

Ask Django

여러분의 파이썬/장고 페이스메이커가 되겠습니다.
EP 6. PayForm 개선 및 SMS 인증

API 페이지를 통한 API 확인

- api.iamport.kr 페이지에서 API Key/Secret 를 통한 인증 수행

PayForm 개선

shop/mixins.py

재사용성을 위해 **IamportBaseForm** 클래스 생성

```
import json
from django import forms
from django.conf import settings
from django.template.loader import render_to_string
from django.utils.encoding import smart_text
from django.utils.safestring import mark_safe
from django.core.serializers.json import DjangoJSONEncoder

class IamportBaseForm(forms.ModelForm):
    template_name = None
    params_names = None
    imp_fn_name = None

    def get_iamport_params(self):
        params = {}
        for name in self.params_names:
            if hasattr(self.instance, name):
                params[name] = getattr(self.instance, name)
        return params
```

```
# IamportBaseForm 코드를 이어 ~~~
```

```
def as_iamport(self):
    hidden_fields = mark_safe(''.join(smart_text(field) for field in self.hidden_fields()))

    iamport_params = self.get_iamport_params()
    json_string = json.dumps(iamport_params, cls=DjangoJSONEncoder, ensure_ascii=False)
    iamport_json_fields = mark_safe(json_string)

    template_name = getattr(self, 'template_name', None)
    if not template_name:
        raise ValueError('아임포트 템플릿 경로를 지정해주세요.')

    return hidden_fields + render_to_string(template_name, {
        'json_fields': iamport_json_fields,
        'iamport_shop_id': settings.IAMPORT_SHOP_ID,
        'imp_fn_name': self.imp_fn_name,
    })

def save(self, commit=True):
    order = super().save(commit=False)
    order.update(commit=commit)
    return order
```

shop/forms.py

IamportBaseForm을 상속받아, PayForm 구현

```
from django import forms
from .models import Order
from .mixins import IamportBaseForm

class PayForm(IamportBaseForm):
    template_name = 'shop/_iamport.html'
    params_names = ['merchant_uid', 'name', 'amount']
    imp_fn_name = 'request_pay'

    class Meta:
        model = Order
        fields = ['imp_uid']
        widgets = {
            'imp_uid': forms.HiddenInput,
        }
```

shop/templates/shop/_iamport.html

불필요한 **success** 체크 루틴 제거

```
<script id="iamport-script">
$(function() {
    var params = {{ json_fields }};
    IMP.init("{{ iamport_shop_id }}");
    IMP.{{ imp_fn_name }}(params, function(response_data){
        /* response_data.success 여부에 상관없이, submit을 합니다. */
        var $form = $("#iamport-script").closest("form");
        $.each(response_data, function(key, value) {
            $form.find("input[name=" + key + "']").val(value);
        });
        $form.submit();
    });
});
</script>
```

shop/templates/shop/pay_form.html

{{ form.as_iimport }}만으로 모든 필드 렌더링

```
<div class="card-body">
    {# form #}
    {{ form.as_iimport }}    {# iimport 렌더링 #}
</div>
```


SMS 본인인증

SMS 본인인증 #doc

처리되는 로직은 결제와 동일합니다.

- 고객이 사용 중인 통신사를 통해 SMS로 전송된 인증번호(6자리)를 입력하면 해당 고객의 이름 / 생년월일 / 성별을 확인해주는 기능
- 아임포트 javascript 1.1.4버전부터 SMS본인인증기능이 제공
- 지원 PG
 - 다날(문자인증) 만 지원 [서비스 신청 페이지](#)
 - 신청하면, 영업일 기준 **5일 이내** 회신
- 설정
 - 위치: 아임포트 관리자 > 시스템설정 > 본인인증서비스
 - 가맹점 아이디와 암호 입력

아임포트 문의/답변

- 통신사에서 개인정보 확인이 성공할 때마다 과금하고 있는 부분이라 각 PG사에서 테스트용도의 계정을 제공하지 못하고 있는 현실입니다. 번거로우시겠지만 실제 계약 후 이용하시는 수 밖에 없을 것 같습니다.
- 다날 외에 모빌리언스, 나이스개인신용정보 등에서 운영하고 있습니다만 서비스상 차이가 없어 아임포트에서는 현재 다날만 지원하고 있습니다.

shop/models.py

```
from iamport import Iamport

class MyIamport(Iamport):
    def sms_cert_verify(self, imp_uid):
        url = '{}certifications/{}'.format(self.imp_url, imp_uid)
        return self._get(url)

class Certificate(models.Model):
    user = models.OneToOneField(settings.AUTH_USER_MODEL)
    merchant_uid = models.UUIDField(default=uuid4, editable=False)
    imp_uid = models.CharField(max_length=100, blank=True)
    meta = JSONField(blank=True, default={})
    success = models.BooleanField(default=False)

    @property
    def api(self):
        return MyIamport(settings.IAMPORT_API_KEY, settings.IAMPORT_API_SECRET)

    def update(self, commit=True, meta=None):
        if self.imp_uid:
            try:
                self.meta = meta or self.api.sms_cert_verify(imp_uid=self.imp_uid)
                assert str(self.merchant_uid) == self.meta['merchant_uid']
                self.success = True
            except Iamport.HttpError:
                self.success = False
        if commit:
            self.save()
```

shop/forms.py


```
from django import forms
from .models import Certificate
from .mixins import ImportBaseForm

class CertificateForm(ImportBaseForm):
    template_name = 'shop/_import.html'
    params_names = ['merchant_uid']
    imp_fn_name = 'certification'

    class Meta:
        model = Certificate
        fields = ['imp_uid']
        widgets = {
            'imp_uid': forms.HiddenInput,
        }
```

**결제코드를 참고하여,
SMS 인증을 구현해보세요.
단, 다날에 가입하셔야만
테스트 가능하십니다.**

감사합니다. ;)



*Life is short,
use Python3/Django.*