

Innehållsförteckning

Användarmanual.....	2
Installation.....	2
Startsida.....	2
Lista av projekt / Söksida.....	2
Tekniker.....	3
Loggning.....	3
Hantering av projekt.....	3
Systemdokumentation.....	4
Datalager:.....	5
load(filename).....	5
get_project_count(db).....	5
get_project(db, id).....	5
get_techniques(db).....	5
get_technique_stats(db).....	5
Presentationslager.....	6
Systemtest.....	6

Användarmanual

Installation

Portfolion behöver ha Python 3.3.x och Flask installerat för att fungera. Python finns att hämta på deras officiella hemsida och Flask kan installeras med hjälp av pip. Om flera versioner av Python är installerat behöver man installera pip för Python 3:

```
$ sudo apt-get install python3-pip  
$ sudo pip-3.3 install flask
```

För att installera portfolion krävs även att Git är installerat. Portfolion installeras med kommandot:

```
$ git clone https://github.com/karkn128/TDP003.git
```

Startsida

Startsidan är uppbyggd så att man ska få en inblick på vad som är aktuellt. Tre projekt med deras namn och korta beskrivning listas upp i den vänstra spalten. I den högra står det lite kort information om oss som har skapat systemet.

Längst upp på alla sidor finns det tre navigationslänkar, en som hänvisar till startsidan, en för projektlistan och en för tekniker som har används bland alla projekt.

Lista av projekt / Söksida

Den här sidan används för att söka efter specifika projekt. Förklaring ges i bilderna nedan.

[Start Lista med projekt](#) [Tekniker](#)

Search fields:

☒ Projekt ID ☒ Projektnamn ☒ Course name ☒ Course ID ☒ Description

Required techniques:

☐ ada ☐ c++ ☐ csv ☐ python

Sort order:

☒ Ascending ☐ Descending

Sort by:

Sökfält. Att ett fält är bockat innebär att sökningen sker i det fältet, om inte så ignoreras det. Alla fält är ibockade som standard.

Search fields:
☒ Projekt ID ☒ Projektnamn ☒ Course name ☒ Course ID ☒ Description

Required techniques:
☐ ada ☐ c++ ☐ csv ☐ python

Sort order:
☒ Ascending ☐ Descending

Sort by:

Här kan man filtrera sökningen så att endast de projekt med de ibockade teknikerna listas.

Search fields:
☒ Projekt ID ☒ Projektnamn ☒ Course name ☒ Course ID ☒ Description

Required techniques:
☐ ada ☐ c++ ☐ csv ☐ python

Sort order:
☒ Ascending ☐ Descending

Sort by:

Här bestämmer man hur resultatet ska listas.

Önskas en lista av alla projekt i systemet så ska man göra en tom sökning.

Tekniker

Här listas alla olika tekniker som har använts av alla projekt. Under varje teknik så listas alla projekt som har använt just den tekniken.

Loggning

Allt som görs på sidan loggas i en separat fil som har namnet "serverlog.log". Denna fil är lokaliserad i mappen "web".

Hantering av projekt

Om man vill skapa, redigera eller radera ett projekt så behöver man öppna "data.json"-filen.

För att skapa ett nytt projekt så ska samma datamall användas som de andra projekten i filen.

Redigering görs genom att ändra på värdena i högerledet.

Vill man radera ett viss projekt så ska hela projektblocket tas bort.

För information om hur formatet för ett projekt i json-filen ser ut; se *systemdokumentation*.

Systemdokumentation

All data som portfolion hanterar är sparad som JSON-format i filen data.json.

Projekt sparas som på formatet nedan.

```
{  
  "start_date": "2009-09-05",  
  "short_description": "no",  
  "course_name": "OK\u00c4NT",  
  "long_description": "no no no",  
  "group_size": 2,  
  "academic_credits": "WUT?",  
  "lulz_had": "many",  
  "external_link": "YY",  
  "small_image": "X",  
  "techniques_used": [  
    "python"  
  ],  
  "project_name": "python data-module test script",  
  "course_id": "TDP003",  
  "end_date": "2009-09-06",  
  "project_no": 1,  
  "big_image": "XXX"  
}
```

Datalager:

Datalagret finns i modulen data.py och är en samling funktioner för att hantera all data i data.json.

load(filename)

Laddar in en JSON-formaterad fil och returnerar en lista med projekt. Vid fel returneras None. Alla funktioner i datalagret tar emot data på formatet som load returnerar.

get_project_count(db)

Returnerar hur många projekt som finns i db.

get_project(db, id)

Returnerar det första projektet som hittas med id. Om projektet inte finns returneras None.

search(db, sort_by='start_date', sort_order='desc', techniques=None, search=None, search_fields=None)

Söker efter projekt i datan som matchar alla angivna parametrar.

Parametrar:

- db (list) – lista som returnerad från load.
- sort_by (string) – Nyckeln på fältet som resultatet ska sorteras efter.
- sort_order (string) – 'asc' för stigande, 'desc' för fallande.
- techniques (list) – Lista med tekniker som projekt måste ha för att returneras.
- search (string) – Söksträngen, om None returneras alla projekt.
- search_fields (list) – Begränsar vilka fält som man letar efter söksträngen i, om listan är tom returneras inga resultat, om den är None så söker man i alla fält.

get_techniques(db)

Returnerar en lista med alla tekniker som finns i db.

get_technique_stats(db)

Alla tekniker i db sätts som nycklar för en dict. För varje teknik så sparas en lista med dicts

som håller alla projekts namn och id som använder sig av tekniken.

Exempel på hur datan som returneras kan se ut:

```
{  
'python': [{'id': 42, 'name': 'TDP003'}, {'id': 17, 'name': 'TDP002'}],  
'c++': [{'id': 3, 'name': 'TDP005'}]  
}
```

Presentationslager

Presentationslagret finns i filen server.py och är portfolions huvudprogram. Det är server.py som körs för att starta hemsidan. Hemsidan hittar man på <http://localhost:5000> när server.py är igång.

De olika sidorna på hemsidan genereras med funktionerna som finns på sidans @app.route(), till exempel listsidan genereras med funktionen som finns under @app.route('/list'). Eftersom att datan i "data.json" ska kunna uppdateras utan serveromstart så kallas data.load() i början av varje funktion.

Loggning sker genom att allt Flask i vanliga fall skriver till terminalen istället skrivs till serverlog.log. Detta sker med hjälp av Pythons loggingsmodul.

Systemtest

Testning av datalagret är automatiserat och sker med hjälp av data_test.py modulen. Presentationslagret har testats helt manuellt. Alla HTML-sidor har verifierats med hjälp av <http://validator.w3.org/>.

En känd bugg är att sökfunktionen endast ger resultat för söksträngar som matchar ett helt fält. Att söka efter "python" i "python test" ger alltså inget resultat.