

TDP003 Projekt: Egna datormiljön

Installationsmanual för Portföljsystem

Författare

Arturas Alekandrausaksa, (NI) Chirster Vesterlund, Benjemin Graf, Vidar Wesrfelt, Gustav P Svensson, Dylan Mäenpää, Denis I. Blazevic, Alexander Jonsson, Deniz Ayar, Vera Antonov, Sokrates Lamprou, Niko Lehto, Albin Vedin, Adam Sterner, Mats Johansson, Jesper Olofsson, David Hilm, Viktor Brandt, David Rahim, Frans Bergström, Carl Lorentsson, Joakim Johansson, Erik Westerlund, Love Bäckman

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Författare	Datum
2.70	Separerade ut revisionshistoriken till en egen fil för att fixa formattering med ltxtable och longtable.	Love Bäckman	160929
2.60	La hur man startar debuggern om servern inte startar. Utvecklade lite om Jinja2 så man vet vad det gör.	Erik Westerlund	160929
2.50	La till författare i revisionbeskrivningen, samt la till ny revisionshistorik och författare från nya commits.	Frans Bergström	160928
2.40	Uppdaterat så pdf:en är synkad med text-filen. Lagt till författare från tidigare commits.	David Hilm	160927
2.30	Ändrade Jinja2 förklaring.	Viktor Brandt	160926
2.30	Ändrade mikroramverk.	David Rahim	160926
2.20	La till redovisningshistorik och beskrivning.	Carl Lorentsson	160926
2.10	La till författare.	Arturas Aleksandrauskas	160926
2.02	Formaterade klart resten.	Denis I. Blazevoc	160925
2.01	Formeterade klart koden.	Denis I. Blazevoc	160925
2.00	Fixade radbrytning och styckeinteldning.	Denis I. Blazevoc	160925
1.91	Korrigerade information om Flask.	Mats Johansson	160923
1.90	La till slutparantes.	Mats Johansson	160923
1.80	La till information om loggfilen.	Benjamin Fischer	160922
1.70	Fixade formatering.	Denis I. Blazevic	160922
1.61	La till installationsinstruktioner för olika linuxdistrubitioner.	Adam Sterner	160922
1.60	Delade upp installera i python till vilken version och Linux	Adam Sterner	160922
1.50	Fixade formattering relaterad till lstlisting.	Vidar Westfelt	160921
1.43	Kapslade in flera kodstängar i lstlising.	Victor Friberg	160921
1.42	Kapslade in en kodsträng i lstlisting.	Victor Friberg	160921
1.41	Mer informaiton om python3 och pip användande.	Vidar Westfelt	160921
1.40	La till python3 och pip användande.	Vidar Westfelt	160921
1.31	Fixade radbyte.	Albin Vedin	160921
1.30	La in kod för att uppdatera de paket som inte är installerade.	Albin Vedin	160921
1.20	La till ett stycke om felsökning och underhåll.	Niko Lehto	160921
1.10	La till information om Python,Flask och Jinja2.	Sokrates Lamprou	160921
1.03	Rättade ett stavfel.	Vidar Westlund	160920
1.02	Korrigerade sektionindelning.	Vidar Westlund	160920
1.01	La till gitignore för build-fil.	Vidar Westlund	160920
1.00	Tog bort onödig informaiton om pip.	Vidar Westlund	160920
0.90	La till information för felhantering ifall flask inte startar.	Christer Vesterlund	160920
0.81	Ändrade kommandot för pip install.	Vera Antonov	160920
0.80	Rättade mindre stavfel och la till en installationsdel om pip ej finns installerat.	Vera Antonov	160920
0.70	La till felhantering vid kopiering med artefakter.	Deniz Ayar	160920
0.61	Ändrade tillbaka ett feländrat ord.	Alexander Jonsson	160919
0.60	La till mer info om hur man startar servern, ändrade dokumentstruktur och små fel i texten	Alexander Jonsson	160919
0.50	La till ett annat sätt att starta servern och förtydligade en mening.	Denis I. Blazevic	160919
0.40	La till instruktioner om hur man testat flask.	Dylan Mäenpää	160919
0.30	La till informaiton om hur man startar servern.	Gustav P Svensson	160919

0.22	Uppdaterade datum.	Vidar Westfelt	160919
0.21	Uppdaterade formatering.	Vidar Westfelt	160919
0.20	La till first draft	Vidar Westfelt	160919
0.10	La till installationsfil för pip	Benjamin Graf	160919

2 Installera verktyg

För att komma igång behöver du se till att du har verktygen `python3`, `Pip`, `Flask` och `Jinja2`.

Python är ett programmeringsspråk som du behöver installera för att kunna använda Flask och Jinja2 medan Pip är den pakethanterare som vi behöver för att kunna installera de två paketen. Flask är ett mikroramverk för webbutveckling som även har en inbyggd webbserver funktion som man kan använda medans utveckling av portföljssystem pågår. Tillsammans med Flask tillkommer Jinja2 templating. Jinja2 är ett modernt mallspråk för Python och används för att generera html output som skickas till användaren. Detta betyder att du inte måste uppdatera html-kod för varje gång du ska ladda upp något nytt på din sida, utan det löser Jinja2. Innan du påbörjar processen av att installera de paket som behövs kan det vara bra att se till att alla de paket du redan har installerade är uppdaterade.

Detta gör du enkelt genom att i terminalen skriva in:

```
$ sudo apt-get update
```

2.1 Installera Python 3

2.1.1 Vilken version har vi?

Python 3 kan installeras antingen som `python` eller `python3`. Python 3 finns oftast förinstallerat på systemet som `python3`. Kolla vilka versioner du har. Använd den som motsvarar senaste versionen av Python 3:

```
$ python -V
> Python 2.7.12
$ python3 -V
> Python 3.5.2
```

Om ditt resultat ser ut som ovan, använd framöver `python3`. Vi kommer att skriva `python3` i resten av manualen, men om du har Python 3 installerat som `python`, använd den i stället.

2.1.2 Linux

Om Python 3 saknas så rekommenderas installation med hjälp av din distributions pakethanterare (i Linux Mint är detta `apt-get`):

Linux Mint eller Ubuntu (`python3` ska vara förinstallerat):

```
$ sudo apt-get python3
```

Arch-Linux (`python3` är standard i Arch-Linux):

```
$ sudo pacman -S python
```

Fedora:

```
$ dnf install python3
```

openSUSE:

```
$ zypper install python3
```

2.2 Installera pip

Om du har Python 3.4 eller senare så ska pip komma förinstallerat tillsammans med Python, grattis! Det finns flera sätt att köra pip. Ofta så går det att skriva `pip` eller `pip3`. För att vara säkra på att vi kör rätt pip version, skriver vi i terminalen.

```
$ python3 -m pip -V
>pip 8.1.2 from /usr/lib/python3.5/site-packages (python 3.5)
```

Se till att du använder den version av `pip` som är installerad för Python 3, och inte den som är installerad för Python 2!

Om `pip` inte finns installerat så kan du installera den själv genom att hämta filen `get-pip.py` från:

<https://packaging.python.org/installing/>

Navigera sedan till samma mapp som den hämtade filen ligger i, starta terminalen och kör kommandot:

```
$ sudo python3 get-pip.py
```

Du kan även navigera direkt till mappen i terminalen genom att skriva in så här (byt ut `Downloads` till den rätta mappen):

```
$ cd ~/Downloads
$ sudo python3 get-pip.py
```

Detta kommer installera eller uppdatera din pip-installation.

2.3 Installera Flask och Jinja2

Installera flask med hjälp av pip, så följer Jinja2 med automatiskt.

```
$ sudo python3 -m pip install flask
```

2.3.1 Testa flask

Skapa en fil som heter `test_flask.py` och öppna den i valbar texteditor. Skriv in denna kod som definierar en server med en startsida som innehåller en enkel text: "Flask hälsar dig välkommen!".

```
# -*- coding: utf-8 -*-
```

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def hello():
```

```
    return "Flask hälsar dig välkommen!"
```

```
if __name__ == "__main__":
```

```
    app.run()
```

Om man får ett fel i stil med:

```
user@localhost:~$ python3 test_flask.py
File "test_flask.py", line 2
from flask import Flask
^
```

Då kan extra artefakter lagts in vid en eventuell kopiering av texten ovan. Prova då att skriva in det för hand istället.

Ett annat fel som kan uppstå är att flask inte startar överhuvudtaget. Då kan felet vara att vid installationen av `pip` har användaren skrivit:

```
$ sudo python get-pip.py
```

istället för:

```
$ sudo python3 get-pip.py.
```

För att komma tillrätta med detta problem så körs detta kommando istället:

```
$ sudo python3 -m pip install flask.
```

Prova att starta programmet igen och nu bör det fungera.

3 Starta servern

Du startar själv servern i terminalen. Navigera till katalogen där din serverfil är. Starta sedan din server genom att köra ett av följande sätt (byt ut `test_flask.py` till din egna serverfil):

```
export FLASK_APP=test_flask.py
flask run
```

eller

```
python3 test_flask.py
```

När servern har startat ska något i denna stil dyka upp i terminalen:

* Running on `http://127.0.0.1:5000/` (Press CTRL+C to quit)

Skriv in adressen i din webbläsare så kan du testa om din kod fungerar (se bild 1). Om ovanstående ej

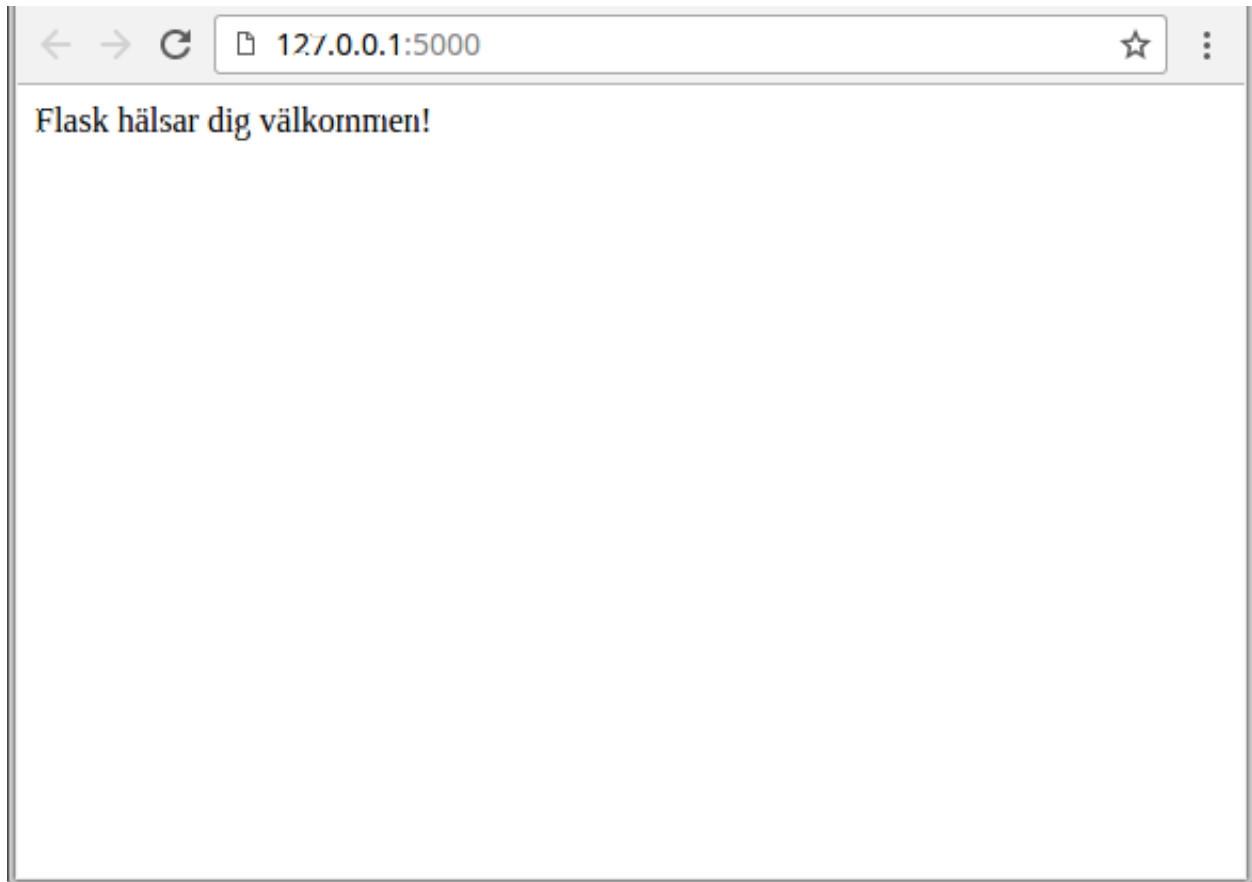


Figure 1: Flask server test

fungerar, se sektion 4.2

4 Felsökning och underhåll

4.1 Logging

Logging av varje anrop till programmet sparas på en separat textfil(Logg.txt). Denna fil hittar du i samma mapp som programfilen är placerad i.

I loggen visas information såsom tid för anrop, vilka parametrar skickades in med anropet och vilka parametrar skickades ut. Här kan även den sökväg på källfilen där logganropet skedde i och funktionsnamnet som innehåller loggningsanropet visas. Utöver detta kan kodraden som loggningsanropet skedde på också visas. Eventuella felmeddelanden som uppstår skrivs också i loggfien. Varje loggmeddelande får var sin flagga för att visavilken typ av meddelande det är (se tabell 2).

Table 2: Flaggor för loggmeddelanden.

Flagga	Innebörden av flaggan
INFO	Anropet har skett utan problem.
ERROR	Något har gått snett men programmet kan fortsätta.
CRITICAL	Ett fel som orsakar att programmet ej kan fortsätta.

4.2 Debugging

Om det inte fungerar att starta din server, kan du starta Flasks debugger. Detta gör du med hjälp av denna kod:

```
export FLASK_DEBUG=1
flask run
```

Här kan du gå igenom koden och få mer information om varför det inte gick att starta servern.