

Настройка Админ Панели

ModelAdmin опции:

actions: tuple[object] - список действий (функций), доступных на странице списка изменений

action_on_top: bool – по умолч. True – расположение панели действий сверху

action_on_bottom: bool – по умолч. False – расположение панели действий снизу

action_selection_counter: bool – по умолч. True – отображение счетчика выбора

date_hierarchy: str – имя атрибута типа DateField или DateTimeField для навигации по дате

empty_value_display: str – значение отображения для пустых полей

exclude: tuple[str] – список/кортеж имен атрибутов, которые следует исключить из формы создания записи (прим. id)

fields: tuple[str] – список/кортеж имен атрибутов, которые будут отображаться в форме создания записи (обратное от exclude)

fieldsets: tuple[tuple[str, dict]] – кортеж/список кортежей по 2 элемента: 1 – название раздела, 2 – словарь с настройкой отображения (

допустимые ключи:

fields – кортеж/список имен атрибутов отображаемых в этом разделе

description – описание раздела

classes – список/кортеж имен CSS классов)

list_display: tuple[str] – список/кортеж имен атрибутов, отображаемых в списке

inlines: tuple[object] – список объектов (классов) типа TabularInlines или StackedInline для совместного отображения (применяется при отображении связанных таблиц)

list_display_links: tuple[str] – список/кортеж имен атрибутов, связанных с страницей “изменения” для объекта

list_editable: tuple[str] – список/кортеж имен атрибутов, которые можно изменять в списке записей (нельзя указать атрибут, который не указан в list_display, а так же нельзя указать атрибут указанный в list_display_links)

list_filter: tuple[str] – список/кортеж имен атрибутов, для активации фильтрации в боковой панели

list_max_show_all: int – количество записей отображаемых за раз при нажатии на кнопку “Показать все”

list_per_page: int – количество записей, отображаемых на одной странице

ordering: tuple[str] – список/кортеж имен атрибутов для указания сортировки, наличие “-” перед именем атрибута, указывает на обратный порядок сортировки

paginator: object – класс пагинатора, который будет использоваться для пагинации

prepopulated_fields: dict[str, tuple[str]] – словарь, с указанием атрибутов, для предварительного заполнения на основании других атрибутов (Нельзя указать атрибуты типов *DateTimeField*, *ForeignKey*, *OneToOneField*, *ManyToManyField*)

preserve_filters: bool – по умолч. True – сохранения фильтров после выхода со страниц создания/редактирования

readonly_fields: tuple[str] – список/кортеж имен атрибутов, недоступных для редактирования

show_full_result_count: bool – отображение полного количества записей, при выборке записей

InlineModelAdmin опции:

model: object – модель, которая использует инлайн (обязательный атрибут)

fk_name: str – имя внешнего ключа (если их несколько)

classes: tuple[str] – список/кортеж CSS классов

max_num: int – максимальное количество форм, которые будут отображаться во вставке

min_num: int – минимальное количество форм

verbose_name: str – переопределение `verbose_name` из класса `Meta` модели

verbose_name_plural: str – переопределение `verbose_name_plural` из класса `Meta` модели

can_delete: bool – можно ли удалить объект в инлайне

show_change_link: bool – будет ли у инлайн объекта ссылка на форму изменения

AdminSite общая кастомизация админ панели, опции:

site_header: str – заголовок размещенный в верхней части панели

site_title: str – заголовок, отображаемый в каждой вкладке в конце

site_url: str – ссылка на кнопке “Просмотр сайта”, по умолч. “/”

index_title: str – текст для размещения в верхней части индексной страницы администратора

empty_value_display: str - значение для пустых полей, действует на модели, у которых данный атрибут не переопределен

enable_nav_sidebar: bool – отображение боковой навигационной панели

Django ORM Query

Вход в SHELL: *python manage.py shell*

Model.objects.all() - запрос на получение всех записей

Фильтрация

Model.objects.filter(условие) – выборка с условием

Ограничения:

__lt, __lte, __gt, __gte, __startswith – добавляются к названию атрибута

OR:

Model.objects.filter(условие | Model.objects.filter(условие)) либо используя Q

from django.db.models import Q

Model.objects.filter(Q(условие) | Q(условие))

AND:

Model.objects.filter(условие, условие) либо

Model.objects.filter(условие) & Model.objects.filter(условие) либо

Model.objects.filter(Q(условие) & Q(условие))

NOT: ~Q

Выборка некоторых атрибутов:

Model.objects.filter(условие).values(имена атрибутов)

JOIN:

Model.objects.select_related(внешний ключ)

прим. *Product.objects.select_related('category')* либо

Model.objects.filter(внешний ключ__атрибут связанной таблицы условие)

прим. *Product.objects.filter(category__name='Food')*

LIMIT:

Добавление среза к выборке, добавляет атрибут LIMIT к SQL запросу

Model.objects.filter(условие)[:5]

ORDER BY:

Model.objects.filter(условие).order_by(атрибут), перед именем атрибута, можно поставить “-”, что изменит сортировку на обратную