

Das Grundgerüst einer Seite in Flask

Flask ist ein Python Framework, das es einem erlaubt Webseiten in Python zu erstellen.

Zu Anfang: Es gibt zwei Arten eine Seite in Flask zu erstellen. Ich beschreibe hier die aufwendigere, die allerdings bei größeren Seiten deutlich übersichtlicher ist.

Wichtig zu wissen ist, dass wir Flask nur auf einem eigenständigem Server verwenden können, nicht auf angemietetem Webspaces. Flask benötigt **kein** Apache und PHP. Es ist jedoch möglich das ganze mit MySQL und SQLite zu verbinden, auf das ich allerdings hier nicht genauer eingehen werde.

Zu erst müssen wir sichergehen, dass Python auf dem Server installiert ist. Bei einem Linux-Server ist dies standardmäßig der Fall. Nun müssen wir ein Tool zum Installieren von Python Erweiterungen installieren, so wie Flask.

Dazu geben wir als Root folgendes ein:

Code

```
apt-get install python-pippip install Flask
```

Nun können wir mit der Erstellung der eigentlichen Seite beginnen.

Wir brauchen für das ganze eine bestimmte Ordnerstruktur:

Code

```
|-> runserver.py |-> name_unserer_seite/ |-> __init__.py |-> somename.py |-> templates
```

Jetzt füllen wir die Seiten mit unserem Code (Bedenke: Alles hinter # ist in Python ein Zitat und dient hier der Erklärung)

runserver.py:

Python: runserver.py

```
from name_unserer_seite import app #Get Sites from __init__.py app.run(host='0.0.0.0', port=
```

__init__.py:

Python: __init__.py

```
# coding: utf8from flask import Flask #Import Flask moduleapp = Flask(__name__) #Important V
```

somename.py:

Python: somename.py

```
from sslcon import app #import app from __init__.pyfrom flask import render_template #import
```

sometemplate.html:

HTML: somemplate.html

```
<!-- Das ist jetzt ein ganz normales HTML Template -->
<head>
<title>{{title}}</title>!--{{}}kommt von Jinja und fügt hier die {{title}} festgelegte Variable ein-->
</head>
<body>
  {{ text }}  <!-- Hier das ganze mit dem Text -->
</body>
```

So, ich denke für den Anfang sollte das genug sein. Wenn Bedarf herrscht, kann ich gerne den weiteren Umgang beschreiben, denn mit Flask (und Jinja(ist in Flask inbegriffen)) lässt sich viel machen. Wir können z.B. auch Daten via JSON weitergeben, oder das ganze mit einer SQLite, MySQL, etc. Datenbank verbinden.

Da dieses jetzt aber den Rahmen hier deutlich sprengen würde, warte ich auf eure Rückmeldungen und werde mich dann nach euren Wünschen richten.