

# Tytuł szkolenia

Flask. Mikroserwisy w Pythonie

**Radosław Janiak**

Trener





## **Radosław Janiak**

Trener programowania, programista.

# Wprowadzenie do Flaska

---

- Flask to lekki framework webowy dla Pythona, zapewniający podstawowe narzędzia do budowy aplikacji internetowych."
- "Znany ze swojej prostoty i elastyczności, pozwala na szybkie tworzenie aplikacji."
- "W odróżnieniu od Django, Flask nie narzuca sztywnych konwencji i jest bardziej 'mikro', co oznacza mniejszą ilość wbudowanych funkcji, ale większą kontrolę dla programisty."

# Instalacja i Konfiguracja

---

- "Instalacja Flaska jest prosta i wymaga tylko jednego polecenia: `pip install Flask`."
- "Podstawowa struktura projektu Flask obejmuje katalog główny, podkatalogi `'templates'` dla szablonów HTML i `'static'` dla plików CSS i JavaScript."
- "Konfiguracja Flaska jest również prosta i zazwyczaj wymaga tylko kilku linii kodu do rozpoczęcia."

# Tworzenie Pierwszej Aplikacji

- Stworzenie podstawowej aplikacji Flask wymaga tylko kilku linii kodu. Najpierw importujemy Flask, a następnie tworzymy instancję aplikacji."
- "Definiujemy trasę i funkcję, która będzie obsługiwać żądania do tej trasy."
- "Aby uruchomić aplikację, używamy flask run. Domyślnie serwer będzie dostępny pod adresem localhost:5000."

- ```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def index():

    return 'Hello World!'

if __name__ == '__main__':
    # app.run(debug=True, port=8080)
    app.run(debug=True)
```

# Routing i Widoki

- "Routing w Flasku pozwala definiować, jak aplikacja odpowiada na żądania do określonych adresów URL."
- "Każda trasa jest powiązana z funkcją, zwaną widokiem, która jest wywoływana, gdy trasa jest dopasowana."
- "Flask obsługuje zmienne w trasach, co pozwala na łatwe przekazywanie danych do widoków."
- `@app.route('/')`
- `@app.route('/cantor/<currency>/<int:amount>')`
- `@app.route('/exchange', methods=['GET', 'POST'])`

# Formularze i Przetwarzanie Danych

- Flask pozwala na łatwe przetwarzanie danych wysyłanych przez formularze HTML."
- "Do walidacji i reprezentacji formularzy często używa się rozszerzenia Flask-WTForms."
- "Przykład pokazuje, jak odbierać dane z formularza i odpowiednio na nie reagować w funkcji widoku."
- # Definicja klasy formularza, używająca Flask-WTF
  - `class MyForm(FlaskForm):`
    - `name = StringField('Imię', validators=[DataRequired()])`
    - `submit = SubmitField('Wyślij')`
  - `def index(): form = MyForm() if form.validate_on_submit(): # Tutaj możemy zrobić coś z przesłanymi danymi`
    - `flash(f'Formularz został wysłany z imieniem: {form.name.data}')`
    - `return redirect(url_for('index'))`
  - `return render_template('index.html', form=form)`

# Bazy Danych

---

- "Flask może być łatwo zintegrowany z różnymi systemami baz danych."
- "Przykładem jest użycie SQLite - lekkiej bazy danych, która nie wymaga osobnego serwera."
- "Pokazujemy, jak zdefiniować model danych i wykonywać podstawowe operacje CRUD (Create, Read, Update, Delete)."

Definiujemy punkty końcowe API do obsługi operacji CRUD:

1. **CREATE:** Tworzenie nowego przedmiotu za pomocą żądania POST.
2. **READ:** Pobieranie szczegółów istniejącego przedmiotu za pomocą żądania GET.
3. **UPDATE:** Aktualizacja istniejącego przedmiotu za pomocą żądania PUT.
4. **DELETE:** Usunięcie istniejącego przedmiotu za pomocą żądania DELETE.



# Bezpieczeństwo

- "Bezpieczeństwo jest kluczowym aspektem każdej aplikacji internetowej. Flask oferuje wiele wbudowanych funkcji do ochrony przed typowymi zagrożeniami."
- Flask i jego rozszerzenie Jinja2 domyślnie zapobiegają atakom XSS poprzez automatyczne escapowanie (tj. konwersję na bezpieczne znaki) wszystkich zmiennych wyświetlanych w szablonach HTML.
- "Omówienie zapobiegania atakom XSS (Cross-Site Scripting) i CSRF (Cross-Site Request Forgery)."
- Flask generuje ukryte pole zawierające token CSRF, który jest sprawdzany przy przetwarzaniu formularza.
- "Ważne jest również bezpieczne przechowywanie haseł i odpowiednia konfiguracja aplikacji."
- `app.config['SECRET_KEY'] = 'bardzo_tajny_klucz'`

# Dodatkowe Zasoby i Następne Kroki

---

- "Aby dalej rozwijać swoje umiejętności w Flasku, zalecane jest eksplorowanie zaawansowanych tematów i tworzenie własnych projektów."
- "Zapoznanie się z dokumentacją Flaska oraz dostępnymi rozszerzeniami."
- "Realizacja prostego projektu, np. bloga czy aplikacji do zarządzania zadaniami, może być świetnym sposobem na praktyczną naukę."

# Ankieta

---



Adres strony:

<https://www.erp.comarch.pl/Szkolenia/Ankiety/surveys>



**COMARCH**  
Szkolenia

# Dziękujemy za udział w szkoleniu

Flask. Mikroserwisy w Pythonie

**Radosław Janiak**

Trener



# Centrum Szkoleniowe Comarch

---

ul. Prof. M.Życzkowskiego 33

31-864 Kraków

Tel. +48 (12) 687 78 11

E-Mail: [szkolenia@comarch.pl](mailto:szkolenia@comarch.pl)

[www.szkolenia.comarch.pl](http://www.szkolenia.comarch.pl)



**COMARCH**  
Szkolenia

[www.szkolenia.comarch.pl](http://www.szkolenia.comarch.pl)