aestasGIS

*Nysted d.15. september 2022*

**Forslag til udvidelser og forbedringer af QGIS DMP Manager – ET QGIS baseret plugin til håndtering og redigering af data fra Miljøportalen**

# Indledning

QGIS DMP Manager er et plugin, som giver brugeren mulighed for at downloade data fra Miljøportalen, redigere disse lokalt og slutteligt uploade de redigerede data til Miljøportalen.

Plugin-et er udviklet over 2-årig periode og kan nu foretage de basale funktioner (”Really need to have”) i forhold til Miljøportalen.

Selve udviklingsforløbet blev meget længere og meget mere kompliceret end først antaget. Derfor mangler plugin-et et sæt funktioner, som var planlagt - men som ikke kom med i 1. udviklingsperiode da ressourcerne blev brugt til implementering af helt basalt nødvendige funktioner.

Dette notat beskriver de forskellige manglende funktioner inklusive tidsestimater. Forslag kommer ikke i en prioriteret orden. Hashtags (# - tegn) refererer til issue nummer i GitHub.

**Alle forslag som er overstreget, er gennemført**

# Mulighed for at bruge MS-SQLServer som data repository #23

QGIS Core support for MS-SQLServer er ikke nær så omfattende som PostgreSQL. En enkelt facilitet - at kunne udforme et vilkårligt komplekst SQL-udtryk med eksempelvis joins og lade resultatet vises som et lag i QGIS kan ikke foretages på samme måde som i PostgreSQL og GeoPackage.

Denne funktion er et absolut ”must” for at kunne bruge MS-SQLServer som lokalt repository for data download’et fra DMP. Muligheden for at benyttet MS-SQLServer som repository er derfor fjernet fra listen over mulige datakilder, som pt. nu er begrænset til PostgreSQL og GeoPackage.

Den bedste metode for at muliggøre brugen af MS-SQLServer som data-repository er at få udviklet den manglende funktionalitet direkte i QGIS.

Jeg har henvendt mig til den ansvarlige QGIS Core developer for dette område, Alessandro Pasotti, og fået et omkostnings-estimat for udviklingsarbejdet.

Estimat: 2.000,- Euro. Selve genetableringen af MS-SQLServer i plugin bagefter: uden beregning.

# ~~Mulighed for masseupload af data #22~~

~~Pt. kan man kun uploade en modificeret (oprettet / rettet /slettet) feature af gang vha. højreklik på den enkelte feature fundet i difference funktion.  
  
Det ville være ønskeligt med en funktion som enten uploadede alle modifikationer fundet vha. difference funktionen i én, samlet aktion. Eller man kunne multi-markere (i stedet for enkelt markere) features i træ strukturen for differencer og sende samtlige markerede features til upload som én aktion.~~

~~Estimat: ca. 10 timer inkl. opdatering af vejledning.~~

# ~~Mulighed for at skifte mappe for .qml filer til automatisk tematisering for data hentet fra DMP #21~~

~~I øjeblikket findes der en fast mappe:~~ **~~%appdata%\QGIS\QGIS3\profiles\default\python\plugins\dmp\_manager\templates~~** ~~(dvs. en undermappe "templates" til plugin installationsmappen) som indeholder .qml filer der benyttes til automatisk tematisering af downloadede temaer fra DMP.~~

~~Det ville være formålstjenligt at denne mappeplacering dels var synlig under administrations-fanebladet og dels kunne ændres ved tryk på en knap, som viser en mappe selektor.~~

~~Use-Case: Flere brugere benytter plugin-et i organisationen, og man ønsker at benytte den samme tematisering for alle brugere. I dette tilfælde kunne man placere mappe med .qml filer på en fælles netværks baseret mappe~~

~~Estimeret tidsforbrug: Ca. 5 timer inkl. redigering af vejledning.~~

# ~~Knap til opdatering af liste med database-forbindelser #12~~

~~Hvis man~~ *~~først~~* ~~starter DMP Manager og~~ *~~dernæst~~* ~~laver en ny forbindelse til en database eller database-fil, vil den nye forbindelse ikke blive vist i DMP Manager's liste over mulige database forbindelser (Faneblad "Administration, combo-boks "Database"). Man bliver nødt til at genstarte DMP Manager eller evt. QGIS for at den nye forbindelse kommer med på listen.~~

~~En løsning ville være at lave en knap til højre for combo-boksen, som kunne aktivere en opdatering af listen uden genstart af DMP Manager/QGIS~~

~~Estimeret tidsforbrug 3 timer inkl. udvidelse af brugervejledning.~~

# ~~Funktion til at hindre brugerfejl ved skift af DMP miljø #7~~

~~Ved skift fra demo-miljø til produktionsmiljø opstår der en risiko for at~~ *~~brugeren~~* ~~laver en fejl:~~

1. ~~Plugin er fra starten sat op til demo-miljø~~
2. ~~Bruger downloader et datasæt og laver modifikationer i redigeringslaget.~~
3. ~~Plugin sættes op til at køre i produktionsmiljø~~
4. ~~Bruger uploader modifikationer lavet i (demo) redigeringslaget til Miljøportalens produktionsmiljø.~~
5. ~~Modifikationerne slår nu igennem i produktionsmiljøet, fordi data i demo og produktion basalt set er ens.~~

~~Risikoen for ovenstående brugerfejl kan fjernes ved at plugin ved download af data navngiver redigerings- og referencelag på en måde der tydelig markerer fra hvilket miljø data stammer fra. Og at upload funktionen i plugin tilføjes en kontrolfunktion, som før upload kontrollerer, at DMP miljøet, der uploades til, er det samme som data i redigeringslaget kommer fra. Hvis de er forskellige, stoppes upload og der vises en fejlmeddelelse.~~

~~Estimeret tidsforbrug for udvikling: 8 timer inkl. udvidelse af brugervejledning~~

# ~~Funktion til automatisk at gemme tematisering #4~~

~~Den nuværende metode til at gemme tematisering for de forskellige typer af datalag er helt manuel og temmelig kryptisk for uerfarne brugere af QGIS. Jeg foreslår, at man udvider administrations-fanebladet med en drop-down liste, som indeholder navne på alle hentede lag fra DMP. Endvidere udvides med en knap som ved aktivering automatisk læser symbolisering/tematisering af det valgte lag i drop-down listen, konverterer dette til en .qml fil og automatisk gemmer denne fil den korrekte mappe at brugeren skal gøre dette manuelt~~

~~Estimeret tidsforbrug for udvikling: 8 timer inkl. udvidelse af brugervejledning~~

# ~~Automatisk navngivning og typesætning af Primary Key baseret på typen af database #3~~

~~Når man pt. skifter databasetype skal man manuelt sætte navn på primary key og om denne er af type tekst (Felt ”Primary key quote” sættes til apostrof). Det er ikke optimalt, da det giver mulighed for fejl og i øvrigt kan gøres automatisk baseret på den valgte databasetype.~~

~~Estimeret tidsforbrug for udvikling: 3 timer inkl. rettelse af brugervejledning~~

# ~~Hurtigt skift mellem Demo og Produktions miljø #2~~

~~Ved opstart at QGIS kan man ikke se, om klienten er sat op til at køre mod "Demo" eller "Produktion". Der er heller ikke nemt at skifte fra det ene til det andet miljø, fordi det omfatter en (kryptisk) redigering i en tekstfil samt genstart af QGIS plugin'et.~~

~~Jeg vil foreslå, at man laver en tilføjelse til systemet, således at bruger dialogen~~ **~~tydeligt~~** ~~viser om plugin kører mod Produktions eller Demo miljøet. Og at der er en funktion, som kan skifte miljø uden at skulle redigere tekstfiler og genstart af plugin.~~

~~Samme funktion kan også laves, så den "låser" til et bestemt miljø. Dvs. den samme situation som nu, blot med den føromtalte tydelige tilkendegivelse af hvilket miljø man arbejder i.~~

~~Estimeret tidsforbrug for udvikling: 9 timer inkl. rettelse af brugervejledning.~~

# ~~Funktion til opsplitning af multipolygoner/multilinjestykker til et sæt at simple polygoner/linjestykker #26~~

~~Når man downloader eksempelvis et sæt af polygoner fra DMP; dernæst retter en eller anden tilfældig attribut fra en feature, risikerer man at featuren ikke kan skrives tilbage til DMP, fordi polygonen fra starten af er en multipolygon, og DMP accepterer ikke multipolygoner.  
(Dvs. at DMP indeholder data, som DMP ville forkaste ved kontrol af data) - ikke pga. evt. rettelser, men fordi den originale geometri er ulovlig.~~

~~Jeg foreslår at lave et værktøj, som kan gennemløbe data og automatisk splitte multi-objekter til simple objekter - men med styrings af div. attribut værdier, således af DMP vil kunne acceptere alle de resulterende features.  
  
Man kan sørge for, at den arealmæssigt største simple resultat-polygon genbruges som den "rettede" udgave af den originale feature og alle andre simple resultat polygoner for tildelt behandlede attribut- værdier baseret på de originale data, men hvor visse attributter, f.eks. "navn" tilføjes et " - 2", "- 3" osv. for at skelne de enkelte nye features fra hinanden.~~

~~Jeg kan pt. ikke overskue, om allerede eksisterende splitte-mekanismer i QGIS kan genbruges til ovenstående formål.  
  
Estimeret tidsforbrug: 20 timer inkl. rettelse af brugervejledning~~

# Udskiftning af c# .NET authentication module til en python funktion. #27

Den nuværende løsning med brug af et .NET C# modul til authentication er temmelig klodset og skrøbelig.

Den bør erstattes af et ”native” Python 3.x modul, således at man kan fjerne afhængigheden af .NET.

Dette vil gøre plugin mere robust. Man kan endvidere også få det indlagt i det officielle QGIS plugin repository, således plugin-et kan downloades ved den sædvanlige procedure for plugion installation

For at få den bedst mulige løsning kræves en mindre rettelse i selve QGIS programmet, således kravet om "token secret" fjernes fra QGIS's Auth2 authentication module.

Estimeret tidsforbrug: 25 timer (Jeg er temmelig usikker på dette estimat)

# ~~Hent det nye objekt ved overførsel af rettelse til DMP #1~~

~~Når man skriver et rettet (ikke insert eller delete) objekt til DMP, får man fra DMP en udgave af den rettede polygon retur som kontrol. Systemet bruger retur-objektet til at opdatere de 2 datasæt - redigeringslaget og referencelaget - således disse nu reflekterer den nye data-situation i DMP.~~

~~Men problemet er, at DMP ved rettelse "pensionerer" det eksisterende objekt og opretter et nyt objekt i deres system med samme objekt-id men med et nyt versions-id. Det nye objekt indeholder de rettelser som kommer fra QGIS klienten.~~

~~Det gamle objekt bliver "pensioneret", dvs. får blot tilføjet en "gyldig til" tid. Og dette pensionerede objekt sendes tilbage som svar til QGIS. Så QGIS indeholder ikke det objekt, som nu er den nye udgave af objektet i DMP's system.~~

~~Pt. kan man omgå dette problem ved at "opfriske" hele laget og downloade det fra DMP på ny. Det er som at skyde en spurv med en kanon.~~

~~Jeg vil foreslår at lave en tilføjelse til QGIS klienten, så den ved modifikation henter det nye objekt og indlæser dette i de to datalag i stedet for det pensionerede objekt. Så får man den rette situation, hvor QGIS lagene indeholder præcis det samme som DMP's datasæt.~~

~~Estimat 8 timer.~~