forms.py

```python
class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        fields = '__all__'
        widgets = {
            'country': AutoCompleteSelect(ajax_url=reverse_lazy('country_list')),
            'rating': RateitjsWidget,
        }
```

blog/widgets.py

```
class AutoCompleteSelect(forms.Select):
    template_name = 'widgets/autocomplete_select.html'

    class Media:
        css = {
            'all': [
                '//cdnjs.cloudflare.com/ajax/libs/select2/4.0.6-rc.0/css/select2.min.css',
            ],
        }
        js = [
            '//code.jquery.com/jquery-2.2.4.min.js',
            '//cdnjs.cloudflare.com/ajax/libs/select2/4.0.6-rc.0/js/select2.min.js',
        ]

    def __init__(self, ajax_url, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.ajax_url = ajax_url

    def build_attrs(self, *args, **kwargs):
        context = super().build_attrs(*args, **kwargs);
        context['style'] = 'min-width: 200px;'
        return context

    def get_context(self, *args, **kwargs):
        context = super().get_context(*args, **kwargs);
        context['ajax_url'] = self.ajax_url
        return context

    def optgroups(self, name, value, attrs=None):
        existed_ids = [_id for _id in value if _id]
        self.choices.queryset = self.choices.queryset.filter(id__in=existed_ids)
        return super().optgroups(name, value, attrs=None)
```

blog/templates/widgets/autocomplete_select.html

```
{% include "django/forms/widgets/select.html" %}

<script>
$(function() {
    var options = {minimumInputLength: 1};

    var ajax_url = "{{ ajax_url|default:"" }}";
    if ( ajax_url.length > 0 ) {
        options['ajax'] = {
            'url': ajax_url,
            'dataType': 'json'
        };
    }

    $('#{{ widget.attrs.id }}').select2(options);
});
</script>
```

blog/widgets.py

blog/views.py

```
from django.http import JsonResponse
from .models import Country

def country_list(request):
    qs = Country.objects.all()
    q = request.GET.get('q')
    qs = qs.filter(name__icontains=q)
    results = [{'id': country.id, 'text': country.name} for country in qs]
    return JsonResponse({
        'results': results,
    })
```

blog/urls.py

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

urlpatterns = [
    path('country/', views.country_list, name='country_list'),
]
```

Life is short,
use Python3/Django.