# Word-eksport

## Systembeskrivelse

Formålet med systemet er å ta inn en JSON-fil fra MIME, for så å eksportere et Word-dokument som representerer informasjonen, basert på en Word-mal.

## Rammeverk/Moduler:

### TkInter

For Brukergrensesnittet er GUI-pakken [TkInter](https://wiki.python.org/moin/TkInter) brukt.

### python-docx

[python-docx](https://python-docx.readthedocs.io/en/latest/) generer/endrer på Word-filer. Denne modulen brukes til å lese inn dokumentet som er generert med python-doxc-template (2.3), for å sette inn hyperlinks, og sideskift.

### python-docx-template

[python-docx-template](http://docxtpl.readthedocs.io/en/latest/) brukes til sette data fra JSON-filen inn i en Word-mal, som er skrvet med [Jinja2](http://jinja.pocoo.org/docs/dev/). python-dox-template har noen begrensninger, som gjør at vi i tillgg måtte bruke python-docx for å sette inn hyperlinks og sideskift.

### Py2exe

[Py2exe](http://py2exe.org/) pakker Python-filene i en mappe med en kjørbar exe-fil. Programmet inneholder filen setup.py, som spesifiserer hva Py2exe skal ta med når exe-filen pakkes. Siden man ikke får kjørt exe-filer på ehelse-PCene, er ikke dette noe som brukes.

## Oppsett/installasjon

Dette er beskrevet i brukermanualen.

## Eksportere til exe-fil

Prosjektmappen inneholder filen setup.py som inneholder en liste over alle pakker som brukes av programmet. Dersom man ønsker å bruke nye pakker, må disse spesifiseres i package-listen før man pakker programmet.

For å pakke filene, start kommandolinjen og naviger til mappen setup.py ligger i. Skriv kommandoen python setup.py py2exe. Programmet blir dat pakket, og to nye mapper: “build” og “dist”. Den kjørbare exe-filen ligger “dist”-mappen, og det er denne mappen du må overføre til en annen PC for å kjøre programmet.

## Arkitektur

Programmet er basert på et MVC-mønster med de følgende klassene:  


*Klassediagram Word-eksport*

* **Program**: Program initsialiserer TkInter, og sender dette inn i MainController. Program har også mainloop-en for GUI.
* **Messages**: Messages inneholder feil- og suksessmeldinger som brukes av MainController.
* **GuiContent**: GuiContent inneholder tekst og innstillinger for for GUI, som for eksempel vindusstørrelse, vindustittel og tekst på knapper.
* **TemplateElements**: TemplateElements inneholder egendefinerte malelementer som settes inn av python-docx. De to elementene som finnes er:
  + NEW\_PAGE
  + URL
* **MainView**: MainView inneholder selve GUI-elementene, i tillegg til noen hjelpemetoder for å sette teksten i feil- og suksessmeldinger pluss filstifeltene.   
    
  Til å bygge GUI, er TkInter brukt. Denne modulen er ikke den letteste å jobbe med mtp. designmuligheter ol. For å plassere GUI-elementene, er det brukt en kombinasjon av TkInters pack(), som bare plasserer elementene under hverandre fra toppen og nedover, og grid(), som gjør det mulig å spesifisere rad og kolonne for elementene.
* **InputHandler**: InputHandler henter innholdet fra JSON-filen, og inneholder metoder for å hente ut data som, temaer, dokumenter, målgrupper for et spesifisert dokument, osv. All data som skal brukes hentes vha. metoder i InputHandler.
* **ExportContent**: ExportContent brukes til å konstruere en liste som kan settes inn i jinja2-malen for Word-dokumentet, ved å hente data fra JSON-filen, vha. InputHandler.
  + add\_topic(topic, documents, input\_handler, target\_group\_id):  
    Denne metoden brukes til å sette inn et tema, med dets tilhørende dokumenter. En instans av InputHandler sendes også inn her, for å kunne hente ut informasjon som navnet på en status som eksisterer i et dokument. target\_group\_id settes til iden til en målgruppe som velges i nedtrekkslisten i GUI, for å kunne generere et Word-dokument som inneholder alle standarder, profiler og støttedokumenter som er relevante for denne målgruppen. Dersom man ønsker å eksportere alle standarder, profiler og støttedokumenter fra JSON-filen, setter man target\_group\_id = 0.
* **WordHandler**: WordHandler brukes til å opprette Word-dokumenter ved bruk av modulen python-docx. Ved initialisering kan man spesifisere template\_path. Dersom man sender inn filstien til et Word-dokument, vil man isteden for å opprette et nytt Word-dokument, lese inn det eksisterende Word-dokumentet.   
    
  WordHandler var i utgangspunktet tenkt til å være klassen som stod for alt som har med skriving til Word-filen å gjøre, men denne funksjonaliteten gjøres nå direkte i MainController vha. modulen python-docx-template. Det WordHandler brukes til nå er å sette inn urler og sideskift etter hver standard.
* **MainController**: MainController er klassen hvor det meste av logikken i programmet ligger. MainController initialiserer MainView, og inneholder metoder som kjøres ved knappetrykk i GUI.
  + upload(): Her hentes filstien til JSON-filen fra dialogboks, og en instans av InputHandler opprettes med den innlastede JSON-filen.
  + upload\_template(): Her hentes filstien til Word-malen fra dialogboks.
  + download(): Her hentes filstien til der brukeren ønsker å lagre Word-dokumentet. Videre blir verdiene fra ExportContent satt inn i Word-malen (som er skrevet med bruk av jinja2) og lagres på harddisken.

Word-filen lastes så inn igjen i programmet (med innholdet fra JSON), for å sette inn lenker og sideskift vha. python-docx, da dette ikke støttes i python-docx-template. Det er opprettet et eget format for å sette inn elementer når man iterere over dokumentet.

* [[url|text|<http://url.com>]]: setter inn en hyperlink med teksten “text”, og urlen <http://url.com>.
* [[newpage]]: setter inn sideskift.