

# Кастомизация админ панели

**№ урока:** 3    **Курс:** Django Essential

**Средства обучения:** Pycharm Community, GIT, аккаунт в GitHub, Python 3.6 и выше, Django проект, Docker образ.

## Обзор, цель и назначение урока

Научиться настраивать админ-панель. После урока вы сможете использовать фильтры в админ-панели, формировать макеты и настраивать списки. А также получите базовые навыки в использовании стилей в админ-панели, что в будущем даст возможность создать индивидуальный стиль под каждого клиента.

## Изучив материал данного занятия, учащийся сможет:

- Создавать суперпользователя со статусом "сотрудник".
- Добавлять в админ-панель любую модель.
- Настраивать отображение списков.
- Добавлять фильтры в админ-панель.
- Настраивать макеты в админ-панели, а также изменить их отображение.
- Добавлять встроенное редактирование записей.
- Изменять стиль админ-панели с использованием html, css и дополнительных приложений.

## Содержание урока

1. Запуск проекта
2. Создание суперпользователя как сотрудника
3. Добавление в админ-панель моделей
4. Вход в админ панель, обзор всех элементов
5. Настройка ModelAdmin
6. Создание отображения списков
7. Фильтры
8. Создание кастомного макета
9. Добавление и настройка встроенного редактирования полей
10. Стили в админ-панели: media, пакеты для админпанелей

## Резюме

- Django админ панель служит для упрощения работы с моделями: для их создания, просмотра, обновления или удаления.
- Чтобы создать пользователя как сотрудника используем ключевой параметр "staff".
- Для добавления в админ-панель нужно добавить модель в файл admin.py. Для этого используем функцию `admin.site.register()`.
- Еще один способ: используя класс `ModelAdmin` и декоратор `@admin.register()`. Этот способ более продвинутый и дает больше возможностей.
- Ключевые элементы админ-панели: Add, Save, Save and add another, или Save and continue editing, Delete, HISTORY и VIEW ON SITE. А также связи между моделями.
- Для настройки отображения списков используем ключевой атрибут `list_display`.
- Нельзя использовать m2m поля в `list_display`.
- Для большей гибкости можно использовать функции.

- Для настройки фильтров используем ключевой атрибут `list_filter`.
- Для настройки макетов - `fields` или `fieldsets`.
- Чтобы добавить встроенное редактирование - используем атрибут `inlines`.
- Чтобы изменить внешний вид можно использовать свой `html` и `css` или использовать пакеты.
- Для своих файлов используем параметр `ADMIN_TOOLS_THEMING_CSS` в настройках.
- Используем `django-admin-tools` для улучшения админ-панели.

### Закрепление материала

- Какая структура админ-панели?
- Какие есть способы добавления модели в админ-панель?
- Какие атрибуты нужно добавить, чтобы настроить отображение списков и фильтров?
- Для чего используется атрибут `fields` и какая его структура?
- Назовите 5 пакетов для изменения стиля админ-панели.

### Дополнительное задание

#### Задание

Настроить админ-панель по максимуму: добавить списки, фильтры, встроенное редактирование, а также изменить стиль.

### Самостоятельная деятельность учащегося

#### Задание 1

Выучить и понять все возможности изменения админ-панели, которые были рассмотрены на уроке.

#### Задание 2

Создать админ-панель для интернет-магазина. Добавить модели с товарами, категориями и покупками. К ним добавить фильтры по цене, дате и количеству, добавить встроенное редактирование к товару.

#### Задание 3

Найдите и подключите к проекту пакет стилей, не используя стиль, который был рассмотрен на уроке.

### Рекомендуемые ресурсы

Django документация:

- <https://djangopackages.org/grids/g/admin-interface/>
- <https://docs.djangoproject.com/en/2.2/ref/contrib/admin/>
- <https://docs.djangoproject.com/en/2.2/intro/tutorial02/#introducing-the-django-admin>