

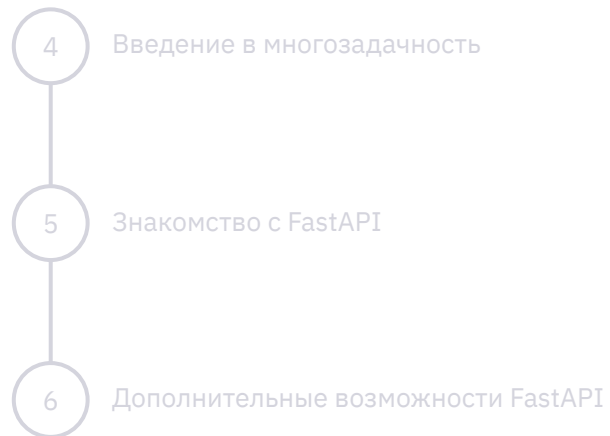
Дополнительные ВОЗМОЖНОСТИ Flask

Урок 3





План курса







Содержание урока



Что будет на уроке сегодня

-  Узнаем про работу с базами данных посредством Flask-SQLAlchemy
-  Разберёмся с созданием форм средствами Flask-WTForm



Flask-SQLAlchemy



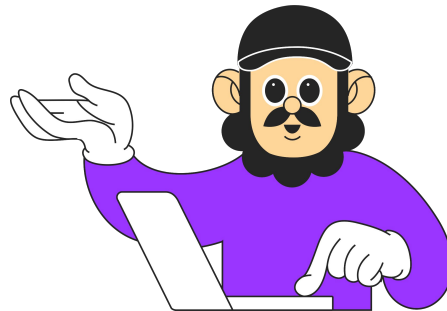
Установка и настройка

Установка:

```
pip install Flask-SQLAlchemy # Windows  
pip3 install Flask-SQLAlchemy # Unix
```

Подключение к БД:

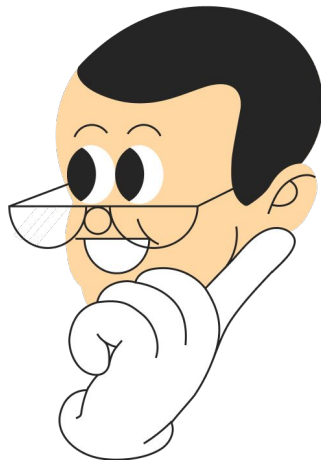
```
from flask_sqlalchemy import SQLAlchemy
```



Создание моделей

При работе с Flask-SQLAlchemy необходимо определить модели данных, которые будут использоваться в приложении.

- Определение классов моделей
- Описание полей моделей (см. следующий слайд)
- Создание связей между моделями
- Создание таблиц в базе данных

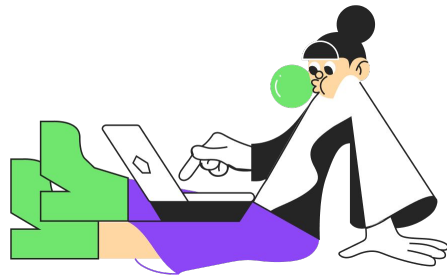




Описание полей моделей

Для описания полей модели используются классы-типы данных из библиотеки SQLAlchemy

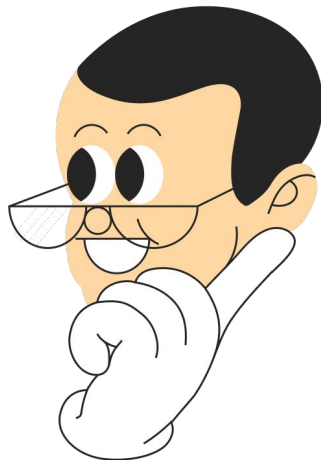
- Integer - целое число
- String - строка
- Text - текстовое поле
- Boolean - булево значение
- DateTime - дата и время
- Float - число с плавающей точкой
- Decimal - десятичное число
- Enum - перечисление значений
- ForeignKey - внешний ключ к другой таблице



Создание моделей

При работе с Flask-SQLAlchemy необходимо определить модели данных, которые будут использоваться в приложении.

- Определение классов моделей
- Описание полей моделей (см. следующий слайд)
- Создание связей между моделями
- Создание таблиц в базе данных

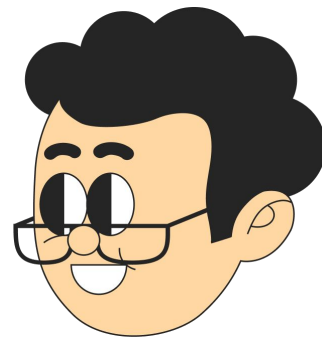




Работа с данными

После определения моделей в Flask-SQLAlchemy можно начать работу с данными в базе данных.

- Создание записей
- Изменение записей
- Удаление записей
- Получение данных из базы данных
 - Фильтрация данных





Flask-WTForm

Установка и настройка

Установка:

```
pip install Flask-WTF # Windows  
pip3 install Flask-WTF # Unix
```

Доступ к формам:

```
from flask_wtf import FlaskForm
```

Настройка защиты от CSRF-атак:

```
app = Flask(__name__)  
app.config['SECRET_KEY'] = 'mysecretkey'  
csrf = CSRFProtect(app)
```



Создание форм

WTForms - это библиотека Python, которая позволяет создавать HTML-формы, а также проводить их валидацию.

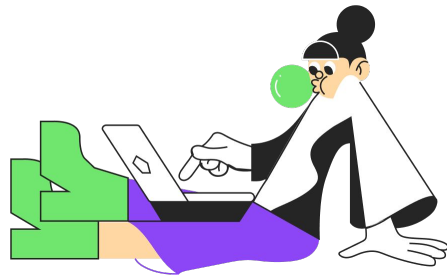
- Определение классов форм
- Описание полей форм (см. следующий слайд)
- Валидация данных формы



Описание полей форм

WTForms предоставляет множество типов полей для формы

- StringField - строковое поле для ввода текста;
- IntegerField - числовое поле для ввода целочисленных значений;
- FloatField - числовое поле для ввода дробных значений;
- BooleanField - чекбокс;
- SelectField - выпадающий список;
- DateField - поле для ввода даты;
- FileField - поле для загрузки файла.



Создание форм

WTForms - это библиотека Python, которая позволяет создавать HTML-формы, а также проводить их валидацию.

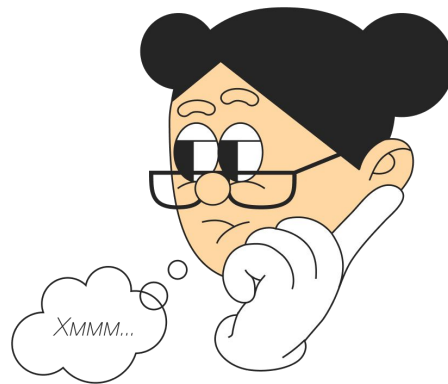
- Определение классов форм
- Описание полей форм (см. следующий слайд)
- Валидация данных формы



Использование форм в приложении

После создания форм рассмотрим, как их использовать в приложении Flask.

- Отображение форм на страницах приложения
- Обработка данных из формы







Итоги занятия



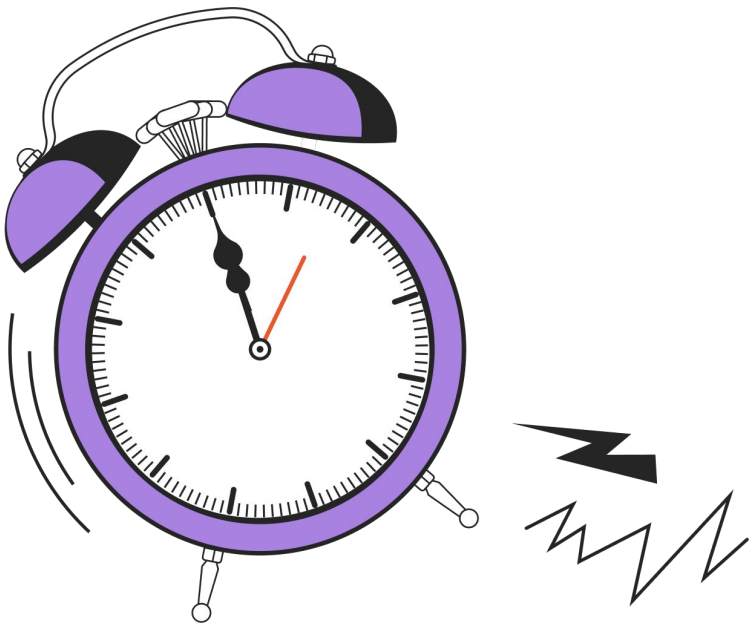
На этой лекции мы

-  Узнали про работу с базами данных посредством Flask-SQLAlchemy
-  Разобрались с созданием форм средствами Flask-WTForm



Задание

1. Для закрепления материалов лекции попробуйте самостоятельно набрать и запустить демонстрируемые примеры.





Спасибо за внимание