

ТЕХНИЧЕСКИЕ НАУКИ



УДК 621.314

Совместная работа Django и Apache: анализ и пути решения ошибок по лог-файлам**Е.Д. Хирова, Д.А. Каменский**

Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

Аннотация

В процессе настройки совместной работы Django и Apache возникают различные ошибки, которые фиксируются в лог-файлах Apache. Цель статьи заключается в анализе возникающих ошибок и предложении эффективных методов их устранения. В работе последовательно рассмотрены следующие ошибки: отказ в доступе (403), несовместимость с под-интерпретаторами Python, ошибки импорта модулей и некорректная работа с путями файловой системы. Для каждой ошибки проанализированы причины возникновения и предложены практические способы их устранения. Представлено систематизированное руководство по отладке ошибок, что сокращает время настройки совместной работы Django и Apache.

Ключевые слова: Django, Apache, mod_wsgi, логирование, ошибки веб-сервера, виртуальное окружение Python, ОС Astra Linux

Для цитирования. Хирова Е.Д., Каменский Д.А. Совместная работа Django и Apache: анализ и пути решения ошибок по лог-файлам. *Молодой исследователь Дона*. 2026;11(4):33–35.

Deploying Django with Apache: Analysis of the Errors and Troubleshooting Using Log Files**Ekaterina D. Khirova, Danil A. Kamensky**

Don State Technical University, Rostov-on-Don, Russian Federation

Abstract

Deploying Django with Apache causes various errors that get recorded in Apache log files. The aim of the article is to analyse the occurring errors and propose efficient troubleshooting techniques. Consistent analysis of the following errors has been performed: access denied (403), incompatibility with Python subinterpreters, module import, and invalid file path errors. Causes inducing each error have been analysed and practical troubleshooting techniques have been proposed. A systemic guide to error debugging that reduces the time for deploying Django with Apache has been presented.

Keywords: Django, Apache, mod_wsgi, logging, web server errors, Python virtual environment, the operating system Astra Linux.

For Citation. Khirova ED, Kamensky DA. Deploying Django with Apache: Analysis of the Errors and Troubleshooting Using Log Files. *Young Researcher of Don*. 2026;11(4):33–35.

Введение. При развёртывании веб-приложений на основе Django часто используется связка с веб-сервером Apache через модуль mod_wsgi, который выступает в роли интерфейса между сервером и Python-кодом. Однако в процессе настройки и эксплуатации такой системы возникают разнообразные ошибки, фиксируемые в лог-файлах Apache. Они содержат информацию о запросах к серверу, ошибках и других событиях, происходящих на сервере. Анализ этих ошибок является ключевым для обеспечения стабильной работы приложения.

Лог-файлы Apache делятся на два основных типа: журнал посещений (access log), содержащий информацию о запросах клиентов, и журнал ошибок (error log), где регистрируются сбои и предупреждения сервера [1].

Актуальность исследования обусловлена возможностью повышения эффективности администрирования веб-серверов и оптимизации обработки данных при использовании связки Apache и Django, позволяющей оперативно разрабатывать веб-приложения с использованием готовых инструментов.

Цель исследования — провести анализ возникающих ошибок в лог-файлах при совместной работе Django и Apache в среде с виртуальным окружением Python (на примере ОС Astra Linux), и предложить эффективные методы их устранения.

Основная часть. Имеем виртуальное окружение с именем `venv` и версией Python 3.11. Веб-сервер Apache работает с модулем `mod_wsgi`, настроенным на использование этого окружения. Проект Django имеет корневую папку проекта «project», внутри которой и есть сам проект, для удобства переименуем его в «add». Ошибки, возникающие в логах, можно разделить на три категории: связанные с конфигурацией Apache, с файлом `wsgi.py` и с самим проектом Django. Рассмотрим некоторые из них.

Ошибки конфигурации Apache. Ошибка *403 Forbidden*. Данная ошибка означает, что сервер получил запрос, но отказывается его выполнять из-за недостатка прав доступа. При попытке запустить Django-проект через Apache эта ошибка чаще всего возникает из-за неверных прав доступа к файлам проекта в конфигурации виртуального хоста. Для ее решения в конфигурационном файле Apache нужно предоставить разрешение `Require all granted` каталогу проекта и файлу `wsgi.py`.

Ошибка «*client denied by server configuration*». Это сообщение в логах Apache означает, что сервер отклонил запрос клиента из-за правил доступа, заданных в конфигурации. Для исправления необходимо убедиться, что прописан абсолютный путь, потому что Apache ожидает именно его в контейнере `<Directory>`.

Ошибка «*Truncated or oversized response headers received from daemon process 'add'*». Данная запись свидетельствует о том, что процесс `mod_wsgi` неожиданно завершился до отправки полных заголовков ответа. Причиной могут быть использование C-расширений в под-интерпретаторе, или ошибки в коде приложения. Для решения следует убедиться, что виртуальное окружение корректно указано в конфигурации Apache (директива `python-home`).

Также в директиве `python-path` указывается корень проекта `django` и путь до каталога с библиотеками виртуального окружения, чтобы файл `manage.py` работал через `venv`: `python-path=/home/user/project:/home/user/venv/lib/python3.11/site-packages`.

После исправления ошибки выше появится сообщение «*UserWarning: NumPy was imported from a Python sub-interpreter but NumPy does not properly support sub-interpreters.*» Оно указывает на использование C-расширений в под-интерпретаторе. Решение — нужно заставить Django-приложение работать в основном интерпретаторе Python, а не в под-интерпретаторах. Для этого в конфигурации Apache добавляется директива `WSGIApplicationGroup %{GLOBAL}`, которая заставит приложение работать в главном интерпретаторе [2].

Важно после любого изменения перезагружать Apache с помощью команды: `sudo systemctl restart apache2`. Данная команда помогает применить изменения, внесённые в конфигурацию веб-сервера [3].

Ошибки, связанные с файлом `wsgi.py`. Ошибка *ModuleNotFoundError: No module named 'django'* возникает, когда интерпретатор Python, используемый `mod_wsgi` не видит установленный Django. Это происходит из-за того, что Apache обращается к системному окружению Python, а не к виртуальному. Несмотря на то, что путь к виртуальному окружению указан в конфигурации Apache с помощью директивы `python-home`, из-за использования ОС Astra Linux требуется дополнительно вручную подменять пути в начале файла `wsgi.py`. На рис. 1 выполняется проверка активного интерпретатора и принудительная установка требуемого окружения, также реализуется механизм защиты от конфликтов, если в системе присутствуют альтернативные установки Python (например, Anaconda). Фрагмент кода в начале файла `wsgi.py` представлен на рис. 1.

```
python_home='/home/katja08/new'
if python_home not in sys.executable:
    sys.executable='/home/katja08/new/bin/python3.11'
    os.environ['PYTHONHOME']=python_home
if 'PYTHONPATH' in os.environ and 'anaconda3' in os.environ['PYTHONPATH']:
    os.environ['PYTHONPATH']=''
```

Рис. 1. Установка требуемого виртуального окружения

Ошибка *ImportError: cannot import name 'Sequence' from 'collections'* возникает, если какая-либо библиотека или код проекта пытаются импортировать `Sequence` из `collections`, что устарело и не работает в Python 3.11+ (начиная с 3.10, многие ABC были перемещены в `collections.abc`). Решение — обновить проблемную библиотеку до версии, совместимой с Python 3.11. Если обновление невозможно (например, пакет больше не поддерживается), можно добавить в файл `wsgi.py` (до импорта Django-проекта), код восстановит `Sequence` в модуле `collections`. Фрагмент кода файла `wsgi.py` представлен на рис. 2.

```
import collections
if sys.version_info >=(3, 10):
    try:
        from collections.abc import Sequence
        collections.Sequence = Sequence
    except ImportError:
        pass
```

Рис. 2. Импорт `Sequence` из `collections.abc`

Данный код проверяет, что версия Python 3.10 или выше. После импортирует `Sequence` из `collections.abc`. Если успешно, присваивает этот класс атрибуту `collections.Sequence`.

Ошибки проекта Django. Ошибка *TypeError: expected str, bytes or os.PathLike object, not PosixPath* возникает, когда какая-то функция ожидает на входе строку (str), байты (bytes) или объект, поддерживающий интерфейс PathLike, а получает объект типа PosixPath. Ошибка говорит о том, что передается объект Path туда, где он не может быть автоматически преобразован в строку. Самый простой способ — преобразовать объект Path в строку там, где это необходимо с помощью str(path). Обычно это требуется применять в файле settings.py в местах, где пути формируются на основе переменной BASE_DIR. К числу таких мест относятся: DATABASES, STATICFILES_DIRS, MEDIA_ROOT и LOGGING.

Заключение. В результате проведенного анализа выявлены и систематизированы наиболее распространенные ошибки, возникающие при совместной работе Django и Apache с модулем mod_wsgi на ОС Astra Linux. Установлено, что их устранение требует, как корректной настройки конфигурационных файлов сервера, так и доработки кода самого приложения. Для каждой ошибки предложены способы устранения с учетом ОС Astra Linux, из-за которой приходится решать одну ошибку в нескольких местах.

Список литературы

1. *Apache Log Files: A Complete Guide to Access and Error Logs*. URL: <https://signoz.io/guides/apache-log/> (дата обращения: 26.02.2026).
2. *mod_wsgi/Configuration/WSGIApplicationGroup*. URL: <https://modwsgi.readthedocs.io/en/develop/configuration-directives/WSGIApplicationGroup.html> (дата обращения: 26.02.2026).
3. Установка и настройка Apache на Ubuntu 22.04 LTS и Debian 12. URL: <https://adminvps.ru/blog/ustanovka-i-nastroyka-apache-na-ubuntu-22-04-lts-i-debian-12> (дата обращения: 26.02.2026).

Об авторах:

Екатерина Дмитриевна Хирова, магистрант кафедры «Радиоэлектроника» Донского государственного технического университета (344003, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1), mdkatja4773@gmail.com

Данил Александрович Каменский, магистрант кафедры «Радиоэлектроника» Донского государственного технического университета (344003, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1), danil.kamensky@yandex.ru

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the Authors:

Ekaterina D. Khirova, Master's Degree Student of the Radio and Electronics Engineering Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, Russian Federation), rndkatja4773@gmail.com

Danil A. Kamensky, Master's Degree Student of the Radio and Electronics Engineering Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, Russian Federation), danil.kamensky@yandex.ru

Conflict of Interest Statement: the authors declare no conflict of interest.

All authors have read and approved the final manuscript.