

# Лекция 6. Порядок разработки приложения Django



## Аннотация

В лекции будет рассмотрен порядок действий при разработке веб-приложения с использованием фреймворка Django.



# Спецификация

Структурированный набор требований к программному обеспечению и его внешним интерфейсам. Предназначен для того, чтобы установить базу для соглашения между заказчиком и разработчиком о том, как должен функционировать программный продукт.

Под **спецификацией** понимается формальное описание функций и данных программы, с которыми эти функции оперируют.

**Спецификация программы** – это описание задачи, которую решает программа. Слово "спецификация" буквально означает "описание" или "получение описания", а "специфицировать" значит "описывать".

В отличие от компьютерной программы спецификация обращена, прежде всего, к человеку и представляет собой описание в терминах, характерных для самой задачи, а не для ее реализации.

**Спецификация** – это задание для программиста, написанное постановщиком задачи.

**Спецификация** – это достаточно точное и достаточно полное описание задачи, которое человеку, участвующему в решении, легче написать, понять и прочесть, чем программисту представить решение этой задачи на доступном ему языке программирования.



[Правила составления Software requirements specification / Хабр \(habr.com\)](#)

**СОДЕРЖАНИЕ СПЕЦИФИКАЦИИ НА ПРОГРАММНЫЙ ПРОДУКТ**  
[spec-context.pdf \(specialist.ru\)](#)

[Технологии разработки программного обеспечения \(kstu.ru\)](#)

- Качество программных систем.
- Основные требования к программам, входящим в программную систему:
- Стадии разработки программного обеспечения.



# Шаблон спецификации требований к ПО

## Введение

- 1.1 Назначение
- 1.2 Соглашения, принятые в документах
- 1.3 Границы проекта
- 1.4 Ссылки

## 2. Общее описание

- 2.1 Общий взгляд на продукт
- 2.2 Классы и характеристики пользователей
- 2.3 Операционная среда
- 2.4 Ограничения дизайна и реализации
- 2.5 Предположения и зависимости

## 3. Функции системы

- 3.x Функция системы X
  - 3.x.1 Описание
  - 3.x.2 Функциональные требования

## 4. Требования к данным

- 4.1 Логическая модель данных
- 4.2 Словарь данных
- 4.3 Отчеты
- 4.4 Получение, целостность, хранение и утилизация данных

## 5. Требования к внешним интерфейсам

- 5.1 Пользовательские интерфейсы
- 5.2 Интерфейсы ПО
- 5.3 Интерфейсы оборудования
- 5.4 Коммуникационные интерфейсы

## 6. Атрибуты качества

- 6.1 Удобство использования
- 6.2 Производительность
- 6.3 Безопасность
- 6.4 Техника безопасности
- 6.x [Другие]

## 7. Требования по интернационализации и локализации

## 8. Остальные требования

## Приложение А. Словарь терминов

## Приложение Б. Модели анализа



## 1. Введение

Введение представляет собой обзор, помогающий читателям разобраться в структуре и принципе использования спецификации требований к ПО.

### 1.1 Назначение

**Определите продукт или приложение,** требования для которого указаны в этом документе, в том числе редакцию или номер выпуска. Если эта спецификация требований к ПО относится только к части системы, идентифицируйте эту часть или подсистему. Опишите типы читателей, которым адресован этот документ, например разработчикам, менеджерам проектов, маркетологам, пользователям, тестировщикам или составителям документации.

### 1.2 Соглашения, принятые в документах

**Опишите все стандарты или типографические соглашения,** включая значение стилей текста, особенности выделения или нотацию. Если вы нумеруете требования вручную, можно определить принятый формат на случай, если кому-нибудь позже понадобится добавить требование.

### 1.3 Границы проекта

**Кратко опишите ПО и его назначение.** Покажите, как связан продукт с пользователями или корпоративными целями, а также с бизнес-целями и стратегиями. Если имеется отдельный документ о концепции и границах проекта, не повторяйте его содержимое, а просто сошлитесь на него. Если спецификацию требований к ПО предполагается разрабатывать постепенно, она должна содержать собственное положение о концепции и границах продукта в качестве подраздела долгосрочной стратегической концепции. Можно предоставить высокоуровневую сводку главной функциональности выпуска или функций, которые он должен выполнять.

Подробнее:

[Шаблон спецификации требований к ПО - К. Вигерс | Бизнес-Анализ в России \(infozone.pro\)](http://infozone.pro)



# Приложения - Django Документация на русском (djangodoc.ru)

## СОДЕРЖАНИЕ

- [Приложения](#)
  - [Проекты и приложения](#)
  - [Настройка приложений](#)
    - [Для авторов приложений](#)
    - [Для пользователей приложения](#)
  - [Конфигурация приложения](#)
    - [Настраиваемые атрибуты](#)
    - [Атрибуты только для чтения](#)
    - [Методы](#)
    - [Пакеты пространства имен как приложения](#)
  - [Реестр приложений](#)
  - [Процесс инициализации](#)
    - [Как загружаются приложения](#)
    - [Поиск проблемы](#)



# 1. Определить цель разработки

Шаг 1. Следует составить начальный список вопросов, ответ на которые сможет получить пользователь, в процессе работы с приложением.

Например:

1. Список книг в хранилище
2. Информация о книге (Название, Автор, Жанр, Язык, Краткая аннотация, ISBN)
3. В каком количестве имеются книги
4. Книги каких авторов имеются
5. Книги каких жанров имеются
6. Книги каких авторов представлены
7. и т.д.

Список начальный, потому что в процессе разработки приложения возможно расширение списка вопросов, как и размера приложения за счет дополнения новых функций.





Шаг 2. Свести список составленных вопросов к одному предложению.  
Оно и будет формулировкой цели разработки.

Например:

Разработать приложение Django, дающее представление о наличии книг в Книжной лавке и о том, книги каких авторов и жанров имеются в хранилище лавки.



## 2. Определить, какие возможности предоставит пользователю приложение (функции приложения):

- 1.Посмотреть список всех книг
  - 2.Посмотреть список всех авторов
  3. Добавить автора в список
  4. Отредактировать данные автора
  - 5.Посмотреть данные конкретной книги
  - 6.Добавить книгу в список книг
- и т.д.



### 3. Разработать логическую модель данных приложения и определиться с выбором СУБД.

Шаг 1. Определиться, какая СУБД будет использоваться (SQLite, MySQL, др.), и понять особенности и требования, предъявляемые к модели данных данной СУБД (например, если реляционная модель, то каковы внутренние ограничения данной модели)

Шаг 2. Определить список информационных объектов, которые должны быть включены в модель, и их характеристики(свойства или атрибуты). Например:

Книга (Название книги, автор, жанр, язык, ISBN)

Жанр (Название жанра)

Автор (Имя автора, Фамилия автора, Дата рождения, Дата смерти, Краткая справка)

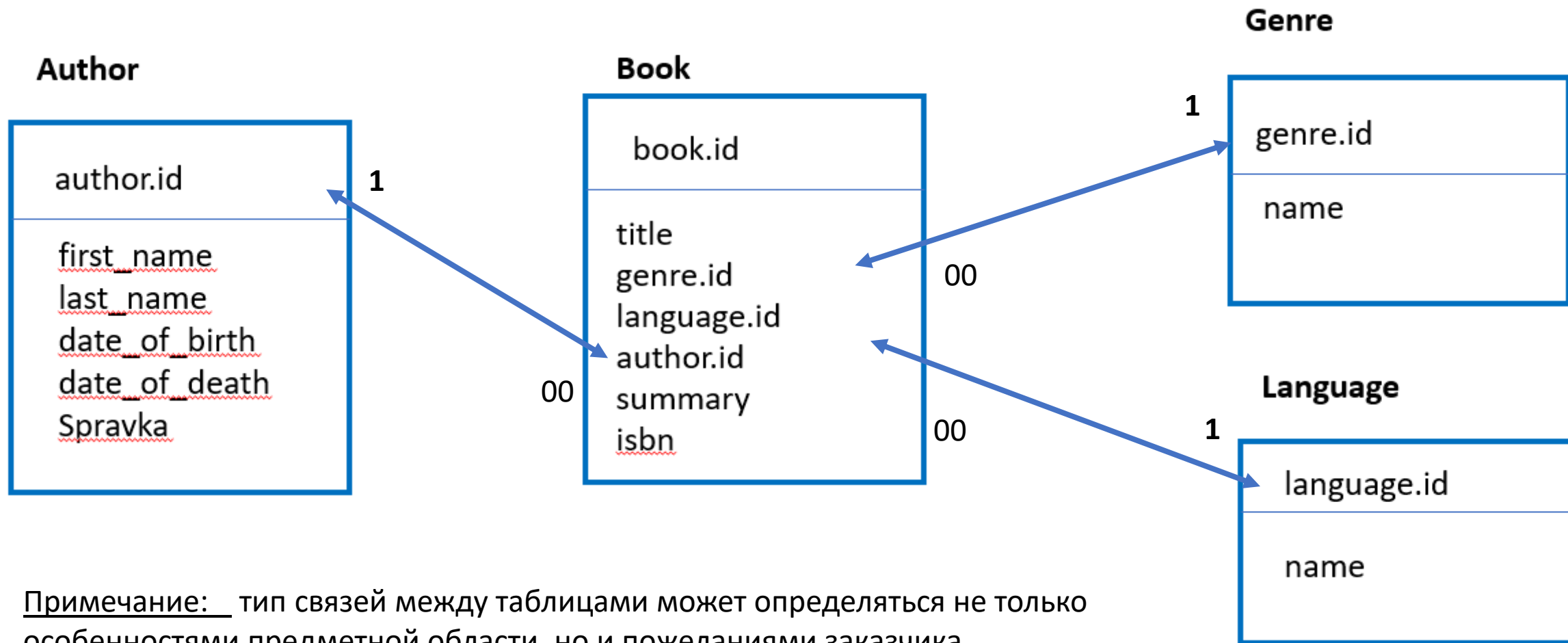
Язык (Название языка)

Шаг 3. Описать связи между информационными объектами.

Шаг 4. Представить данные каждого информационного объекта в виде таблицы и нормализовать таблицы до 3 НФ.



Например:



Примечание: тип связей между таблицами может определяться не только особенностями предметной области, но и пожеланиями заказчика.

Например, между таблицами Author и Book может быть установлена связь Многие-ко-Многим, ведь у книги может быть несколько авторов, так же и между таблицами Language и Book, ведь есть книги, изданные на двух языках, например, серии Билингва. Это же касается и жанра книги.



## 4. Определить название сайта и описать планируемый контент – какие страницы web-приложения необходимо будет создать.

Например:

Название сайта: Книжная лавка

Планируемые страницы сайта:

1. Главная страница сайта Книжная лавка (корень сайта)
  2. Страница с отображением списка книг
  3. Страница с отображением всех авторов
  4. Страница с возможностью добавить автора в список
  5. Страница с возможностью изменить данные автора
  6. Страница для отображения данных конкретной книги
  7. Страница с возможностью добавить книгу
- и т.д.



## 5. Определить структуру и список элементов для каждой страницы сайта, а также продумать, каким будет дизайн каждой страницы.

Например.

1. Главная страница сайта Книжная лавка (корень сайта) будет содержать:

- Заголовок
- Изображение
- Поясняющий текст
- Список пунктов меню

2. Страница со списком авторов будет содержать:

- Список авторов
- Кнопки-меню для выхода на страницы добавления и удаления автора



## 6. Приступить к начальному этапу разработки проекта.

1. В среде PyCharm создать проект.
2. Описать среду разработки: выбрать версию Python и подключить Django и другие необходимые библиотеки
3. Создать проект Django и запустить для проверки локальный отладочный сервер 127.0.0.1.8000.
4. Создать первое приложение проекта Django.

### Например

Создать в PyCharm проект, например, World Books и в контейнере этого проекта Django проект с именем, например, WebBooks.

Создать первое приложение с именем, например, catalog в составе Django проекта.



## 7. Зарегистрировать созданное приложение в списке `INSTALLED_APPS` в файле `settings.py`:

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'catalog'  
]
```





8. Приступить к формированию таблицы контента – файла `urls.py` и описать маршрут (`path`) к главной странице сайта.

```
urlpatterns = [  
    path('', views.index, name='index'),  
]
```

9. Приступить к формированию файла представлений `views.py` и описать первую функцию-представление, которая указана в первом маршруте в файле `urls.py`:

```
def index(request):  
    return HttpResponse('Это корневая страница сайта Книжная лавка')
```



## 10. Описать модель данных приложения в файле models.py:

```
# Create your models here.

class Genre(models.Model):
    name = models.CharField(max_length=200,
                             help_text=" Введите жанр книги", verbose_name="Жанр книги")

    def __str__(self):
        return self.name

class Language(models.Model):
    name = models.CharField(max_length=20,
                             help_text=" Введите язык книги", verbose_name="Язык книги")

    def __str__(self):
        return self.name
```



```
class Author(models.Model):
    first_name = models.CharField(max_length=100, help_text="Введите имя автора",
                                   verbose_name="Имя автора")
    last_name = models.CharField(max_length=100, help_text="Введите фамилию автора",
                                 verbose_name="Фамилия автора")
    date_of_birth = models.DateField(help_text="Введите дату рождения", verbose_name="Дата рождения",
                                     null=True, blank=True)
    date_of_death = models.DateField(help_text="Введите дату смерти", verbose_name="Дата смерти",
                                     null=True, blank=True)
    spravka = models.TextField(max_length=1000, help_text="Введите краткую справку", verbose_name="Краткая справка",
                              null=True, blank=True)

    def __str__(self):
        return self.last_name
```



```

class Book(models.Model):
    title = models.CharField(max_length=200,
                             help_text="Введите название книги",
                             verbose_name="Название книги")
    genre = models.ForeignKey('Genre', on_delete=models.CASCADE,
                              help_text=" Выберите жанр для книги",
                              verbose_name="Жанр книги", null=True)
    language = models.ForeignKey('Language',
                                  on_delete=models.CASCADE,
                                  help_text="Выберите язык книги",
                                  verbose_name="Язык книги", null=True)
    author = models.ManyToManyField('Author', help_text="Выберите автора книги", verbose_name="Автор книги", null=True)
    summary = models.TextField(max_length=1000,
                               help_text="Введите краткое описание книги",
                               verbose_name="Аннотация книги")
    isbn = models.CharField(max_length=4,
                            help_text="Должно содержать 4 символа",
                            verbose_name="ISBN книги")

    def display_author(self):
        return ', '.join([author.last_name for author in self.author.all()])
    display_author.short_description = 'Авторы'

    def get_absolute_url(self):
        # Возвращает url-адрес для доступа к определенному экземпляру книги.
        return reverse('book-detail', args=[str(self.id)])

    def __str__(self):
        return self.title

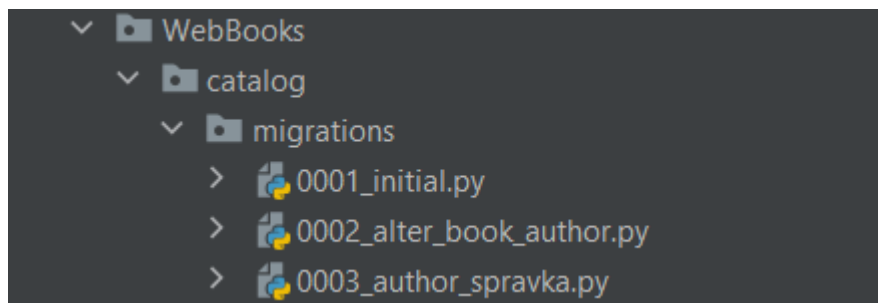
```



После того, как модель данных будет описана в файле `models.py`,  
Следует создать объекты модели (реально таблицы данных),  
создав и выполнив операцию миграции.

Напомню, миграция базы создается в момент создания и после каждого изменения структуры данных.

Все миграции сохраняются в файле `migrations` приложения и это дает возможность выполнить откат к предыдущему состоянию модели или можно удалить все миграции.



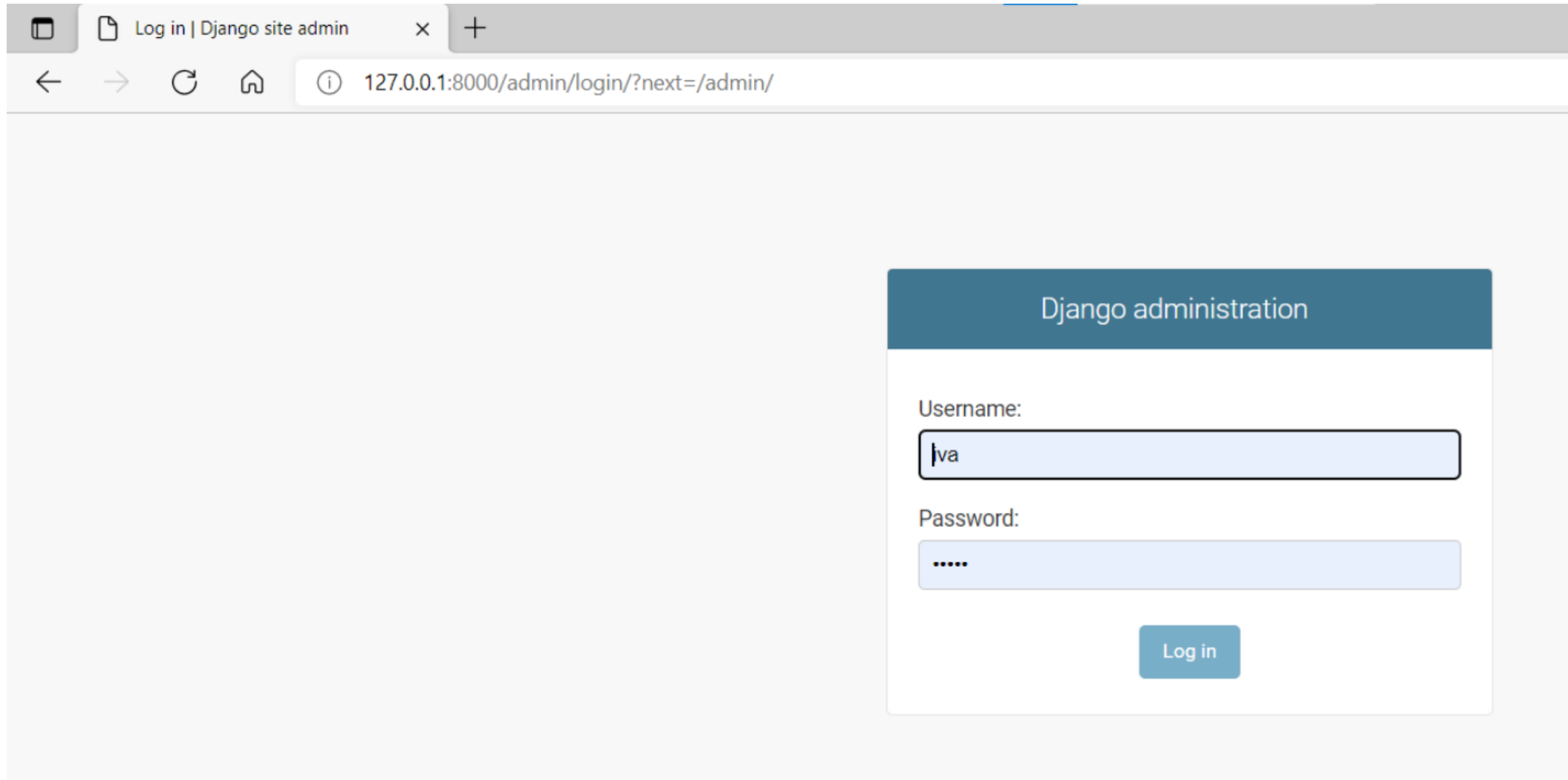
```
python manage.py migrate catalog 002
```

или

```
python manage.py migrate catalog zero
```



## 11. Подключить административную панель Django Admin и заполнить таблицы данными:



The screenshot shows a web browser window with a single tab titled "Log in | Django site admin". The address bar displays the URL "127.0.0.1:8000/admin/login/?next=/admin/". The main content area features a login form titled "Django administration". The form includes a "Username:" label, a text input field containing "va", a "Password:" label, and a password input field with masked characters ".....". A blue "Log in" button is positioned at the bottom right of the form.

Log in | Django site admin

127.0.0.1:8000/admin/login/?next=/admin/

Django administration

Username:

va

Password:

.....

Log in



## 12. Сформировать перечень и URL-адреса страниц сайта.

Например:

catalog/ - домашняя страница

catalog/books - список всех книг

catalog/authors - список всех авторов

catalog/book/<id> - просмотр данных конкретной книги

catalog/author/<id> - просмотр данных конкретного автора

и т.д.



## 13. Описать маршруты в файле urls.py:

```
from django.contrib import admin
from django.urls import path, include
from catalog import views
from django.conf.urls import url

urlpatterns = [
    path('', views.index, name='index'),
    path('authors_add/', views.authors_add, name='authors_add'),
    path('edit1/<int:id>/', views.edit1, name='edit1'),
    path('create/', views.create, name='create'),
    path('delete/<int:id>/', views.delete, name='delete'),
    path('admin/', admin.site.urls),
    url(r'^books/$', views.BookListView.as_view(), name='books'),
    path(r'^author/$', views.AuthorListView.as_view(), name='authors'),
    url(r'^book/(?P<pk>\d+)$', views.BookDetailView.as_view(), name='book-detail'),
]
```





## 14. Описать функции-представления в файле views.py:

```
def create(request):
    if request.method=="POST":
        author = Author()
        author.first_name = request.POST.get("first_name")
        author.last_name = request.POST.get("last_name")
        author.date_of_birth = request.POST.get("date_of_birth")
        author.date_of_death = request.POST.get("date_of_death")
        author.spravka = request.POST.get("spravka")
        author.save()
        return HttpResponseRedirect("/authors_add/")

def delete(request, id):
    try:
        author = Author.objects.get(id=id)
        author.delete()
        return HttpResponseRedirect("/authors_add/")
    except Author.DoesNotExist:
        return HttpResponseRedirect("<h2>Автор не найден</h2>")
```



## 15. Создать базовый шаблон и шаблон для главной страницы сайта.

```
1 {% extends "base_generic.html" %}
2
3 {% block content %}
4 <h1>Главная страница</h1>
5
6     <p>Добро пожаловать на сайт <em>Книжная лавка!</em></p>
7     <p>Здесь Вы найдете много замечательных книг Ваших любимых авторов</p>
8
9 <h2>Обновляемый контент</h2>
10
11     <p>База данных сайта Книжная лавка содержит следующее количество записей: </p>
12     <ul>
13         <li><strong>Количество книг:</strong> {{ num_books }} </li>
14         <li><strong>Количество авторов книг:</strong> {{ num_authors }} </li>
15     </ul>
16 {% endblock %}
```



## 16. На основе базового шаблона сформировать шаблоны для вывода других страниц сайта.

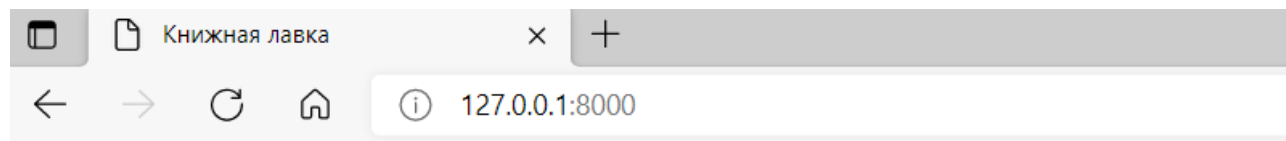
```
{% extends "base_generic.html" %}

{% block content %}
    <h1>Список книг в БД</h1>
    {% if book_list %}
        <ul>
            {% for book in book_list %}
                <li>
                    <a href="{|Book.get_absolute_url |}">{{book.title}} </a> ({{book.genre}})
                    <a href="/book/update/{{book.id}}">Изменить</a>
                    <a href="/book/delete/{{book.id}}">Удалить</a>
                </li>
            {% endfor %}
        </ul>
    {% else %}
        <p>В базе данных нет книг</p>
    {% endif %}
{% endblock %}
```



Далее вести разработку сайта, дополняя приложение новыми элементами: маршрутами, функциями-представлениями, шаблонами, формами и т.д.





# "Книжная лавка"

- [Главная страница](#)
- [Все книги](#)
- [Все авторы](#)
- [Редактировать авторов](#)
- [Добавить книгу](#)

## Главная страница

Добро пожаловать на сайт *Книжная лавка!*

Здесь Вы найдете много замечательных книг Ваших любимых авторов

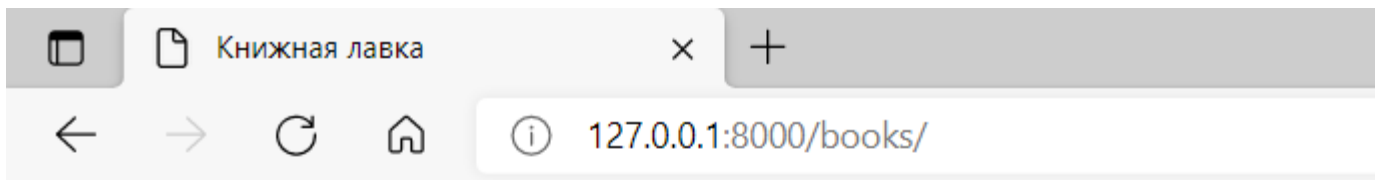
## Обновляемый контент

База данных сайта Книжная лавка содержит следующее количество записей:

- **Количество книг: 5**
- **Количество авторов книг: 8**

ООО "Книжная лавка", 2021.





# "Книжная лавка"

- [Главная страница](#)
- [Все книги](#)
- [Все авторы](#)
- [Редактировать авторов](#)
- [Добавить книгу](#)

## Список книг в БД

- [Три мушкетера](#) (Приключения) [Изменить](#) [Удалить](#)
- [Графиня де Монсоро](#) (Приключения) [Изменить](#) [Удалить](#)
- [Война и мир](#) (Проза) [Изменить](#) [Удалить](#)
- [Щелкунчик и мышинный король](#) (Сказка) [Изменить](#) [Удалить](#)
- [Хроники заводной птицы](#) (Проза) [Изменить](#) [Удалить](#)

ООО "Книжная лавка", 2021.





# "Книжная лавка"

- [Главная страница](#)
- [Все книги](#)
- [Все авторы](#)
- [Редактировать авторов](#)
- [Добавить книгу](#)

## Список авторов в БД

- [Александр Пушкин](#)

Справка - 26 мая [6 июня] 1799, Москва — 29 января [10 февраля] 1837, Санкт-Петербург) — русский поэт, драматург и прозаик, заложивший основы русского реалистического направления[2], литературный критик[3] и теоретик литературы, историк[3], публицист, журналист[3]; один из самых авторитетных литературных деятелей первой трети XIX века. Ещё при жизни Пушкина сложилась его репутация величайшего национального русского поэта[4][5]. Пушкин рассматривается как основоположник современного русского литературного языка

- [Лев Толстой](#)

Справка - 28 августа [9 сентября] 1828, Ясная Поляна, Тульская губерния, Российская империя — 7 [20] ноября 1910, станция Астапово, Рязанская губерния, Российская империя) — один из наиболее известных русских писателей и мыслителей, один из величайших писателей-романистов мира[4]. Участник обороны СевастополяПерейти к разделу «#Военная служба». Просветитель, публицист, религиозный мыслитель, его авторитетное мнение послужило причиной возникновения нового религиозно-нравственного течения — толстовстваПерейти к разделу «#Философия». За свои взгляды был отлучён от церкви. Член-корреспондент Императорской Академии наук (1873), почётный академик по разряду изящной словесности (1900)[5]. Был номинирован на Нобелевскую премию по литературе (1902, 1903, 1904, 1905). Впоследствии отказался от дальнейших номинаций. Классик мировой литературы.

- [Перси Бишп Шелли](#)

Справка - (англ. Percy Bysshe Shelley; 4 августа 1792, графство Суссекс — 8 июля 1822, утонул в Средиземном море между Специей и Ливорно) — английский писатель, поэт и эссеист. Один из классиков британского романтизма[5]. Был женат на Мэри Шелли.

- [Эрнст Теодор Амадей Гофман](#)

Справка - нем. Ernst Theodor Wilhelm Hoffmann; произ. Хофман; 24 января 1776, Кёнигсберг, Королевство Пруссия — 25 июня 1822, Берлин, Королевство Пруссия) — немецкий писатель-романтик, сказочник, композитор, художник, юрист. Из преклонения перед Амадеем Моцартом в 1805 году сменил имя «Вильгельм» на «Амадей» (Amadeus). Заметки о музыке публиковал под именем Иоганн Крайслер (нем. Johannes Kreisler).

- [Харуки Мураками](#)



# "Книжная лавка"

- [Главная страница](#)
- [Все книги](#)
- [Все авторы](#)
- [Редактировать авторов](#)
- [Добавить книгу](#)


## Список авторов в БД


id	Имя	Фамилия	
2	Александр	Пушкин	<a href="#">Изменить</a> , <a href="#">Удалить</a>
3	Лев	Толстой	<a href="#">Изменить</a> , <a href="#">Удалить</a>
4	Перси Бишп	Шелли	<a href="#">Изменить</a> , <a href="#">Удалить</a>
5	Эрнст Теодор Амадей	Гофман	<a href="#">Изменить</a> , <a href="#">Удалить</a>
6	Харуки	Мураками	<a href="#">Изменить</a> , <a href="#">Удалить</a>
16	Юлий	Цезарь	<a href="#">Изменить</a> , <a href="#">Удалить</a>
21	Фея Прекрасная	Лесная	<a href="#">Изменить</a> , <a href="#">Удалить</a>
22	Александр	Дюма-отец	<a href="#">Изменить</a> , <a href="#">Удалить</a>

## Добавить в БД автора книги

Имя автора:

Фамилия автора:

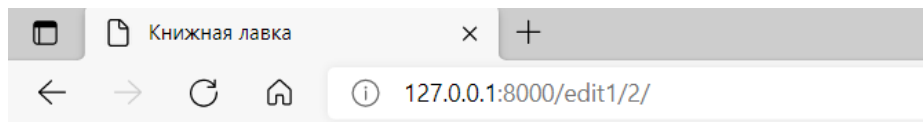
Дата рождения:  

Дата смерти:  

Краткая справка:







## "Книжная лавка"

- [Главная страница](#)
- [Все книги](#)
- [Все авторы](#)
- [Редактировать авторов](#)
- [Добавить книгу](#)

### Список авторов в БД

#### Александр Пушкин

Измените имя

Измените фамилию

Измените дату рождения

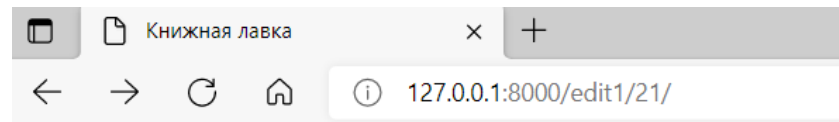


Измените дату смерти



Измените справку

ООО "Книжная лавка", 2021.



## "Книжная лавка"

- [Главная страница](#)
- [Все книги](#)
- [Все авторы](#)
- [Редактировать авторов](#)
- [Добавить книгу](#)

### Список авторов в БД

#### Фея Прекрасная Лесная царица

Измените имя

Измените фамилию

Измените дату рождения



Измените дату смерти



Измените справку

ООО "Книжная лавка", 2021.





# "Книжная лавка"

- [Главная страница](#)
- [Все книги](#)
- [Все авторы](#)
- [Редактировать авторов](#)
- [Добавить книгу](#)

**Название книги:**   
Введите название книги

**Жанр книги:**   
Выберите жанр для книги

**Язык книги:**   
Выберите язык книги

**Автор книги:**   
Цезарь  
Лесная царица  
Дюма-отец  
Выберите автора книги

**Аннотация книги:**

Введите краткое описание книги

**ISBN книги:**   
Должно содержать 4 символа



catalog_book (db0)						
Структура Данные Ограничения Индексы Триггеры DDL						
Табличный вид Форма						
Отфильтровать да... Всего загружено строк: 6						
id	title	summary	isbn	genre_id	language	
1	1 Три мушкетера	Историко-приключенческий роман Александра Дюма-отца, впервые опубликованный в парижской ...	1234567891234	4		
2	2 Графиня де Монсоро	Роман Александра Дюма-отца, созданный в соавторстве с Огюстом Маке и опубликованный в 1846 году...	1019283745622	4		
3	3 Война и мир	Роман-эпопея Льва Николаевича Толстого, описывающий русское общество в эпоху войн против ...	0987654321123	1		
4	4 Щелкунчик и мышиный король	Рождественская повесть-сказка Эрнста Теодора Амадея Гофмана, опубликованная в сборнике «Детские ...	2223334445551	7		
5	5 Хроники заводной птицы	Роман японского писателя Харуки Мураками, написан в 1992-1995 годах. В центре произведения - ...	1111122222333	1		
6	6 Записки о галльской войне	Сочинение Гая Юлия Цезаря, в восьми книгах которого он в присущей ему точной, сжатой и энергично...	1234	1		



Когда разработку приложения можно считать законченной?

Наверное, когда приложение выполнено в запланированном объеме.

Возможности для развития бесконечны.

Можно изменять и развивать структуру и содержимое базы данных

Можно создавать новые страницы, шаблоны и формы.

Можно создавать новые функции для работы с приложением.



Когда разработка приложения и отладка его на локальном сервере завершена, можно развернуть приложение на стороннем хосте для функционирования в “боевом” режиме.

[Развертывание приложений Django / Хабр \(habr.com\)](#)

[Django Руководство часть 11: Разворачивание сайта на сервере - Изучение веб-разработки | MDN \(mozilla.org\)](#)

[Django - Урок 001. Развёртывание сайта на Django + PostgreSQL + Gunicorn + Nginx \(evileg.com\)](#)



Спасибо за внимание!

