

Python Django telepítése és futtatása cPanelen

A [Django](#) egy Python-alapú keretrendszer amelynek segítségével gyorsan és egyszerűen készíthetünk weboldalakat. Ezen leírás segítségével beállítható a Django keretrendszer.

A beállítást követően egy működő Django oldalt kapunk amely a következőkre biztosít lehetőséget:

- Statikus oldalak betöltése egy adott domain néven
- Django adminisztrációs felület betöltése
- SQLite adatbázis használat

PYTHON ALKALMAZÁS LÉTREHOZÁSA A CPANEL FELÜLETEN

1. Jelentkezzen be a cPanel fiókjába
2. Keresse meg a **Python alkalmazás beállítása** menü pontot, és kattintson rá
3. Python alkalmazás beállítása
 1. A megjelenő oldalon a kattintson az **Alkalmazás létrehozása** gombra
 1. A Python verziónál (**Python version**) válassza ki a **3.8.1**-es verziót.
 2. Adja meg az alkalmazás könyvtárát (**Alkalmazás root /home/cpanel_username/**), például: alkalmazas
 3. Az **Alkalmazás URL** részénél válassza ki a domain nevet, majd az utána következő mezőt hagyja üresen.
 4. Hagyja üresen az **Alkalmazás indító fájl** illetve a **Alkalmazás belépési pont** mezőket is.

A cPanel ebben az esetben automatikusan létre fogja hozni a *passenger_wsgi.py* fájlt és az alap értelmezett indítási környezetet.

1. 1. 5. A **Passenger napló fájl** felirat után található mezőben megadható a napló fájl neve amely segítséget adhat a hibakeresésben.

6. Ezt követően kattintson a jobb felső sarokban található **Létrehozás** gombra. A cPanel ezt követően létrehozza a Python alkalmazást és beállítja a környezeti változókat.
7. Az oldal felső részén megjelenő **Enter to the virtual environment. To enter to virtual environment, run the command:** után található parancsot másolja ki, mivel az alkalmazás beállításához szüksége lesz erre a parancsra.

A DJANGO BEÁLLÍTÁSA

A Python alkalmazás létrehozást követően az alábbi feladatokat parancssorból kell végrehajtani (A cPanel felületén elérhető **Terminál** menü pont is használható erre a célra, de SSH kliens segítségével is csatlakozhat a tárhelyhez.):

- A Django telepítése
- A Django projekt létrehozása és beállítása
- A Passenger beállítása a Django-val történő használathoz

A fenti lépések menete:

1. Jelentkezzen be a tárhelyére SSH-n keresztül, vagy indítsa el a cPanel felületéről elérhető **Terminál** menü pontot.
2. Aktiválja a virtuális környezetet az előzőleg kimásolt parancs beillesztésével és lefuttatásával, például:

```
source /home/cpanel_username/virtualenv/alkalmazas/3.8/bin/activate && cd /home/cpanel_username/alkalmazas
```

A parancssor most a (**alkalmazas: 3.8**) felirattal kezdődik, jelezve, hogy a alkalmazas virtuális környezetben dolgozik a Python 3.8 alkalmazással. A cikkben szereplő összes alábbi parancs feltételezi, hogy Ön a Python virtuális környezetben dolgozik. Ha kijelentkezik az SSH-munkamenetből (vagy deaktiválja a virtuális környezetet a **deactivate** paranccsal), az alábbi lépések bármelyikének végrehajtása előtt győződjön meg róla, hogy újra aktiválta a virtuális környezetet.

3. A Django telepítéséhez adja ki a következő parancsokat:

```
cd ~  
pip install django==2.1.8
```

A Django telepítés ellenőrzéséhez adja ki a következő parancsot:

```
django-admin --version
```

4. Hozza létre a Django projektet az alábbi parancs kiadásával:

```
django-admin startproject alkalmazas ~/alkalmazas
```

5. Az alábbi parancsok kiadásával hozza létre a statikus projekt könyvtárakat:

```
mkdir -p ~/alkalmazas/templates/static_pages
mkdir ~/alkalmazas/static_files
mkdir ~/alkalmazas/static_media
```

6. Valamelyik szövegszerkesztő segítségével nyissa meg a `~/alkalmazas/alkalmazas/settings.py` fájlt, és végezze el az alábbi módosításokat a fájlban:

- Keresse meg az **ALLOWED_HOSTS** sort, majd módosítsa az alábbiak szerint. cserélje ki az `example.com` domain nevet a saját domain nevére:

```
ALLOWED_HOSTS = ['example.com']
```

- Keresse meg a **TEMPLATES** blokkot, majd módosítsa az alábbiak szerint:

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]
```

- Keresse meg a **STATIC_URL** sort, és adja hozzá a következő sorokat:

```
STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static_files')

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, "static_media")
```

7. Nyissa meg egy szövegszerkesztő segítségével a `~/alkalmazas/alkalmazas/urls.py` fájlt majd törölje ki a teljes tartalmat, és illessze be az alábbi tartalmat:

```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static
from django.conf.urls import url
from django.views.generic.base import TemplateView
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    url(r'^$', TemplateView.as_view(template_name='static_pages/index.html'), name='home'),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```

8. Nyissa meg egy szövegszerkesztő segítségével a `~/alkalmazas/passenger_wsgi.py` fájlt majd törölje ki a teljes tartalmat, és illessze be az alábbi tartalmat:

```
import os
import sys

import django.core.handlers.wsgi
from django.core.wsgi import get_wsgi_application

# Set up paths and environment variables
sys.path.append(os.getcwd())
os.environ['DJANGO_SETTINGS_MODULE'] = 'alkalmazas.settings'

# Set script name for the PATH_INFO fix below
SCRIPT_NAME = os.getcwd()

class PassengerPathInfoFix(object):
    """
    Sets PATH_INFO from REQUEST_URI because Passenger doesn't provide it.
    """
    def __init__(self, app):
        self.app = app

    def __call__(self, environ, start_response):
        from urllib.parse import unquote
        environ['SCRIPT_NAME'] = SCRIPT_NAME
        request_uri = unquote(environ['REQUEST_URI'])
        script_name = unquote(environ.get('SCRIPT_NAME', ''))
        offset = request_uri.startswith(script_name) and len(environ['SCRIPT_NAME']) or 0
        environ['PATH_INFO'] = request_uri[offset:].split('?', 1)[0]
        return self.app(environ, start_response)

# Set the application
application = get_wsgi_application()
application = PassengerPathInfoFix(application)
```

9. Ezt követően hozza létre az alapértelmezett `index.html` fájlt a `~/alkalmazas/templates/static_pages` könyvtárban belül. ez a fájl egy alapértelmezett statikus fájl amely csak a **Hello World** szöveget jeleníti meg.
10. Adja ki az alábbi parancsot:

```
python ~/alkalmazas/manage.py migrate
```

11. Hozza létre az adminisztrátor felhasználót az alábbi parancs segítségével:
Ehhez először adja ki az alábbi parancsot:

```
python ~/alkalmazas/manage.py createsuperuser
```

Amikor az alkalmazás bekéri az adminisztrátori felhasználói nevet (**Username**) akkor gépelje be az új felhasználói nevet majd nyomja le az enter billentyűt.

Amikor az alkalmazás bekéri az email címet (**Email address**) akkor gépelje be az email címet majd nyomja le az enter billentyűt.

Amikor az alkalmazás bekéri az adminisztrátori jelszót (**Password**) akkor gépelje be az új jelszót majd nyomja le az enter billentyűt.

12. A következő parancs kiadásával összeállíthatja a statikus tartalmat:

```
python ~/alkalmazas/manage.py collectstatic
```

Amikor az alkalmazás rákérdez arra, hogy felülírja a meglévő fájlokat akkor írja be a **yes** szót, majd nyomjon entert.

13. A cPanel felületén indítsa újra a Python alkalmazást.

- Ehhez a cPanel felületén keresse meg a **Python alkalmazás beállítása** menü pontot, és kattintson rá
- A **Web alkalmazások** fülénél keresse ki az adott alkalmazást, majd a **Műveletek** oszlopban kattintson az **Alkalmazás újraindítása** gombra

14. Utolsó lépésként tesztelhető az alkalmazás a beállított URL címen keresztül:

- Böngészőben nyissa meg a weboldalt (<http://www.pelda.hu/>) ahol az index.html fájlnak kell megjelennie.
Fontos, hogy a pelda.hu domain nevet cserélje ki az Ön által használt domain névre.
- Következő lépésként nyissa meg a <http://www.pelda.hu/admin/> címet amelyen a Django alkalmazás adminisztrációs felületének kell megjelennie.
Fontos, hogy a pelda.hu domain nevet cserélje ki az Ön által használt domain névre.

Abban az esetben ha a böngészőben nem jelenne meg a kívánt tartalom akkor próbálja meg az alkalmazást parancssorból futtatni a következő parancs segítségével:

```
python ~/alkalmazas/passenger_wsgi.py
```

Ebben az esetben a konzolon meg fog jelenni a hiba oka. Ha a Python alkalmazás beállítása során megadta a naplózási útvonalat akkor a hibák a napló fájlban is rögzítésre kerülnek.

TOVÁBBI INFORMÁCIÓ

Mivel a jelenlegi alkalmazás már fut így ténylegesen el lehet kezdeni fejleszteni a saját alkalmazásunkat, amelyhez az alábbi hivatkozásokon található további segítséget:

- A Django hivatalos dokumentációját elérheti a következő weboldalon:
<http://docs.djangoproject.com>.
- A Django kiterjesztésekkel kapcsolatosan látogassa meg a következő oldalt
<https://github.com/django-extensions/django-extensions>.
- A *south* library egy népszerű kiegészítő az adatbázis migrációkhoz amelyről további információt kaphat az alábbi oldalon <https://pypi.python.org/pypi/South>.
- A *fabric* library segít egyszerűsíteni az alkalmazások fejlesztését ezzel kapcsolatosan részletesebb információt itt találhat <http://docs.fabfile.org>.

Változat #1

DotRoll Tudásbázis hozta létre 2023-09-21 10:00:40 CEST

DotRoll Tudásbázis frissítette 2023-09-21 10:01:42 CEST