



Community urls.py

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

app_name = 'community'

urlpatterns = [
    path('', views.index, name='index'),
    path('create/', views.create, name='create'),
    path('<int:review_pk>/', views.detail, name='detail'),
    path('<int:review_pk>/comment', views.comment, name='comment'),
]
```

article 앱, community 앱에서 → article:update 이런식으로
동일한 view 함수를 가질 수 있다 community.update 함수도

from django.urls import path
from . import views : 여기서 views를 import 하겠다 view의
community / : index 함수로 이동

community/create/
: view의 create로 이동

게시글 마지막 번호
int를 입력받겠다

'특정 게시글 / 댓글'이라는 뜻

Community models.py

```
from django.db import models
from django.conf import settings
```

```
# Create your models here.
```

```
class Review(models.Model):
```

```
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    title = models.CharField(max_length=100)
    content = models.TextField()
    movie_title = models.CharField(max_length=100)
    rank = models.IntegerField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
```

```
class Comment(models.Model):
```

```
    review = models.ForeignKey(Review, on_delete=models.CASCADE)
    contents = models.CharField(max_length=200)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
```

```
def __str__(self):
```

```
    return self.contents
```

installed - app의 설정
= settings.py (가장)

user : review = 1:N
review : comment = 1:N

□ = models.ForeignKey
→ □ 를 외부로 달았다 (연결하겠다)

settings.py의
저장된 user model과
연결한다

모델이 사라질때
같이 삭제된다는
의미

Comments의
fields 중 contents로
return 하겠다
⇒ contents가 보임

```
from django import forms
from models import Review, Comment

class ReviewForm(forms.ModelForm):
    class Meta:
        model = Review
        # fields = '__all__'
        exclude = ('user',)

class CommentForm(forms.ModelForm):
    class Meta:
        model = Comment
        # fields = '__all__'
        exclude = ('user', 'review', )
```

같은 위치 내의
모델 파일에서
review와 comment를
import

리뷰모델과 연결된
form

모델과 연결된 form

리뷰모델의
fields 중
'user'은 제외한다

```

from django.shortcuts import render, redirect, get_object_or_404
from .models import Review, Comment
from .forms import ReviewForm, CommentForm
from django.views.decorators.http import require_http_methods, require_POST, require_safe
from django.contrib.auth.decorators import login_required

```

```

def index(request):
    reviews = Review.objects.all()
    context = {'reviews': reviews, }
    return render(request, 'community/index.html', context)

```

index는 모든 review를 보여준다
 reviews = Review.objects.all() → HTML에 모든 review를 담아서 보내겠다

```

@login_required
@require_http_methods(['GET', 'POST'])
def create(request):

```

로그인한 사용자만 create 가능
 GET, POST 요청만 받겠다

```

    if request.method == 'POST':
        form = ReviewForm(request.POST)
        if form.is_valid():
            review.user = request.user
            review = form.save()
            return redirect('community:detail', review.pk)

```

① 먼저, review를 작성할 빈 form 제공하기
 ②-1: 이제 form은 작성 내용이 담겨있음
 ②-2: 작성된 form이 유효하면 detail로 redirect
 ①-1: 담아서 보내는 내용 = 빈 review form

```

        form = ReviewForm()
        context = {'form': form, }
        return render(request, 'community/create.html', context)

```

③ valid X면
 create.html에 전달 (GET 요청)
 index.html에서 전달 받음

```

def detail(request, review_pk):
    review = get_object_or_404(Review, pk=review_pk)
    comment_form = CommentForm()
    comments = Comment.objects.all()
    context = {
        'review': review, 해당 review
        'comment_form': comment_form, 빈 comment form
        'comments': comments, 여타까지의 comments
    }
    return render(request, 'community/detail.html', context)

```

pk도 담아서 detail로 연결 (detail url의 <int: review-pk>가 된다!)
 Review에 pk에 해당하는 object는 가져옴
 pk가 담아서 detail로 연결 (detail url의 <int: review-pk>가 된다!)
 get_object_or_404: obj: obj= '모형' objects.get(pk=')
 raise Http404

```

def comment(request, review_pk):
    if request.user.is_authenticated:
        review = get_object_or_404(Review, pk=review_pk)
        comment_form = CommentForm(request.POST)
        if comment_form.is_valid():
            comment = comment_form.save(commit=False)
            comment.review = review
            comment.user = request.user
            comment.save()
            return redirect('community:detail', review.pk)

```

① 로그인한 사용자만
 review_pk로 id값 저장!
 POST로 받아오는 valid BUT! review id는 비어있다
 review_pk로 하면 안되니까?
 Q. review_pk로 하면 안되니까?
 Q. review_pk로 하면 안되니까?

```

    else:
        return redirect('accounts:login')

```

로그인부터
 'next' url이 경로에 설정된다

```


```

인덱스

```
{% extends 'base.html' %}

{% block content %}
<div class='container mt-3 mb-3'>
  <div class='text-center'>
    <h3>INDEX</h3>
    {% comment %} <a href="{% url 'community:create' %}"> CREATE </a> {% endcomment %}
    {% for review in reviews %}
      <div>
        <a href="{% url 'community:detail' review.pk %}">{{ review.title }}</a>
      </div>
    {% endfor %}
  </div>
</div>

{% endblock content %}
```

views.py의 index 함수에서
reviews = Review.objects.all()을
context에 담아서 가져옴

주석

review.title은 누르면
detail로 접근!
(detail로 이동이므로 pk 전달 필요)

디테일

```
{% extends 'base.html' %}
{% block content %}
<h3>제목: {{ review.title }}</h3>
<p>내용: {{ review.content }}</p>
<h4>영화제목: {{ review.movie_title }}</h4>
<p>평점: {{ review.rank }}</p>
<p>생성일자: {{ review.created_at }}</p>
<p>수정일자: {{ review.updated_at }}</p>

{% for comment in comments %}
  <p>
    {{ comment }}
  </p>
{% endfor %}

<form action="{% url 'community:comment' review.pk %}" method = 'POST'>
  {{ csrf_token }}
  {{ comment_form.as_p }}
  <button>SUBMIT</button>
</form>

{% endblock content %}
```

review 정보

views.py의 detail에서
comments = review.comment_set.all()
context에 담아서 가져옴

review.pk 아님~

view.py에서
comment_form이라고 정의했으니까!

Q. model의
-- str -- 텍스트인가?

Accounts urls.py

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

app_name = 'accounts'

urlpatterns = [
    path('', views.index, name='index'),
    path('login/', views.login, name='login'),
    path('logout/', views.logout, name='logout'),
    path('signup/', views.signup, name='signup'),
]
```

Accounts models.py

```
from django.db import models
from django.contrib.auth.models import AbstractUser

class User(AbstractUser):
    pass
```

AbstractUser 상속받아
그대로 사용하겠다
settings.py 에
AUTH_USER_MODEL = 'accounts.user'

Accounts views.py

```
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth import get_user_model
from django.contrib.auth import login as auth_login
from django.contrib.auth import logout as auth_logout
from django.contrib.auth.forms import AuthenticationForm
from .forms import CustomUserCreationForm
```

```
Custom
User = get_user_model()
```

```
def index(request):
    return render(request, 'community/index.html')
```

```
def login(request):
    if request.method == 'POST':
        form = AuthenticationForm(request, request.POST)
        if form.is_valid():
            auth_login(request, form.get_user())
            next_url = request.GET.get('next')
            return redirect(next_url or 'community:index')
        else:
            form = AuthenticationForm()
            context = {'form': form, }
            return render(request, 'accounts/login.html', context)
```

```
def logout(request):
    auth_logout(request)
    return redirect('community:index')
```

```
def signup(request):
    if request.user.is_authenticated:
        return redirect('community:index')
    if request.method == 'POST':
        form = CustomUserCreationForm(request.POST)
        if form.is_valid():
            user = form.save()
            auth_login(request, user)
            return redirect('community:index')
        else:
            form = CustomUserCreationForm()
            context = {'form': form, }
            return render(request, 'accounts/signup.html', context)
```

get_user_model
return currently active user model

Q. 어떤 active user model이 AUTH_USER_MODEL 인가?

① 먼저 빈
login form
보여주기

① 먼저 빈
signup form
보여주기

인자 2개 필요!
꼭 확인하기!!

index 요청이 오면
index.html로 연결

답아서 보내지 X
= redirect

② 오려 form은
html에서 요청받은
form이야

login.html에
context를
담아서 전달
= render

다음으로
next 연결
다음으로
index로!

form에 작성된 form을
담았음

3기아웃하고
index로 가

html 제출 버튼으로
POST 요청 받고
도착

form에 작성내용 담고
여기로 이동