LU06.A01 - Einfaches Flask-Routing

In dieser Aufgabe wirst du eine einfache Flask-Anwendung erstellen, die verschiedene Endpunkte (Endpoints) bedient. Deine Aufgabe ist es, die spezifizierten Routen zu implementieren und die entsprechenden Antworten als Text zurückzugeben.

Endpoints

1. **`/` (GET)**: Gibt den Text `Willkommen bei meiner Flask-App!` zurück.
2. **`/info` (GET)**: Gibt den Text `Dies ist die Info-Seite.` zurück.
3. **`/user/<username>` (GET)**: Gibt den Text `Hallo, [username]!` zurück, wobei [username] durch den in der URL angegebenen Benutzernamen ersetzt wird.
4. **`/post` (POST)**: Akzeptiert Daten und gibt den Text `Daten erfolgreich erhalten!` zurück.
5. **`/feedback` (GET und POST)**:
   - Bei einem GET-Request gibt es den Text `Bitte geben Sie Ihr Feedback ab.` zurück.
   - Bei einem POST-Request gibt es den Text `Danke für Ihr Feedback!` zurück.
6. **`/item/<int:item_id>` (GET)**: Gibt den Text `Artikel-ID: [item_id]` zurück, wobei [item_id] durch die in der URL angegebene Artikel-ID ersetzt wird.

Code-Vorlage

main.py

```python
from flask import Flask, request


app = Flask(__name__)


# TODO: Implementiere eine Route für '/', die bei einem GET-Request den Text 'Willkommen bei meiner Flask-App!' zurückgibt.
def home():
    """Gibt den Text 'Willkommen bei meiner Flask-App!' zurück."""
    pass


# TODO: Implementiere eine Route für '/info', die bei einem GET-Request den Text 'Dies ist die Info-Seite.' zurückgibt.
def info():
    """Gibt den Text 'Dies ist die Info-Seite.' zurück."""
    pass
```

```python
# TODO: Implementiere eine Route für '/user/<username>', die bei einem GET-Request den Text 'Hallo, [username]!' zurückgibt, wobei [username] durch den in der URL angegebenen Benutzernamen ersetzt wird.
def user(username):
    """Gibt den Text 'Hallo, [username]!' zurück, wobei [username] durch den in der URL angegebenen Benutzernamen ersetzt wird."""
    pass


# TODO: Implementiere eine Route für '/post', die bei einem POST-Request den Text 'Daten erfolgreich erhalten!' zurückgibt.
def post_data():
    """Akzeptiert Daten und gibt den Text 'Daten erfolgreich erhalten!' zurück."""
    pass


# TODO: Implementiere eine Route für '/feedback', die bei einem GET-Request den Text 'Bitte geben Sie Ihr Feedback ab.' zurückgibt und bei einem POST-Request den Text 'Danke für Ihr Feedback!' zurückgibt.
def feedback():
    """Bei einem GET-Request gibt es den Text 'Bitte geben Sie Ihr Feedback ab.' zurück. Bei einem POST-Request gibt es den Text 'Danke für Ihr Feedback!' zurück."""
    pass


# TODO: Implementiere eine Route für '/item/<int:item_id>', die bei einem GET-Request den Text 'Artikel-ID: [item_id]' zurückgibt, wobei [item_id] durch die in der URL angegebene Artikel-ID ersetzt wird.
def item(item_id):
    """Gibt den Text 'Artikel-ID: [item_id]' zurück, wobei [item_id] durch die in der URL angegebene Artikel-ID ersetzt wird."""
    pass


if __name__ == '__main__':
    app.run(debug=True)
```

Vorgehen

1. Akzeptiere das GitHub Classroom Assignment
2. Klone dein persönliches Repository in die Entwicklungsumgebung
3. Löse die Aufgabenstellung

API-Testing

Verwenden Sie diesen Postman Link und eine Postman Konfiguration zu laden.

Abgabe

Die Abgabe der Lösung erfolgt als Push in das persönliche GitHub-Repository.

⇒ *GitHub Repo für externe Besucher*

GitHub Repository https://github.com/templates-python/m323-lu06-a01-routing

Lernende am BZZ müssen den Link zum GitHub Classroom Assignment verwenden

© Kevin Maurizi

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/m323/learningunits/lu06/aufgaben/routing

Last update: **2024/03/28 14:07**