

TDP003 Projekt: Egna datormiljön

Installationsmanual

Författare

Erik Lindow, erili445@student.liu.se

Erik Edling, eried975@student.liu.se

Examinator

Klas Arvidsson, klas.arvidsson@liu.se

Revisionshistorik

ver.	Revisionsbeskrivning	Datum
1.0	Dokumentet klart	141007
1.1	Lade till figurer under punkt 2.1 och 2.2.1	141009
1.2	Lade till beskrivning av hur man installerar python3.4 under punkt 1.1 Lade till punkt 2.1 som Mappstrukturer. Lade till Obligatoriska fält för JSON filen i punkt 2.3.3	141016

1. Installation av portfolion

1.1. Systemkrav och programvarukrav

För att kunna köra och använda portfolion på operativsystemet Linux Mint, se till att Python 3.4 är installerat, för att göra detta på ett smidigt sätt kan man skriva följande i terminalen:

`"python3 -V"`(observera att V ska vara en stor bokstav). Finns inte python3 på systemet kan du skriva följande:

`"sudo apt-get update"` - Alltid bra att göra så man får de nyaste paketlistorna. Efter det kan du skriva in följande för att installera python3:

`"sudo apt-get install python3.4"`

När python3 är installerat uppnås de grundläggande systemkraven för att använda portfolion däremot behövs fortfarande följande två program installeras för att kunna använda den:

- Flask
- Jinja2

Flask och Jinja2 behövs installeras för att portfolion på ett smidigt sätt ska kunna samarbeta mellan den visuella delen av portfolion(HTML och CSS) och den mer underliggande funktionella delen(Datalagret i python) .

I denna installationsguide kommer det visas hur man installerar Flask och Jinja2. Detta görs i nästa avsnitt 1.2.

1.2. Installera Flask

Innan vi börjar installera Flask kan det vara så att "pip" behöver installeras om det inte redan finns. För att installera "pip" kan man göra på följande vis:

1. Ladda ner filen på följande sida och spara den i en varlfri mapp på datorn:
<https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py>
2. I terminalfönstret, gå till mappen du sparat filen på. För att göra detta kan man skriva följande kommando: `"cd <hela sökvägen>"`
3. Skriv följande kommando för att installera "pip": `"sudo python3 get-pip.py"`
4. För att sedan installera själva flask skriver man följande kommando: `"sudo pip3.4 install flask"`. I detta steg installerades även Jinja2 vilket betyder att vi uppnått de grundläggande kraven för att använda sig utav portfolion.

1.3. Nedladdning av portfolion

Nedladdning av portfolion kan göras på följande sida:

<http://www-und.ida.liu.se/~erili445/MyPortfolio.zip>

Den behöver du bara ladda ned och lägga i valfri destination.

2. Underhåll av portfolion

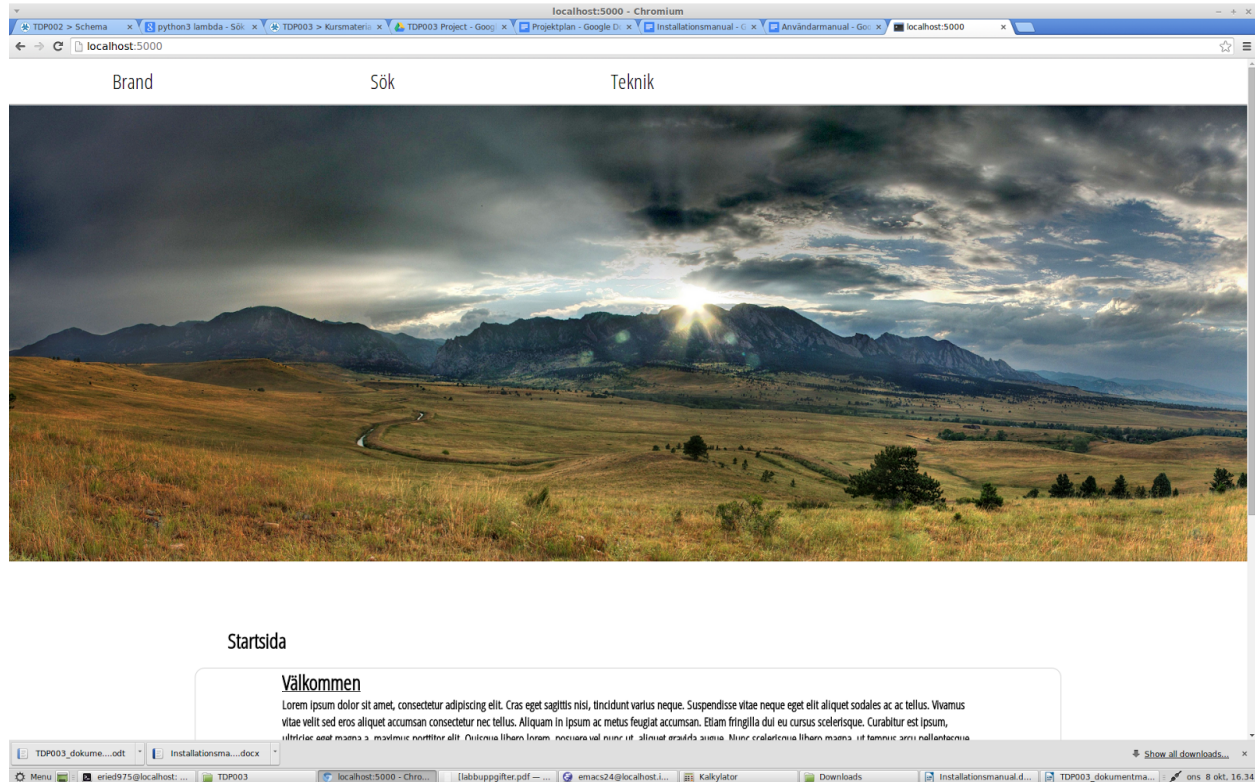
2.1. Mappstrukturer

Här är en liten överblick på hur mappstrukturen ser ut i portfolion:

```
MyPortfolio/  
  static/  
    images/  
      *.jpg *.png *.gif  
    style/  
      main.css  
  templates/  
    *.html  
  data.json  
  data.py  
  myFlaskProject.py
```

2.2. Hur man startar och stänger av servern

För att kunna komma in på portfolion och använda den behöver man starta själva servern som håller sidan igång. För att starta servern behöver du först gå till mappen "MyPortfolio" i ditt terminalfönster. I den mappen finns en fil som heter "myFlaskProject". Det är denna fil som man använder för att starta själva servern. Skriv följande kommando i din terminal för att starta servern: `python3 myFlaskProject.py`. Nu är portfolion igång! För att se om det fungerar kan du öppna webbläseren och skriva in följande url "<http://localhost:5000/>". Denna sida bör nu dyka upp:



Figur 1

För att sedan stänga av servern räcker det med att trycka på dessa två knappar samtidigt: "`<CTRL> och C`". Då ska du komma tillbaka till ditt terminalfönster som väntar på en vanlig inmatning.

2.3. Lägg till nya projekt och redigera gamla i portfolion

Som det ser ut nu finns det bara några exempelprojekt i portfolion för att enkelt kunna demonstrera hur portfolion ser ut och hur den fungerar. Filen som används för att redigera och lägga till nya program heter "data.json" och den finner du under mappen MyPortfolio. Om du först öppnar filen i ett vanligt textredigeringsprogram så ser du en kanske lite märklig struktur i filen. Denna struktur krävs för att datalagret ska kunna läsa in filen korrekt.

Strukturen ser ut som så att all projektinformation du kan hitta om ett projekt börjar efter ett tecken som ser ut såhär "{", informationen om just det projektet slutar på "}". Allting däremellan är information om själva projektet t.ex. Projektnamn, kursnamn etc. Alla dessa projekt som du ser separeras med ett "," tecken. Alla projekt ligger därefter i en lista, listan börjar längst upp på sidan med ett "[" tecken och slutar längst ner på sidan med ett "]" tecken.

2.3.1. Redigera gamla projekt

Eftersom det redan finns ett exempelprojekt i portfolion kan du till en början redigera informationen som du behagar. Det du dock inte bör göra är att ändra eller lägga till nya rubriker i projektet. Det är rubriker som krävs för att datalagret ska kunna behandla projekt informationen. Däremot är det fritt fram att ändra på all information i spalt 2 som visas på figur 2 t.ex. "short_description" : "Vanlig text" kan man ändra till "short_description" : "Min nya vanliga text".

2.3.2. Lägga till nya projekt

För att sedan lägga till nya projekt räcker det med att kopiera ett projekt som redan finns och sedan klistra in det innanför tecknerna "[" och "]" och direkt efter ett annat projekt. OBS! Var noga med att det finns ett "," tecken mellan alla projekt. Det visas i figur 2 vart alla dessa kommatecken ska vara. Viktigt att notera när du lägger till projekt är vilka fält som måste finnas enligt punkt 2.2.3.

Figur 2

2.3.3 Obligatoriska fält

Det här är en lista över de fält som måste finnas med i ett projekt för att databasen ska kunna hitta informationen projektet.

- 'project_no' - Projektet måste ha ett ID
- 'short_description' - Kort information om projektet, det här är det som visas när man söker efter projekt och är det som användaren bestämmer om hen vill läsa just det projektet.
- 'course_id' - Kursnamn.

```

{   Spalt 1   Spalt 2
{
  "start_date": "2009-09-05",
  "short_description": "no",
  "course_name": "OK\u00c4NT",
  "long_description": "no no no", < Dessa kommatecken separerar
  "group_size": 2,                projekt innehåll
  "academic_credits": "WUT?",
  "lulz_had": "many",
  "external_link": "YY",
  "small_image": "nature.jpg",
  "techniques_used": [
    "python"
  ],
  "project_name": "python data-module test script",
  "course_id": "TDP003",
  "end_date": "2009-09-06",
  "project_no": 1,
  "big_image": "nature-eye.jpg"
}, < Dessa kommatecken separerar projekt
{
  "start_date": "2009-09-07",
  "short_description": "no",
  "course_name": "OK\u00c4NT",
  "long_description": "no no no",
  "group_size": 4,
  "academic_credits": "WUT?",
  "lulz_had": "few",
  "external_link": "YY",
  "small_image": "nature-face.jpg",
  "techniques_used": [
    "c++",
    "csv",
    "python"
  ],
  "project_name": "NEJ",
  "course_id": "TDP003",
  "end_date": "2009-09-08",
  "project_no": 3,
  "big_image": "nature.jpg"
},

```

- 'lulz_had' - Hur kul var projektet? Sifferskala eller bara text, båda funkar bra här.
- 'small_image' - En liten bild som visas tillsammans med short_description vid sökresultat som ska visa användaren tillsammans med short_description om projektet är av intresse för hen.
- 'start_date', 'end_date' - Start och Slutdatum är viktigt för att en läsare ska kunna avgöra ett projekts relevans.
- 'long_description' - Projektets huvudsakliga text, kroppen.

Kom ihåg: Ju fler punkter man lägger till ett projekt utöver minimum, desto bättre och mer relevant blir det. T.ex. är ju 'big_image' väldigt bra, annars blir det en tom ruta istället, det är dock inget krav för att portfolion ska fungera.

2.3. Felsökning av portfolion.

Logfiler till portfolion sparas under mappen "MyPortfolio" i en fil som heter datalog.log. I denna fil loggas Datum och tid, anropsnummer samt kort information om vad som anropats. T.ex hur många projekt som returnerats.

2.3.1 Fel som loggas

När det blir något fel i portfolion loggas det ned i logfilen med väldigt bra förklaring av vart felet inträffat. Detta är felen som loggas ned (dom är väldigt exakta vilket betyder):

1. WARNING Expecting ',' delimiter: line 10 column 5 (char 218) in database file - Detta betyder att det måste finnas ett ',' tecken på rad 10 kolumn 5 i databas filen.
2. WARNING Invalid control character at: line 6 column 28 (char 129) in database file - Detta betyder att ett ogiltigt tecken finns på rad 6 kolumn 28. Det kan vara t.ex. ett <ENTER> på den platsen.
3. WARNING Database file was not found! - Detta betyder helt enkelt att databas filen(data.json) inte kunde hittas.

2.3.2 Vanliga fel

1. "TypeError: 'NoneType' object is not iterable" - Detta fel kan uppstå av en rad olika orsaker men i sin helhet betyder det att strukturen är fel i filen "data.json". Dessa är några av de vanligaste felen som kan vara i filen.
 - Det absolut vanligaste felet som kan uppstå är "TypeError: 'NoneType' object is not iterable". Detta fel brukar uppstå om man använt sig utav knappen <Enter> i sin projekt information. Av naturliga skäl under rubriken long_description då det är där man har sin längre text. Men hur gör man då för att kunna bryta raden?? Jo, detta kan man göra genom att skriva "\\n" som betyder <Enter>. Efter att du rättat till felet kan du se om det fortfarande är fel på koden. Är det fortfarande fel kan du titta på nästa punkt om det hjälper.

- Ett annat vanligt fel är att man inte lagt till ett “,” tecken mellan alla projekt och rubriker. För att rätta till felet behöver man bara se till att det finns ett komma mellan alla projekt och rubriker. Det visas i figur 2 vart dessa kommatecken ska vara.
 - Se till att all information under rubrikerna “group_size” och “projekt_no” har citationstecken(“) runt sig. T.ex. Strukturen visas i figur *. Som du ser i bilden är det två stycken rubriker vars information inte har citationstecken runt sig.
2. “Keyerror: ‘rubrik’” (där ‘rubrik’ kan vara t.ex ‘start_date’) är ett annat fel som kan uppstå om man t.ex. råkat ändra på någon av alla rubrikerna i något/några av projekten. Står det t.ex. “Keyerror: ‘start_date’” vet vi att rubriken ‘start_date’ har i något av alla projekten manipulerats till ett fel namn. Kolla igenom filen “data.json” efter rubriken som är fel och ändra den till sin rätta rubrik.