

Technische Dokumentation und Komponentenbeschreibung:

Implementierung einer PayPal-Zahlungsauslösung mit Python und Flask

Auswahl des Frameworks: Flask vs. Django Framework

Bei der Auswahl des geeigneten Frameworks für das Projekt bestand die Wahl zwischen Flask und Django. Die Wahl fiel schließlich auf das Framework Flask. Flask ist ein auf Python basierendes Microframework, das sich insbesondere durch seine Flexibilität und Anpassbarkeit auszeichnet. Es bietet eine Reihe nützlicher Tools und Funktionen, die die Entwicklung von Webanwendungen in Python erleichtern Python¹ erleichtern.

Django hingegen eignet sich für die Erstellung komplexer Websites und Anwendungen und ist besonders für Entwickler nützlich, die bereits Erfahrung mit Python haben². Das Framework bietet zahlreiche vorgefertigte Funktionen und Bibliotheken, die die Webentwicklung effizienter machen und gleichzeitig die Sicherheit und Skalierbarkeit der Anwendungen gewährleisten.

Einer der Gründe, warum wir uns letztendlich für Flask entschieden haben, ist die gute Anpassbarkeit des Frameworks. Im Gegensatz zu Django können Entwickler bei Flask ihre eigenen Komponenten und Tools auswählen und implementieren. Dies hat den Vorteil, dass man bei technischen Entscheidungen nicht durch vorgegebene Konfigurationsmöglichkeiten eingeschränkt ist³.

Darüber hinaus eignete sich Flask besonders gut für den Einsatz einer statischen Website im Projekt. Ein weiterer Vorteil ist die einfache und klare Struktur, die die Entwicklung und Wartung von Flask-Anwendungen erleichtert. Darüber hinaus ist die Lernkurve von Flask oft einfacher als die von Django, was es für Anfänger zugänglicher macht².

Ein weiterer Vorteil von Flask ist seine Erweiterbarkeit. Entwickler können verschiedene Tools und Bibliotheken verwenden, um ihre Websites zu erweitern und anzupassen. Jedes Flask-Projekt hat seinen eigenen, einzigartigen „internen Technologie-Stack“ aus frei verfügbaren Frameworks und Bibliotheken⁴.

Es ist wichtig zu beachten, dass die Wahl zwischen Flask und Django von den spezifischen Anforderungen des Projekts abhängt. Flask bietet Flexibilität, Anpassbarkeit und eine einfache Struktur, weshalb es für unser PayPal-Zahlungsauslösungsprojekt sehr gut geeignet war⁵.

Funktionalitäten des PayPal Python SDK:

Das PayPal Python SDK bietet eine Vielzahl von Funktionen zur Integration von PayPal-Zahlungen in Python-Anwendungen. Das SDK bietet unter anderem folgende Funktionen:

1. Erstellung von Zahlungen:

Das SDK ermöglicht es Entwicklern, Zahlungen zu erstellen und zu verarbeiten. Es bietet Methoden und Funktionen, um Zahlungsinformationen zu erfassen, Transaktionen durchzuführen und den Zahlungsstatus abzufragen⁶.

¹ <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3-de>

² <https://bmu-verlag.de/django-framework-mit-python/>

³ <https://cynoteck.com/de/blog-post/flask-vs-django/>

⁴ <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3-de>

⁵ <https://web-forward.de/2020/09/flask-vs-django-das-python-duell/>

⁶ <https://paypal.github.io/sdk>

2. Verwaltung von Zahlungen:

Das SDK ermöglicht es Entwicklern, Zahlungen zu verwalten, einschließlich Rückerstattungen, Stornierungen und Zahlungsdaten zu erfassen⁵.

3. Integration von PayPal-Buttons:

Das SDK bietet Funktionen zur Integration von PayPal-Buttons in Webanwendungen. Entwickler können benutzerdefinierte Zahlungsformulare erstellen und PayPal-Zahlungen nahtlos in ihre Anwendungen integrieren⁷.

4. Zugriff auf PayPal-APIs:

Für den Zugriff auf die RESTful APIs von PayPal stellt das SDK eine Schnittstelle zur Verfügung. Entwickler können auf Funktionen wie Zahlungsabwicklung, Transaktionsdetails und Kontoinformationen zugreifen⁶.

Es ist wichtig zu beachten, dass das PayPal Python SDK in einigen Quellen als veraltet oder als nicht mehr aktiv unterstützt gilt⁸. Es wird daher empfohlen, die offizielle PayPal-Dokumentation zu konsultieren und alternative Lösungen zu prüfen, um sicherzustellen, dass die aktuellsten und unterstützten Funktionen für die Integration von PayPal in eine Flask-Anwendung verwendet werden.

Synergieeffekte zwischen Flask und dem Paypal Paython-SDK:

Flask und das PayPal Python SDK ergänzen sich gut. Sie ermöglichen die nahtlose Integration von PayPal-Zahlungen in Flask-Webanwendungen. Es gibt viele Ressourcen und Tutorials, die zeigen, wie man PayPal mit Flask integrieren kann, einschließlich Video-Tutorials⁹, Stack Overflow-Beiträgen¹⁰, und Webentwicklungstutorials¹¹. Das PayPal Python SDK bietet eine einfache Möglichkeit, PayPal-Zahlungen in Flask-Webanwendungen zu integrieren, indem Entwicklern eine API für den Zugriff auf die PayPal-Plattform und die Verarbeitung von Zahlungen zur Verfügung gestellt wird¹². Flask bietet eine einfache und klare Struktur, die die Entwicklung von Webanwendungen erleichtert, und seine Flexibilität und Anpassbarkeit ermöglicht es Entwicklern, ihre eigenen Komponenten und Tools auszuwählen und zu verwenden^{13 14}. Dies macht Flask zu einer attraktiven Option für unser Projekt, da wir eine einfache und flexible Lösung für die Integration von PayPal-Zahlungen in Webanwendungen benötigen.

Technischer Aufbau der Python App

In Python-Projekten wird oft eine virtuelle Umgebung verwendet, um Abhängigkeiten und Bibliotheken für das Projekt zu isolieren und zu verwalten. Eine virtuelle Umgebung ermöglicht es, verschiedene Versionen von Python und Bibliotheken für verschiedene Projekte zu verwenden, ohne dass es zu Konflikten kommt. Die virtuelle Umgebung kann mit dem Modul "venv" erstellt werden,

⁷ <https://github.com/paypal/PayPal-Python-SDK>

⁸ <https://developer.paypal.com/braintree/docs/guides/paypal/server-side/python>

⁹ <https://www.youtube.com/watch?v=HlwRzATH6iU>

¹⁰ <https://stackoverflow.com/questions/7394111/integrating-paypal-with-flask>

¹¹ <https://pythonprogramming.net/paypal-flask-tutorial/>

¹² <https://github.com/topics/paypal-rest-api?l=python>

¹³ <https://web-forward.de/2020/09/flask-vs-django-das-python-duell/>

¹⁴ <https://geekflare.com/de/flask-vs-django/>

das in Python 3 enthalten ist. Nachdem die virtuelle Umgebung erstellt wurde, kann sie aktiviert werden, um sie für das Projekt zu verwenden. In Flask-Projekten wird die virtuelle Umgebung häufig verwendet, um die Abhängigkeiten für die Flask-Anwendung zu verwalten. Sie kann mit dem Python-Befehl **python -m venv env** aufgerufen werden.

Integration des PayPal SDK:

Um das PayPal Python SDK in ein Python-Projekt zu integrieren, können die folgenden Schritte befolgt werden:

- Zuerst installiert man das PayPal Python SDK, indem man das Paket mit pip installiert:
 - **pip install paypalrestsdk**
- Danach ist es wichtig zu beachten, dass man eine eigene Client-ID und eigene Client Secret Keys von PayPal benötigt, um das SDK nutzen zu können. Diese können über das PayPal Developer Portal bezogen werden.

App.py:

Dieses Python-Skript startet die Flask-Anwendung. Die PayPal-Integration wird durch das PayPal Python SDK ermöglicht, das zuvor konfiguriert wurde. Die Flask-Webanwendung in unserem Projekt hat zwei Routen: Die erste Route rendert eine HTML-Datei (index.html), die als Startseite dient, und die zweite Route wird aufgerufen, wenn ein Benutzer eine Zahlung tätigt. In der zweiten Route wird eine PayPal-Zahlung erstellt und ausgeführt. Ist die Zahlung erfolgreich, wird eine Erfolgsmeldung ausgegeben, andernfalls eine Fehlermeldung. Die Integration von PayPal in Flask wird durch das PayPal Python SDK ermöglicht, das in diesem Abschnitt verwendet wird, um auf die PayPal-Plattform zuzugreifen und Zahlungen zu verarbeiten.

Index.html:

In dieser Datei wird die Flask-Webanwendung verwendet, um den PayPal-Zahlungsvorgang zu starten. Der Code enthält ein PayPal-Button-Element und zwei JavaScript-Skripte, die für die PayPal-Integration zuständig sind. Das erste Skript lädt die PayPal-Checkout-Bibliothek, während das zweite Skript die PayPal-Button-Logik enthält. Das Skript verwendet die PayPal RESTful API, um eine Zahlung zu erstellen und auszuführen. Die URLs für die Erstellung und Ausführung der Zahlung sind in den Variablen "CREATE_PAYMENT_URL" und "EXECUTE_PAYMENT_URL" definiert und können entsprechend angepasst werden. Der Code verwendet die Sandbox-Umgebung von PayPal, um Testzahlungen durchzuführen.

PayPal-Zahlungsauslösung Javascript React App

React:

React ist eine JavaScript-Bibliothek zur Erstellung von Benutzeroberflächen (UI) für Webanwendungen. Es wurde von Facebook entwickelt und ermöglicht die Entwicklung von interaktiven und reaktiven UI-Komponenten. Wir haben uns wegen der guten Dokumentation und der einfachen Integration für die JavaScript-Bibliothek React entschieden.

PayPal JavaScript SDK:

Das PayPal JavaScript SDK ermöglicht die Integration der PayPal-Zahlungsabwicklung in eine Website oder Anwendung. Es zeigt die von PayPal unterstützten Zahlungsmethoden an, um den Bezahlvorgang für Käufer zu vereinfachen. Das SDK kann zum Rendern von Schaltflächen, Symbolen für Zahlungsmethoden und Formularfeldern für Kredit- und Debitkarten verwendet werden. Es kann auch die Zulässigkeit einer bestimmten Zahlungsmethode überprüfen. Das SDK kann mit Hilfe von

Konfigurations- und Anpassungsparametern personalisiert werden, um die optimale Finanzierungsquelle und die den Käufern anzuzeigenden Schaltflächen zu bestimmen. Das SDK bietet außerdem eine umfassende Objekt- und Methodenreferenz sowie Optionen zur Leistungsoptimierung. Die Verwendung des PayPal JavaScript SDK war in unserem Javascript-Projekt obligatorisch.

Einbinden des PayPal JavaScript SDK:

Um das JavaScript SDK von PayPal in eine React-App zu integrieren, kann das Paket "`@paypal/react-paypal-js`" verwendet werden. Das Paket kann mit dem Befehl "`npm install @paypal/react-paypal-js`" in einem React-Projekt installiert werden. Anschließend können die erforderlichen Komponenten aus dem Paket in der React-Komponente importiert werden. Der Befehl "`npm create-react-app my-app`" erstellt ein neues React-Projekt mit dem Namen "my-app". Der Befehl führt das Paket "`create-react-app`" führt das Paket aus, das eine vorkonfigurierte React-Anwendung mit den erforderlichen Dateien und Verzeichnissen erstellt. Die erstellte Anwendung enthält eine grundlegende React-Struktur, einschließlich einer Startdatei, einer Hauptkomponente, einer CSS-Datei und einer HTML-Datei. Um die PayPal-Zahlungsauslösung in die Anwendung zu integrieren, muss das JavaScript-SDK von PayPal in die HTML-Datei eingebunden werden. Dazu wird ein Skript-Tag mit der Client-ID und der Währung eingefügt. Anschließend können die PayPal-Komponenten in der React-Komponente verwendet werden, um die Integration abzuschließen.

Index.html:

Der HTML-Code ist die Basis der React-Anwendung in unserem Projekt. Im Head-Bereich des Dokuments werden verschiedene Metadaten und Verweise auf externe Ressourcen wie das Favicon, die Ansichtsgröße, die Beschreibung der Website und der Verweis auf das Manifest für Progressive Web Apps definiert. Ein Script-Tag würde eingefügt, um das JavaScript-SDK von PayPal zu laden und die Client-ID und die Währung anzuzeigen. Im body-Bereich des Dokuments befinden sich ein noscript-Element, das eine Meldung anzeigt, wenn JavaScript deaktiviert ist, und ein div-Element mit der ID "root", das als Container für die gerenderte React-Komponente dient. Der Kommentar im HTML-Code enthält Hinweise zur Verwendung der Datei, wie z.B. das Hinzufügen von Webfonts, Meta-Tags oder Analysedaten, das Platzieren der gebündelten Skripte im body-Tag und die Verwendung der Befehle "`npm start`" oder "`npm run build`" für die Entwicklung bzw. Erstellung eines Produktionsbundles. Die Grundstruktur wurde vom PayPal Developer Portal übernommen und dient als universelle Vorlage für JavaScript-Projekte.

App.js:

In der Javascript-Datei erstellt der React-Code eine einfache Anwendung, die die Integration der PayPal-Kaufabwicklung ermöglicht. Der Code importiert die erforderlichen Module, einschließlich React und useState, sowie die Datei "PayPal" aus dem Komponentenordner. Die Funktion "App" definiert den Hauptteil der Anwendung. Mit Hilfe des useState-Hooks wird ein Zustand "checkout" und die zugehörige Set-Funktion "setCheckOut" erzeugt, wobei der Initialwert des Zustands auf "false" gesetzt wird. In der Return-Anweisung der Funktion "App" wird ein div-Element mit der Klasse "App" erzeugt. Wenn der Zustand "checkout" den Wert "true" hat, wird die Komponente "PayPal" gerendert. Andernfalls wird eine Schaltfläche gerendert, die den Zustand "checkout" auf "true" setzt, wenn sie angeklickt wird. Der Code exportiert die Komponente "App" als Standardexport. Der Quellcode ermöglicht es dem Benutzer, auf den "Checkout"-Button zu klicken, um die PayPal-Zahlungsauslösung zu starten. Wenn der Benutzer auf die Schaltfläche klickt, wird der "Checkout"-Status auf "true" gesetzt und die "PayPal"-Komponente wird gerendert, die die eigentliche PayPal-Zahlungslogik enthält.

PayPal.js:

Innerhalb der Datei wird der React-Code erstellt, der die Komponente namens "Paypal" enthält, die die Integration von PayPal-Checkout in eine React-Anwendung ermöglicht. Der Code importiert die notwendigen Module, einschließlich React, useRef und useEffect. Die Funktion "Paypal" wird definiert und stellt den Hauptbestandteil der Komponente dar. Mit Hilfe des useRef-Hooks wird eine Referenz auf das PayPal-Element erstellt. Mit Hilfe des useEffect Hooks wird das PayPal Checkout SDK initialisiert und konfiguriert. Es wird ein PayPal-Button erzeugt, der beim Anklicken eine Zahlungsanforderung ausgibt. Wenn die Zahlung autorisiert wird, wird die Zahlung ausgeführt und die Zahlungsdetails werden in der Konsole ausgegeben. Die Komponente gibt das PayPal-Element zurück, das von der useRef-Funktion referenziert wird.