

Django 3 telepítése osztott tárhelyre

Ha még nem telepített Django-t a webtárhelyére (rendszergazdai hozzáférés nélkül), akkor a telepítési folyamat több lépésben történik ami bonyolultabbá teszi a az első telepítést. A fő különbség az, hogy nem használható a beépített Django szerver (pl. `python manage.py runserver`) weboldal futtatásához (ezenkívül nem ajánlott éles használatra sem). Ahhoz, hogy a Django kiszolgálhassa weboldalt, egy „közbenső szoftvert” kell használnia, amely a hagyományos szerverkérelmeket Django-kérésekké alakítja át. A cPanel-t futtató webtárhelyek esetében ez a közbenső szoftver a [Phusion Passenger](#).

Mielőtt a telepítést elkezdenénk, kérjük vegye figyelembe, hogy a Django 2.2 és újabb verziók natív módon nem futhatnak a webtárhelyeken az SQLite újabb verziója nélkül, több esetben jelentkezett ez a probléma ami miatt telepíteniük kellett a Django 2.1-es vagy régebbi verzióját. Feltéve, hogy már megpróbálta telepíteni a Django 3 -at a webtárhelyére, és megpróbálta futtatni, az alábbi hibát kaphatta:

```
$ python manage.py migrate  
django.core.exceptions.ImproperlyConfigured: SQLite 3.8.3 or later is required (found 3.7.17).
```

Abban az esetben ha Önnél is jelentkezett a fenti probléma kéjük, hogy olvassa el az [SQLite frissítési útmutatót](#), amely végigvezeti az SQLite legújabb verziójának webtárhelyen történő futtatásának lépésein.

PYTHON ALKALMAZÁS LÉTREHOZÁSA

A cPanel felületén keresztül történik a beállítás ez egyes cPanel telepítések esetében eltérhet.

1. A cPanel-ben minden Python-környezet egy domain névhez vagy aldomain névhez van társítva. Győződjön meg arról, hogy domain neve fel van véve a tárhelyére, illetve ha aldomain név alatt szeretné használni akkor azt létrehozta e.
2. A cPanel keresősávba írja be a `python` parancsot, és nyomja meg az Enter billentyűt. A Python webes alkalmazások oldalra érkezik, ahol virtuális környezeteket hozhat létre a Pythonban kifejlesztett webes alkalmazásaihoz, és automatikusan összekapcsolhatja azokat a Phusion Passenger webservereivel.

3. Új virtuális környezet létrehozásához kattintson a jobb oldalon található „Alkalmazás létrehozása” gombra: ahol az alábbi űrlap jelenik meg. Töltse ki az alábbi minta alapján:

Python alkalmazás beállítása

1. Válassza ki a virtuális környezethez használt Python verziót, javasoljuk a legújabb verzió használatát (jelenleg: 3.7.11)
2. Írja be a domain nevéhez vagy aldomain nevéhez társított mappa nevét, amelyet a webes alkalmazás fájljainak tárolására használnak
3. Válassza ki a webes alkalmazáshoz használt domain nevet vagy aldomain nevet. Hozzáadhat egy URL -címet (nem kötelező), például a `www.domain.hu/django` címet
4. Ez a 2 mező a Phusion Passenger alkalmazáshoz tartozik, amely futtatni fogja a webes alkalmazást. Hagyja üresen (automatikusan kitölti őket a felület)
5. Írja be a fájlnevet és a Phusion Passenger naplófájl elérési útját. Hasznos lesz tudni annak helyét az alkalmazás állapotának ellenőrzéséhez (cserélje ki a „felhasználót” a saját cPanel felhasználónevére)

Ha elkészült, kattintson a „Létrehozás” gombra. A cPanel automatikusan létrehozza a Python virtuális környezetét, és összekapcsolja a Passenger-el.

4. Ezt követően csatlakozzon a szerverhez SSH-val a kedvenc terminál kliensével vagy a cPanel beépített Terminál kliensének használatával (írja be a „terminált” a felső keresősávba). Aktiválja a virtuális környezetet a következő paranccsal:

```
$ source /home/hash/virtualenv/name-of-env/3.7/bin/activate
```

Cserélje ki a felhasználói nevet (hash) a cPanel felhasználónevére és a name-of-env részt az előző lépésben megadott névre. Mielőtt folytatnánk, navigáljon a webes alkalmazás mappájához:

```
(name-of-env) $ cd /home/hash/name-of-folder
```

5. Most már elkezdhetjük telepíteni a Django 3 programot, és elindíthatjuk a környezetében. Először frissítsük a pip-et, ha még nem tettük volna meg:

```
(name-of-env) $ pip install --upgrade pip
```

DJANGO 3 TELEPÍTÉSE:

```
(name-of-env) $ pip install django
```

A telepítés után indíthat el egy projektet (cserélje le a saját_projekt változót saját projekt nevére):

```
(name-of-env) $ django-admin startproject saját_projekt
```

Lépjen a projekt mappájába, majd futtassa le a migrációs feladatokat:

```
(name-of-env) $ cd saját_projekt
(name-of-env) $ python manage.py migrate
```

Ha a fenti lépés hibát adna vissza, vagy az SQLite verzióhibát ad vissza, akkor az SQLite újabb verzióját kell használnia, amint azt ebben az útmutatóban korábban említettük.

6. Az utolsó lépések a következők:

- Hozzon létre egy Rendszergazda felhasználót (=admin), ahhoz hogy be tudjon jelentkezni a Django adminisztrációs részébe
- Majd frissítse a Passenger által használt fájlt az új projekt belépési pontjaként

A Django rendszerben a rendszergazda létrehozásához adja ki a következő parancsot:

```
(name-of-env) $ python manage.py createsuperuser
```

Adjon meg egy felhasználónevet, e-mail címet és jelszót (kétszer).

Ezt követően meg kell mondjuk a Passenger-nek, hogy hol található az új alkalmazás, ahhoz hogy betöltse és megjelenítse Django demóoldalát a Python alkalmazás beállításaiiban megadott (3. lépésben megadott) URL-címen. Előtte azonban ellenőrizzük, hogy a Passenger jól működik-e a szerveren.

Nyissa meg az URL-t, ahol a Python csupasz oldalát kell látnia

Python kezdő oldal

Ha a Passenger jól működik akkor a fentihez hasonló oldalt kell látnia. Ha nem akkor kérjen segítséget ügyfélszolgálatunktól.

A PYTHON ALKALMAZÁS MÓDOSÍTÁSA

Navigáljon vissza az alkalmazás kezdőkönyvtárához, és szerkessze meg a `passenger_wsgi.py` fájlt:

```
(name-of-env) $ cd ..
(name-of-env) $ nano passenger_wsgi.py
```

A `sys.path` sor alatt adja hozzá az alábbi sort (a másodikat):

Ügyelve arra, hogy az alkalmazás nevét módosítsa arra amelyet a létrehozásakor megadott.

```
sys.path.insert(0, os.path.dirname(__file__)) # Már szerepel a fájlban ezt a sort ne törölje ki
sys.path.insert(0, 'saját_projekt')          # Ezt a sort kell hozzáadnia
```

Ezután távolítsa el az alábbi a sorokat (a nano szerkesztőben görgessen lefelé a „def” sorig, és nyomja meg a Ctrl+K billentyűkombinációt, ahányszor sort kell távolítania):

```
def application(envIRON, start_response):  
    start_response('200 OK', [('Content-Type', 'text/plain')])  
    message = 'It works!\n'  
    version = 'Python %s\n' % sys.version.split()[0]  
    response = '\n'.join([message, version])  
    return [response.encode()]
```

Adja hozzá a következő sorokat az előbb eltávolított sorok helyett (a nano szerkesztőben a beillesztés parancsikonja ugyanaz, mint a rendszeren, pl. Ctrl+V):

```
import saját_projekt.wsgi  
application = saját_projekt.wsgi.application
```

Mentse el a módosításokat (a nano szerkesztőben a mentés billentyűkombinációja a: Ctrl+O) ugyan ezen a néven.

A PASSENGER ALKALMAZÁS ÚJRAINDÍTÁSA ÉS A DJANGO HASZNÁLATÁNAK MEGKEZDÉSE

Utolsó lépésként újra kell indítani a Passenger alkalmazást. Ehhez írja be a python szót a cPanel felső keresősávjába, és nyomja meg az Enter billentyűt a Python webes alkalmazások oldal megnyitásához. Keresse meg a webes alkalmazást a listában, és indítsa újra (az óramutató járásával ellentétes irányú nyíl ikon). Ezt követően navigáljon a Django alkalmazásnál megadott URL címhez, ahol látnia kell a Django bemutató weboldalát.

Django 3
Image not found or type unknown
Django 3

Innentől kezdve a Django 3 használható, a /admin URL címet beírva juthat el a Django 3 adminisztrációs felületére.

Változat #1

DotRoll Tudásbázis hozta létre 21 szeptember 2023 09:55:19

DotRoll Tudásbázis frissítette 21 szeptember 2023 09:57:02