

УДК 004.423

БИБЛИОТЕКА МРТТ ДЛЯ ФРЕЙМВОРКА DJANGO В PYTHON

Халевин Т.А.

*студент направления подготовки информатики и вычислительной техники,
Хакасский государственный университет имени Н.Ф. Катанова,
г. Абакан, Россия ¹*

Аннотация: В данной статье рассматривается использование библиотеки МРТТ (Modified Preorder Tree Traversal) для работы с древовидными структурами в веб-приложениях на основе фреймворка Django. Описывается процесс установки и настройки МРТТ, создание моделей с иерархиями, регистрация в административной панели Django и основные возможности библиотеки для управления иерархическими данными. Приведены примеры кода, иллюстрирующие создание вложенных моделей, а также обсуждаются преимущества использования МРТТ в проектах Django.

Ключевые слова: Django, МРТТ, древовидные структуры, иерархические модели, иерархии данных, административная панель, веб-приложение, вложенные данные.

MPTT LIBRARY FOR THE DJANGO FRAMEWORK IN PYTHON

Khalevin T.A.

*student of computer science and computer engineering department,
N.F. Katanov Khakass State University,
Abakan, Russia*

Abstract: This article discusses the use of the MPTT (Modified Preorder Tree Traversal) library for working with tree structures in web applications based on the

¹ Научный руководитель: Голубничий А.А. старший преподаватель кафедры ПОВТиАС, Хакасский государственный университет имени Н.Ф. Катанова, г. Абакан, Россия

Django framework. It describes the process of installing and configuring MPTT, creating models with hierarchies, registration in the Django administration panel and the basic capabilities of the library for managing hierarchical data. Examples of code examples illustrating the creation of nested models, and also discusses the advantages of using MPTT in Django projects.

Keywords: Django, MPTT, tree structures, hierarchical structures, hierarchical models, data hierarchies, admin panel, web application, nested data.

В данной статье будет рассмотрена библиотека MPTT для фреймворка Django и её функциональность. Данная статья не будет углубляться во все функции библиотеки MPTT, а только рассмотрит самые основные и базовые моменты при работе с данной библиотекой. MPTT (Modified Preorder Tree Traversal) – это библиотека для Django, которая предоставляет инструменты для работы с древовидными структурами данных, такими как иерархии или вложенные объекты [1].

Библиотека MPTT полезна в тех случаях, когда необходимо хранить и обрабатывать данные, которые организованы в виде деревьев, например, категории, комментарии с вложенными ответами или другие типы вложенных объектов.

Основные особенности MPTT:

- MPTT позволяет создавать, изменять иерархии, а также эффективно извлекать связанные данные.
- MPTT использует модифицированный алгоритм обхода дерева для оптимизации операций с деревом. Это означает, что получение всех потомков или предков узла осуществляется быстро.
- Библиотека полностью совместима с Django ORM, что позволяет использовать ее вместе с другими инструментами Django.
- MPTT предоставляет удобные методы и свойства, такие как получение корневых элементов, листьев, потомков, предков и так далее.

Для установки библиотеки МРТТ в проект Django используется менеджер пакетов `pip`. Следует убедиться, что у вас установлена актуальная версия Python и Django. Выполните команду `pip install django-mptt`. Если вы используете виртуальное окружение, не забудьте активировать его [2].

После установки библиотеки МРТТ, создадим простое приложение Django командами в терминале, представленными на рисунке 1.

```
django-admin startproject lib_mptt
cd lib_mptt/
python manage.py startapp car
```

Рисунок 1 – Создание простого приложения Django [разработано автором]

В данном коде создается Django проект, и создается дополнительное приложение `car`. Вы можете назвать проект и приложение по своему усмотрению [3].

Теперь необходимо добавить наше приложение и библиотеку в `INSTALLED_APPS` Django. Для этого необходимо перейти в файл `settings.py` и добавить их. Пример представлен на рисунке 2.

```
# settings.py
INSTALLED_APPS = [
    ...,
    'car',
    'mptt',
    ...
]
```

Рисунок 2 – Добавление приложения и библиотеки [разработано автором]

Создадим модель, которая будет иметь иерархическую структуру `CarCategory`, которая представляет категории автомобилей, поддерживая вложенность. Пример такой модели представлен на рисунке 3.

```
from django.db import models
from mptt.models import MPTTModel, TreeForeignKey

# Модель для представления категорий автомобилей с поддержкой MPTT
class CarCategory(MPTTModel):
    name = models.CharField(max_length=100) # Название категории
    parent = TreeForeignKey(
        'self', # Связь с самой собой для поддержки иерархии
        on_delete=models.CASCADE,
        null=True, # Корневые узлы могут не иметь родителя
        blank=True,
        related_name='children'
    )

    class MPTTMeta:
        | order_insertion_by = ['name'] # Задать порядок вставки

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

Рисунок 3 – Модель CarCategory [разработано автором]

Теперь модель CarCategory создана и использует MPTT для поддержки иерархий. Основные моменты:

- Поле parent позволяет определять родительские и дочерние категории.
- Класс MPTTMeta определяет дополнительные метаданные, в нашем случае это порядок вставки.

Теперь необходимо добавить отображение данной модели в админ панели Django. Для этого регистрируем её в нашем проекте используя код, представленный на рисунке 4.

```
from django.contrib import admin
from .models import CarCategory

admin.site.register(CarCategory)
```

Рисунок 4 – Регистрация модели в админ панели [разработано автором]

Применим миграции к базе данных и создадим суперпользователя. Для этого необходимо выполнить команды в терминале, представленные на рисунке 5.

```
python manage.py makemigrations
python manage.py migrate
python manage.py createsuperuser
# Заполняете данные вашего суперпользователя
python manage.py runserver
```

Рисунок 5 – Применение миграций к базе данных и создание суперпользователя.

Теперь при переходе в админ панель Django по адресу <http://127.0.0.1:8000/admin/>, мы увидим созданную модель. Создадим базовый экземпляр, и несколько его наследников. Результат представлен на рисунке 6.

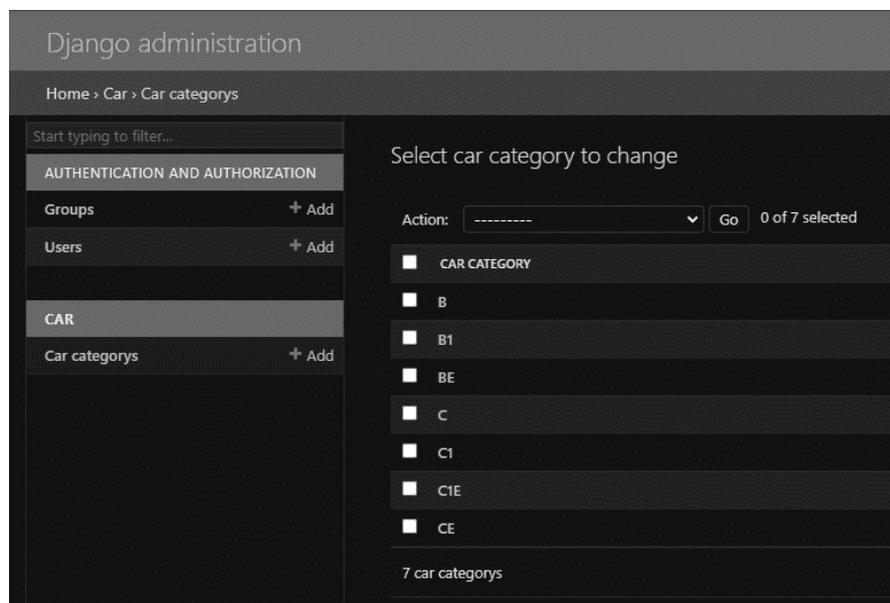


Рисунок 6 – Отображение экземпляров модели CarCategory [разработано автором]

При таком отображении экземпляров модели, мы не видим вложенность моделей. Какая модель наследуется от какой? При таком отображении мы будем вынуждены заходить в каждый экземпляр и смотреть от кого он наследуется. Чтобы упростить этот процесс, библиотека МРТТ предоставляет специальный класс `МРТТModelAdmin` для работы с иерархическими моделями в админ панели. Он позволяет удобно отображать древовидные структуры в административной панели. Для этого необходимо немного изменить регистрацию модели как это показано на рисунке 7.

```
from django.contrib import admin
from mptt.admin import MPTTModelAdmin
from .models import CarCategory

# Регистрация модели CarCategory с использованием MPTTModelAdmin
admin.site.register(CarCategory, MPTTModelAdmin)
```

Рисунок 7 – Измененная регистрация модели в админ панели [разработано автором]

Проверим результат на сайте, для этого достаточно обновить страницу. Результат представлен на рисунке 8.

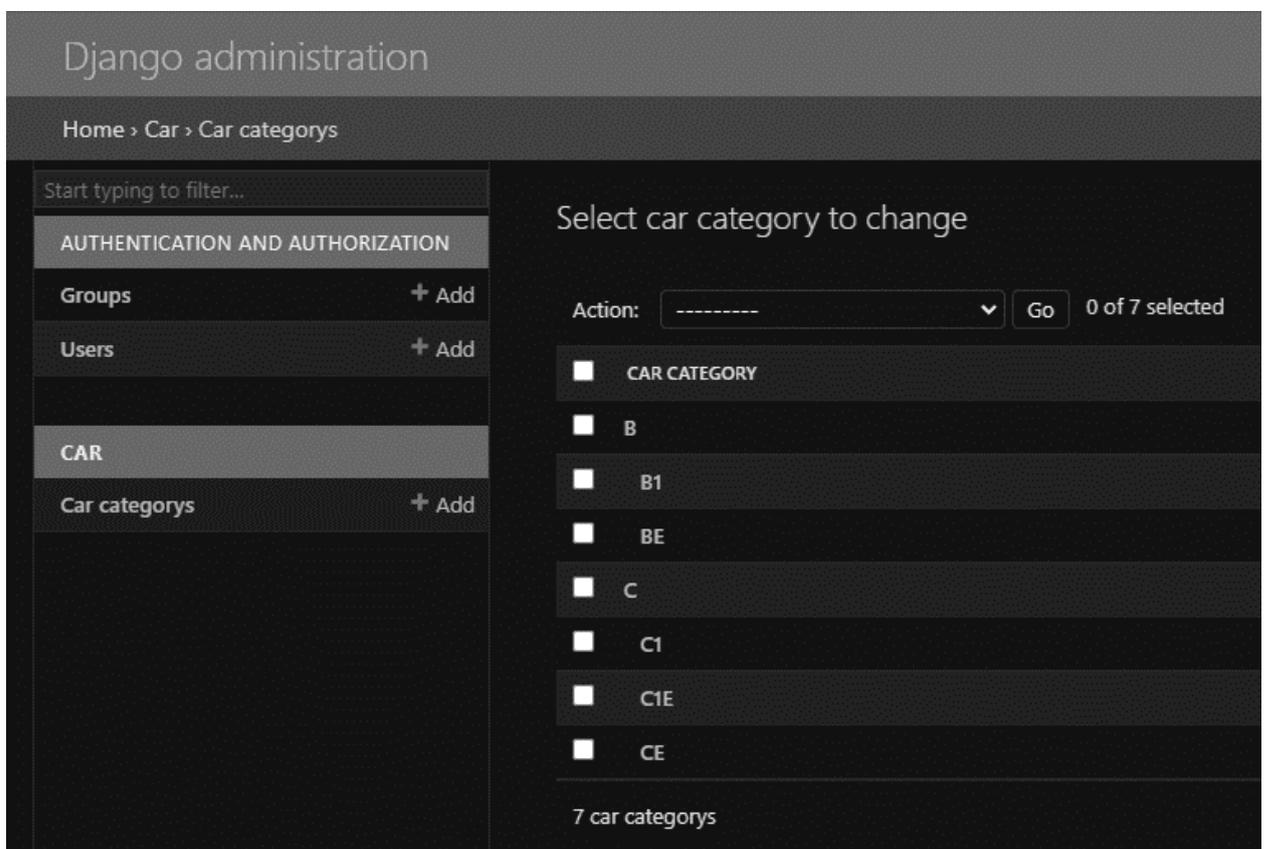


Рисунок 8 – Измененное отображение экземпляров модели CarCategory [разработано автором]

Теперь модели наследники находятся под своим родителем и отображаются с отступом.

Библиотека MPTT хорошо интегрируется с Django ORM, что позволяет использовать ее вместе с другими инструментами Django, такими как админка, формы и запросы. Благодаря оптимизированному алгоритму обхода дерева,

операции с иерархическими структурами остаются эффективными даже при больших объемах данных.

В заключение, МРТТ – это отличный выбор для разработчиков Django, которым необходимо работать с иерархическими данными. Благодаря простоте использования, эффективности и интеграции с Django, библиотека может значительно упростить задачи, связанные с деревьями, и предоставить мощные инструменты для управления сложными иерархиями.

Библиографический список:

1. Гуриков, С.Р. Основы алгоритмизации и программирования на Python [Текст] / С.Р. Гуриков. – М.: ИНФРА-М, 2023. – 343 с.
2. Гэддис Т. Начинаем программировать на Python [Текст] / Т. Гэддис. – СПб.: БХВ-Петербург, 2021. – 768 с.
3. Меле А. Django 4 в примерах. Разрабатывайте мощные и надежные веб-приложения на Python с нуля [Текст] / А. Меле – М.: ДМК-Пресс, 2023.

Оригинальность 88%