

Her er litt drodning rundt persistente uuids for bilder, håper ikke jeg har oversett noe viktig!

Fire deloppgaver:

1. I Oracle (i `usd_felles`), opprette uuids for eksisterende bilder og legge til uuid på innkommende bilder.

Tidsestimat: Ferdig i forbindelse med ordimage-migreringen. (Foreløpig kun deployet på MUSUTV) .

1. Få uuid-ene ut til musit-public (koden i `music-public-populate`, hente uuid fra oracle i stedet for å generere nye on-the-fly).

(Unimusportalen gir pr idag kun tilgang til publiserte bilder) Tidsestimat: Koden er ferdig, men utestet (`branch persistente_ids`)

1. Gi verden et endepunkt for tilgang til disse bilder via uuid:

Tidsestimat: Se nedenfor

1. Noen snags vedrørende naturhistorie og deres uuid-er?

Tidsestimat: ukjent

Spørsmålet er hvor dette endepunktet skal implementeres.

Utad tror jeg det bør ligge under: <https://www.unimus.no/api/media.../> med uuid og en måte å spesifisere ønsket format og/eller versjon på. (Her finnes jo også en standard man på sikt kan tenke seg å støtte deler av, for å hente ut et utsnitt av et bilde etc) (Og sende med token for tilgang til ikke-publiserte bilder)

I så fall bør dette endepunktet ligge i `musit-public-api`, men spørsmålet er uansett hvordan dette endepunktet skal implementeres.

Et par alternativer:

1. La `musit-public-api` få tilgang til Oracle og kunne hente bilder her direkte.
2. La dagens backend få tilgang til Oracle og la `musit-public-api` kalle et endepunkt her.
3. Ny backend for å snakke med Oracle. (som `musit-public-api` kaller)

4. La `web_hent_bilde.php` fortsette å ha eneansvar for å hente ut bilder.
(som `musit-public-api` kaller)

Diskusjon:

Alternativ 1) (La `musit-public-api` få tilgang til Oracle)

Åpenbare fordeler:

- Slipper ny deployment-enhet

Åpenbare ulemper:

- Ingen spesielle?

Tidsestimat: Tilgang til publiserte bilder (og evt upubliserte uten autorisering):
1-2 uker Tilgang til publiserte bilder og upubliserte (sistnevnte med autorisering): 2-3 uker

Alternativ 2) (Legge hent-bilde inn i dagens backend) Åpenbare fordeler:

- Slipper ny deployment-enhet
- Får autorisering mot ikke-publiserte bilder nesten gratis

Åpenbare ulemper:

- Vanskelig/umulig å teste i dev pga docker-problematikk
- Kanskje dumt å blande endepunkter ment for offentligheten inn i backend ment for "interne" brukere, i tilfelle de offentlige endepunktene skulle bli superpopulære.

Tidsestimat: 1-3 uker

Alternativ 3) (Ny "backend" for Oracle-spesifikke endepunkt) Åpenbare fordeler:

- Trolig lettere å teste i dev enn alt 2)
- Slipper å blande offentlige endepunkter med "interne".

Åpenbare ulemper:

- Ny deployment enhet
- Mer jobb med auth, men kanskje ikke så mye?

Tidsestimat: 2 -3 uker (Involverer Jarle S og Svein (Svein for deployment))
Alternativ 4) (Surfe videre på web_hent_bilde en stund til) Da må php-skriptet utvides og det må lages litt mer kode i Oracle for å hente via uuid. Åpenbare fordeler:

- Slipper ny deployment-enhet

Åpenbare ulemper:

- web_hent_bilde har gjort jobben litt for lenge, bør etterhvert få mest mulig over på enhetlig plattform (nodejs og openshift).
- ikke opplagt hvordan man skal lage autorisering for tilgang til ikke-publiserte bilder

Tidsestimat: (Involverer Jarle E. og Jarle S.) Hvis vi ignorerer autorisering: Et par dager
Med autorisering for ikke-publiserte bilder: Vanskelig å si, kommer an på hvordan dette skal gjøres, men kan raskt bli 2-3 uker.