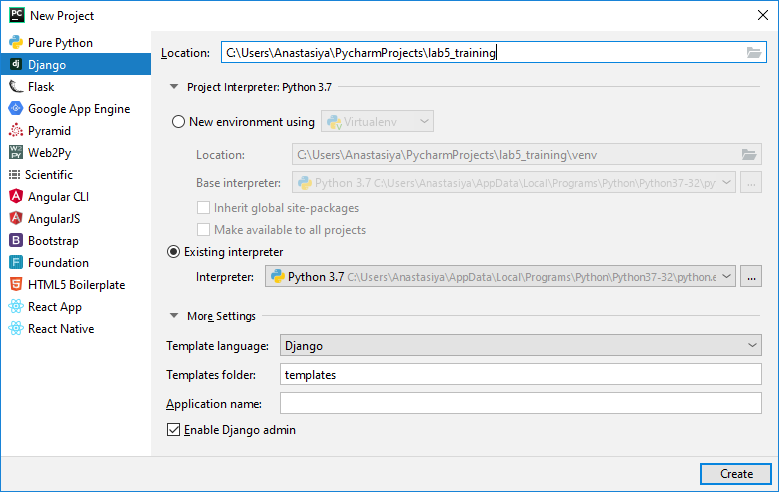
1. Создаем проект типа Django с имеющимся интерпретатором – в нем уже настроен Django 1.11.23.



1. **В терминале** пишем команду создания приложения:

python manage.py startapp riddles

1. Создаем тестовую функцию-вид (представление) для обработки запроса от клиента **в существующем файле** riddles/views.py:

from django.http import HttpResponse

def index(request):

return HttpResponse("Hello, World!")

1. **Создадим** файл riddles/urls.py.
2. Привяжем наш вид index из riddles/views.py к URL – **напишем** в riddles/urls.py

from django.conf.urls import url

from . import views

app\_name = 'riddles'

urlpatterns = [

url(r'^$', views.index, name='index')

]

1. Добавим обработку запросов к путям, начинающимся на "riddles", в соответствии с правилами из riddles/urls.py – **допишем** в существующем файле urls.py, **лежащем в корне проекта, куски выделенные жирным**:

from django.conf.urls import **include,** url

from django.contrib import admin

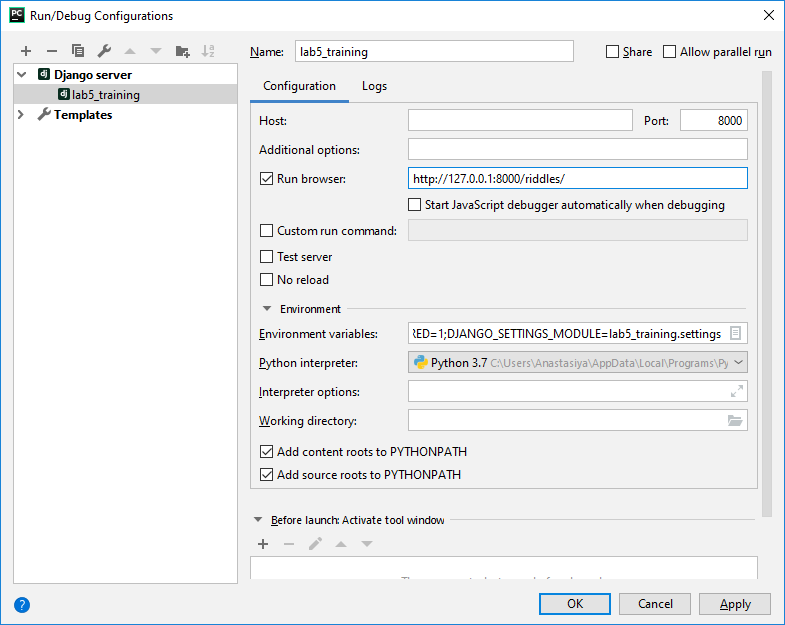
urlpatterns = [

**url(r'^riddles/', include('riddles.urls')),**

url(r'^admin/', admin.site.urls),

]

1. Запускаем, для проверки в адресной строке браузера после http://127.0.0.1:8000/ дописываем "riddles/", должны увидеть «Hello, World!»
2. Чтобы не переходить по ссылке каждый раз и не дописывать "riddles/" – зайдем возле зелененькой стрелочки для запуска проекта в выпадающий список, «Edit configurations…», ставим галку «Run browser…» и пишем в адресную строку «http://127.0.0.1:8000/riddles/»:



1. Cоздадим модели Riddle и Option: в Riddle будет содержаться загадка, в Option — один из возможных ответов на нее – напишем в существующий файл riddles/models.py:

from django.db import models

**class Riddle(models.Model):**

**riddle\_text = models.CharField(max\_length=255)**

**pub\_date = models.DateTimeField('date published')**

**class Option(models.Model):**

**riddle = models.ForeignKey(Riddle, on\_delete=models.CASCADE)**

**text = models.CharField(max\_length=255)**

**correct = models.BooleanField(default=False)**

1. Добавим наше приложение riddles в список установленных на платформу Django приложений – допишем строчку в существующий файл settings.py:

INSTALLED\_APPS = [

**'riddles.apps.RiddlesConfig',**

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

]

1. Создадим миграцию – код на языке SQL для создания в БД таблиц, соответствующих нашим моделям – напишем **в терминале**:

python manage.py makemigrations riddles

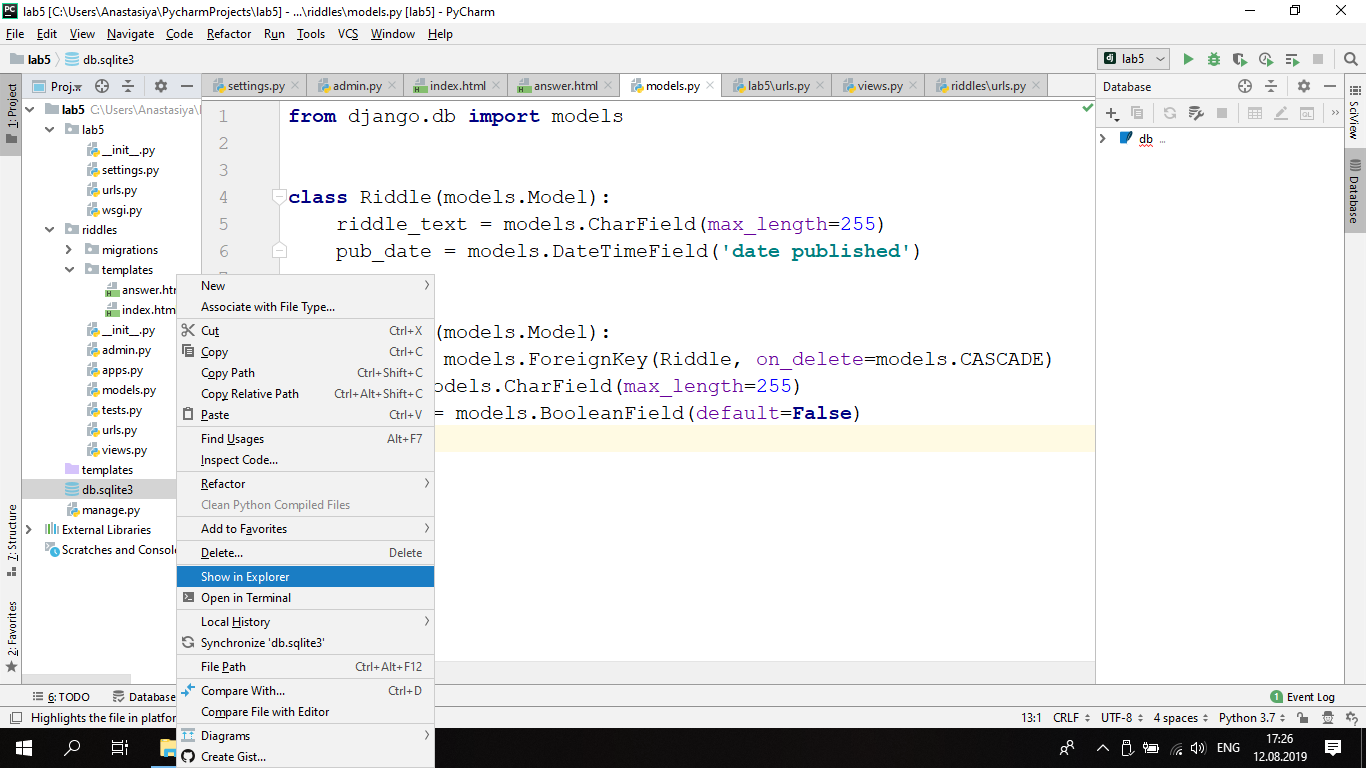
1. Выведем сгенерированный код на языке SQL в консоль – напишем **в терминале**:

python manage.py sqlmigrate riddles 0001

1. Выполним сгенерированный код на языке SQL – напишем **в терминале**:

python manage.py migrate

1. Теперь можно открыть папку с БД в Проводнике, нажать по ней правой кнопкой, открыть с помощью DB Explorer:



1. Дадим администраторам возможность наполнять созданные в БД таблицы – напишем в существующий файл riddles/admin.py

from django.contrib import admin

**from .models import Option, Riddle**

**admin.site.register(Riddle)**

**admin.site.register(Option)**

1. Для создания учетной записи администратора введем **в терминале**:

python manage.py createsuperuser

1. Запустим проект и для администрирования зайдем через браузер по адресу:

http://127.0.0.1:8000/admin/

1. Создадим в панели администрирования загадку и пару-тройку вариантов ответа к ней.
2. **Создадим** **папку** для шаблонов страниц: riddles/templates
3. **Создадим** **файл HTML** – макет главной страницы со списком ссылок на загадки – riddles/templates/index.html:

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>Title</**title**>  
</**head**>  
<**body**>  
**<h1>Available Riddles</h1>  
{% if message %}  
 <p><strong>{{ message }}</strong></p>  
{% endif %}  
{% if latest\_riddles %}  
 <ul>  
 {% for riddle in latest\_riddles %}  
 <li>  
 <a href="/riddles/{{ riddle.id }}/">  
 {{ riddle.riddle\_text }}  
 </a>  
 </li>  
 {% endfor %}  
 </ul>  
{% else %}  
 <p>No riddles are available right now.</p>  
{% endif %}**  
</**body**>  
</**html**>

1. **Создадим** **файл HTML** – макет страницы с выбором ответа на выбранную загадку – riddles/templates/answer.html:

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>Title</**title**>  
</**head**>  
<**body**>  
**<h1>{{ riddle.riddle\_text }}</h1>  
<p> <strong>{{ method }}</strong> </p>  
{% if error\_message %}  
 <p>  
 <strong>{{ error\_message }}</strong>  
 </p>  
{% endif %}  
<form action="answer/" method="post">  
 {% csrf\_token %}  
 {% for option in riddle.option\_set.all %}  
 <input type="radio" name="option" id="option{{ forloop.counter }}" value="{{ option.id }}" />  
 <label for="option{{ forloop.counter }}">**

**{{ option.text }}**

**</label><br>  
 {% endfor %}  
 <input type="submit" value="Answer" />  
</form>**  
</**body**>  
</**html**>

1. Напишем нормальные виды (представления) вместо тестового для рендеринга наших шаблонов – в существующем файле riddles/views.py:

**from** django.shortcuts **import** get\_object\_or\_404, render, redirect  
  
**from** .models **import** Riddle, Option  
  
  
*# главная страница со списком загадок***def** index(request):  
 message = **None  
 if "message" in** request.GET:  
 message = request.GET[**"message"**]  
 *# создание HTML-страницы по шаблону index.html  
 # с заданными параметрами latest\_riddles и message* **return** render(  
 request,  
 **"index.html"**,  
 {  
 **"latest\_riddles"**:  
 Riddle.objects.order\_by(**'-pub\_date'**)[:5],  
 **"message"**: message  
 }  
 )  
  
  
*# страница загадки со списком ответов***def** detail(request, riddle\_id):  
 error\_message = **None  
 if "error\_message" in** request.GET:  
 error\_message = request.GET[**"error\_message"**]  
 **return** render(  
 request,  
 **"answer.html"**,  
 {  
 **"riddle"**: get\_object\_or\_404(Riddle, pk=riddle\_id),  
 **"error\_message"**: error\_message  
 }  
 )  
*# продолжение – на следующей странице*

*# обработчик выбранного варианта ответа -  
# сам не отдает страниц, а только перенаправляет (redirect)  
# на другие страницы с передачей в GET-параметре  
# сообщения для отображения на этих страницах***def** answer(request, riddle\_id):  
 riddle = get\_object\_or\_404(Riddle, pk=riddle\_id)  
 **try**:  
 option = riddle.option\_set.get(pk=request.POST[**'option'**])  
 **except** (KeyError, Option.DoesNotExist):  
 **return** redirect(  
 **'/riddles/'** + str(riddle\_id) +  
 **'?error\_message=Option does not exist'**,  
 )  
 **else**:  
 **if** option.correct:  
 **return** redirect(  
 **"/riddles/?message=Nice! Choose another one!"**)  
 **else**:  
 **return** redirect(  
 **'/riddles/'**+str(riddle\_id)+  
 **'?error\_message=Wrong Answer!'**,  
 )

1. Пропишем, в ответ на какие URL должны вызываться созданные виды в файле riddles/urls.py:

**from** django.conf.urls **import** url  
  
**from** . **import** views  
  
app\_name = **'riddles'**urlpatterns = [  
 url(**r'^$'**, views.index, name=**'index'**),  
 url(**r'^([0-9]+)/$'**, views.detail, name=**'detail'**),  
 url(**r'^([0-9]+)/answer/$'**, views.answer, name=**'answer'**)  
]

1. Запускаем проект, ходим по ссылкам, проверяем работу ответов…