LU06j - Blueprints in Flask

Blueprints in Flask sind eine Möglichkeit, die Organisation einer Anwendung zu verbessern, insbesondere wenn diese wächst und mehrere Komponenten enthält. Sie dienen zur Modularisierung einer Flask-Anwendung und können als Vorlage für Applikationsstrukturen betrachtet werden.

Was sind Blueprints?

Ein Blueprint ist im Wesentlichen ein Modul, das Routen und Fehlerbehandlungen kapselt. Es kann mehrere Routen und Vorlagen, statische Dateien und vieles mehr enthalten. Blueprints bieten eine saubere Trennung der Funktionen und erleichtern die Wiederverwendung von Code.

Warum Blueprints verwenden?

- **Modularität**: Blueprints erlauben es, Funktionen in verschiedene Module zu trennen.
- **Wiederverwendbarkeit**: Blueprints können in mehreren Projekten wiederverwendet werden.
- **Einfachheit**: Durch die Verwendung von Blueprints wird der Code übersichtlicher und einfacher zu verwalten.

Beispiel für einen Blueprint

Ein einfacher Blueprint könnte folgendermaßen aussehen:

```
from flask import Blueprint, request, jsonify

shopping_item_bp = Blueprint('shopping_item_bp', __name__)

@shopping_item_bp.route('/shoppingItems', methods=['GET'])
def get_items():
    items = dao.get_all_items()
    return jsonify(items), 200

@shopping_item_bp.route('/shoppingItems/<int:item_id>', methods=['GET'])
def get_item(item_id):
    item = dao.get_item(item_id)
    if item:
        return jsonify(item), 200
    return jsonify({'error': 'Item not found'}), 404
```

Registrierung von Blueprints

Nach der Erstellung des Blueprints muss dieser in der Hauptanwendung registriert werden:

```
from flask import Flask

app = Flask(__name__)

# Importieren und Registrieren des Blueprints
from my_blueprint_module import shopping_item_bp

app.register_blueprint(shopping_item_bp)
```

Mit dieser Methode können Sie mehrere Blueprints in einer einzigen Anwendung registrieren, um verschiedene Aspekte zu verwalten.

Blueprints und URL-Präfixe

Beim Registrieren eines Blueprints können Sie optional einen URL-Präfix angeben:

```
app.register_blueprint(my_blueprint, url_prefix='/shopinglist')
```

Dadurch wird der Endpunkt des Blueprints zu `/shopinglist/shoppingItems` anstatt nur `/shoppingItems`.

Blueprints und Fehlerbehandlung

Blueprints können auch eigene Fehlerbehandlungen enthalten. Dies ist nützlich, wenn Sie spezielle Fehlerseiten für verschiedene Teile Ihrer Anwendung haben möchten.

```
@my_blueprint.errorhandler(404)
def not_found_error(error):
    return 'Diese Seite wurde nicht gefunden.', 404
```

M323-LU06

© Kevin Maurizi

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

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

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