Python Practice

Добавление логирования в базу данных

Автор курса

Кременtарь Ксения

Ведущий Python разработчик

Системный архитектор

в компании K-Solutions

После урока обязательно

Повторите этот урок в видео формате на ITVDN.com

Проверьте как Вы усвоили данный материал на TestProvider.com

Добавление логирования в базу данных

На этом уроке

1. Познакомимся с принципами построения логирования в базу данных.
2. Научимся создавать таблицы логов под нужды проекта.
3. Добавим таблицы логов для проекта golden-eye.
4. Познакомимся с возможностями библиотеки traceback.
5. Добавим нужные индексы для таблиц логов.
6. Внесем изменения в код проекта, для сохранения данных в таблицах логов при обращении к удаленным API.

Логирование в базу данных — зачем?

Цели логирования в БД схожи с целями для логирования в файл:

- Для просмотра логов службой поддержки (или теми, кто будет поддерживать проект), при возникновении вопросов по работе системы.
- Для анализа и сбора статистики.
- Для контроля надежности и стабильности системы.

Но для просмотра логов в файлах, необходимо иметь доступ на сервер, на котором эти файлы находятся. Логи же, сохраненные в базу данных, могут быть доступны для просмотра пользователями, как и любая другая информация из БД. Главное - организовать отображение логов на сайте. В рамках нашего курса мы добавим отображение логов в проекте golden-eye на последующих уроках.

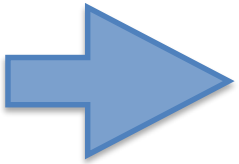
Какую информацию логировать?

Интересует логирование запросов-ответов по работе с удаленными API. Будем логировать следующую информацию:

- время начала и конца обработки запроса.
- url, на который отправлен запрос.
- текст запроса.
- HTTP метод запроса.
- полученный текст ответа.
- заголовки запроса.
- текст ошибки, если она произошла.

Структура базы данных

api_logs	
request_url	varchar, not null
request_data	text, null
request_method	varchar, not null
request_headers	text, null
response_text	text, null
created	datetime, not null
finished	datetime, not null
error	text, null



```python
class ApiLog(_Model):
    class Meta:
        db_table = "api_logs"

    request_url = CharField()
    request_data = TextField(null=True)
    request_method = CharField(max_length=100)
    request_headers = TextField(null=True)
    response_text = TextField(null=True)
    created = DateTimeField()
    finished = DateTimeField()
    error = TextField(null=True)
```

Логирование ошибок

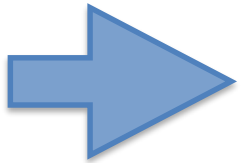
Помимо логирования запросов-ответов и любой другой информации в таблицы логов, настоятельно рекомендуется добавлять отдельную таблицу для логирования непредвиденных ошибок, возникающих в работе проекта.

В нее можно и нужно логировать не только текст возникающей ошибки, но и полный стектрейс ошибки.

Назвать ее можно, например, error_logs. Очень важно логировать только непредвиденные ошибки, не захламлять таблицу лишними записями. При нормальной работе проекта, в ней не должно быть много записей, а если почему-то их стало много — это сигнал о том, что что-то поломалось после очередного обновления и т.п.

Таблица error_logs

error_logs	
request_url	varchar, not null
request_data	text, null
request_method	varchar, not null
created	datetime, not null
error	text, not null
traceback	text, null



```python
class ErrorLog(_Model):
    class Meta:
        db_table = "error_logs"

    request_data = TextField(null=True)
    request_url = TextField()
    request_method = CharField(max_length=100)
    error = TextField()
    traceback = TextField(null=True)
    created = DateTimeField()
```

Встроенный модуль traceback

При сохранении стека ошибки в логах ErrorLog можно потом инспектировать ошибку и видеть не только ее текст, но и файл и строку python кода, которая вызвала ошибку. Это очень удобно при отладке приложений не только на этапе разработки, но (что еще более важно!) и на этапе поддержки проекта после размещения на production среде.

Встроенный модуль traceback предоставляет возможность для извлечения, форматирования и печати трассировок стека Python программ.

Основные методы этого модуля описаны в официальной документации, https://docs.python.org/3.6/library/traceback.html Нас интересует метод traceback.format_exc(), который возвращает строку с информацией об исключении и трассировочную информацию.

Индексы в таблицах логов

Индексы в таблицы логов логично добавлять на поля времени — начала или окончания работы с API. Остальные поля будут не очень хорошо индексироваться.

Опишем индексы в моделях. Создадим таблицы в базе данных — допишем функцию init_db для инициализации базы данных.

Двигаемся дальше!

Итак, все готово для внесения изменений в код. Что именно нам нужно сделать?

- Добавить использование новых моделей логов, а именно: сохранения (создание записей в базе данных) при отправке запросов и получения ответов по API. Присутствие API в каждом скрипте будет не очень удобным — лучше вынести его в базовый скрипт. Давайте попробуем!
- Дописать тесты — проверить, что создаются записи в таблицах логов. Проверять содержимое записей, конкретные поля.
- Проверить создание записей в таблице error_logs в случае ошибки. Для этого нужно эмулировать ошибку, например - ошибку таймаута. Для этого вынесем значение таймаута в модуль config.py, чтоб можно было автоматически выставлять его в тестах, да и в целом конфигурировать.

Что получилось к настоящему моменту

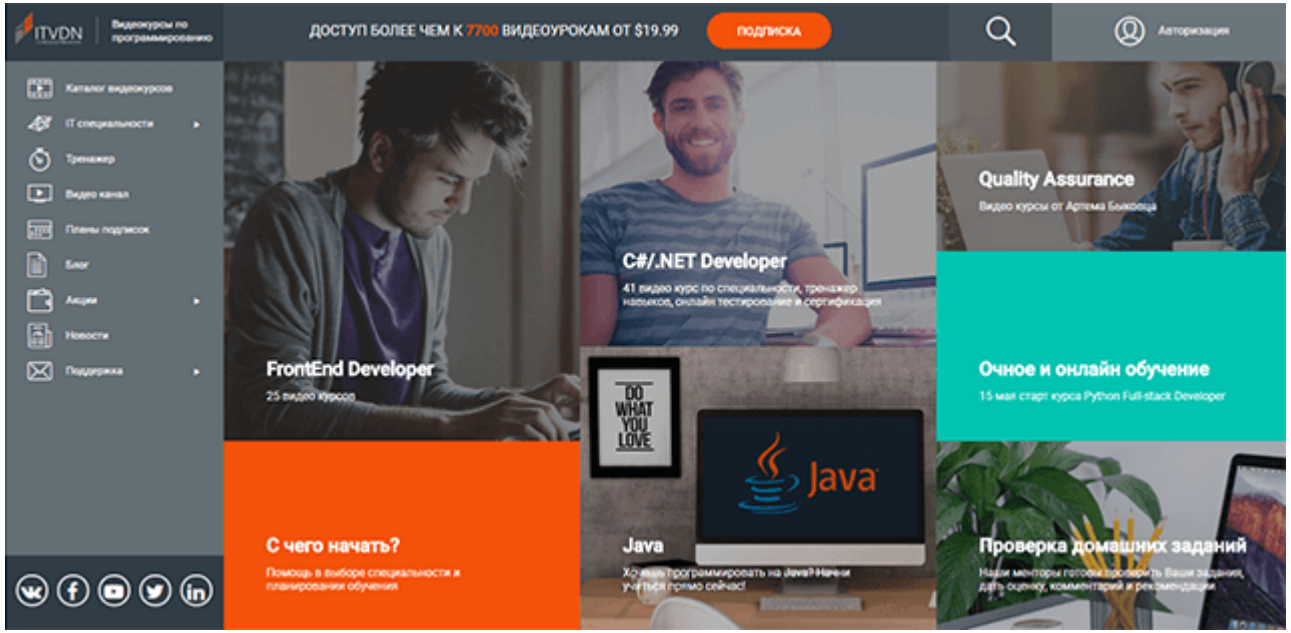
- Функционал базового скрипта api расширен, добавлена отправка запроса и сохранение логов в базу данных.
- Добавлено сохранение ошибки в таблицу error_logs.
- Добавление нового скрипта нового API — теперь просто и быстро. Мы сможем в этом убедиться!

Что дальше?

- Добавление новых курсов валют BTC => UAH, BTC => RUB, BTC => USD
- Реализация нового API.
- Добавление динамического импорта нужного модуля API.

Смотрите наши уроки в видео формате

ITVDN.com



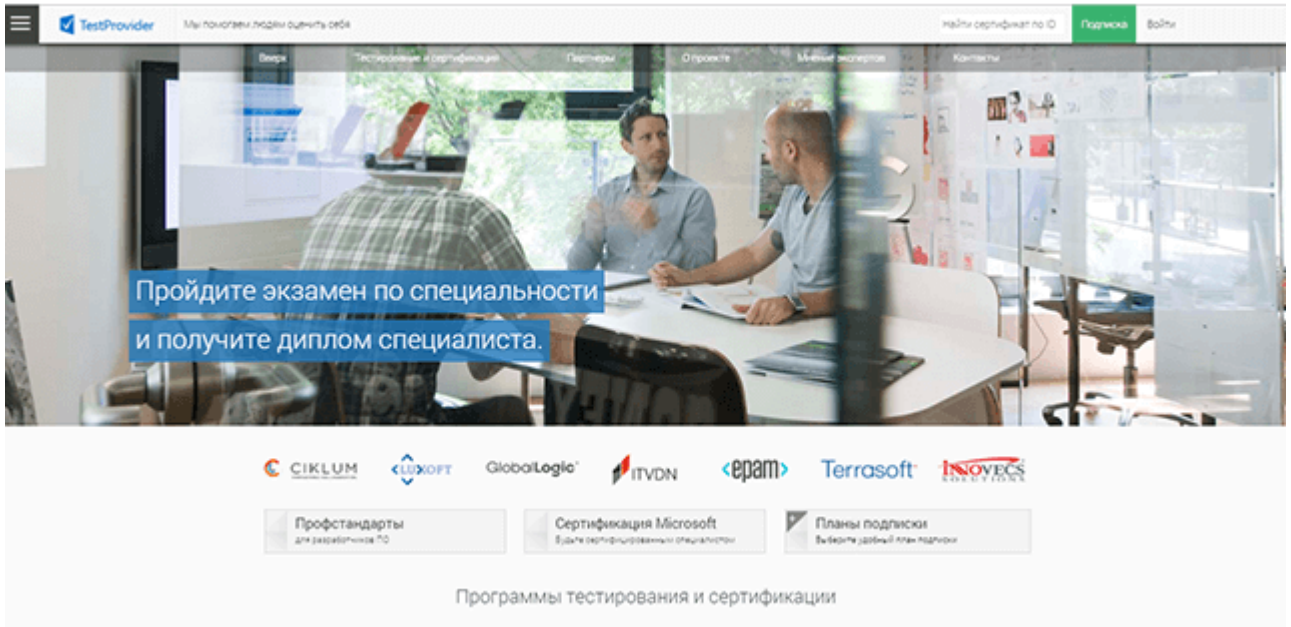
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.

ITVDN
IT VIDEO DEVELOPERS NETWORK

Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

TestProvider

Q&A

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association