

3. Historikk i matrikkelen

Historikk i matrikkelen

I "lov om eigedsregistrering" (matrikkelloven) og dennes forskrifter beskrives krav og bestemmelser rundt føring og lagring av data i matrikkelen. I tillegg spesifiserer arkivloven en del krav rundt lagring av føringer og resultater av dette. For å tilfredsstille kravene og ønskene i lovverket har det i matrikkelen blitt implementert en arkivarisk historikk der alle føringer tas vare på i systemet. Dette kan da brukes for å se tidligere situasjoner. I dagens system benyttes dette i hovedsak i feilrettinger og ved klager på føring.

Historikken i matrikkelen skal tilfredsstille geodatalovens forskrifter når det gjelder implementasjon av INSPIRE direktivet. Av dette følger en del feltnavn som blir beskrevet her.

Innføring av historikk i matrikkelen og det nye matrikkelAPI-et har noen konsekvenser for modellen og tjenestene man benytter seg av. Dette dokumentet beskriver hvordan vi kan se historikk og hvordan man henter historiske data i tjenestene.

Historikk i modellen

Hver endring av et objekt i matrikkelen fører til at systemet tar vare på en ny versjon av objektet. På denne måten får vi en tidslinje av versjoner der det er mulig å ta steg bakover og se forskjeller fra versjon til versjon. Hver enkel versjon er komplett og inneholder alle data slik de var i tidsrommet versjonen var gyldig. For å synliggjøre dette og gjøre det mulig å verifisere dette er det lagt til nye felter i systemet. Disse feltene er:

- oppdateringsdato
- oppdatertAv
- sluttdato
- avsluttetAv
- versjonId

Disse feltene beskriver en versjon av et objekt og er realiserte på både bobler og komponenter i matrikkelen. Ikke alle bobler er av typer som har historikk, men de fleste domeneboblene har dette.

Feltene `oppdateringsdato` og `oppdatertAv` viser når boblen sist ble endret, og hvilken matrikkelbruker som utførte endringen. Disse feltene og `versjonId` er alltid satt på en boble som har disse feltene. `VersjonId` er et felt som teller opp hver gang man endrer et objekt. Man kan

dermed se på en boble eller en komponent hvor mange ganger brukere har gjort noe med dem.

Feltene `sluttdato` og `avsluttetAv` er kun satt til ekte verdier på versjoner av objekter som er historiske. Dette kan være fordi noen har oppdatert objektet og vi har fått en ny versjon, eller det kan være fordi noen har slettet objektet. Hvis det er snakk om en oppdatering vil man ha en ny versjon som har `oppdateringsdato` lik denne versjonens `sluttdato` og `versjonId` lik dennes `versjonId + 1`.

En versjons levetid er definert som fra inklusivt `oppdateringsdato` til eksklusiv `sluttdato`.

sluttdato for sanntidsversjoner

Hvis et objekt ikke har noen `sluttdato` satt er datoen satt til "nåtid". Dette er en kunstig dato langt inn i framtiden som symboliserer at dette er et levende objekt. Dette er det samme som vi benytter i `matrikkelContext` som blir sendt inn i alle webtjenestekall for å vise at vi ønsker sanntidsdata.

Denne datoen er: 9999-01-01 00:00:00.00

Eksempel:

Vi har en adresse som kun har én versjon. Denne oppstod da adressen ble opprettet. Den har følgende data:

Vegadresse

id = 2

husnummer = 10

vegId = 1

oppdateringsdato = 01.01.2016 kl 00:00:00.00

oppdatertAv = bruker1

versjonId = 1

sluttdato = 01.01.9999 kl 00:00:00.00

avsluttetAv = **null**

Vi endrer så husnummer på adressen. I systemet vil det da eksistere to versjoner av objektet som ser slik ut:

Vegadresse

id = 2

husnummer = 10

vegId = 1

```
oppdateringsdato = 01.01.2016 kl 00:00:00.00  
oppdatertAv = bruker1  
versjonId = 1  
sluttdato = 12.01.2017 kl 11:35:21.000123  
avsluttetAv = bruker2
```

og

Vegadresse

```
id = 2  
husnummer = 999  
vegId = 1  
oppdateringsdato = 12.01.2017 kl 11:35:21.000123  
oppdatertAv = bruker2  
versjonId = 2  
sluttdato = 01.01.9999 kl 00:00:00.00  
avsluttetAv = null
```

Det er nå objektet med `versjonId` 2 som er den aktive versjonen av objektet, mens versjon 1 har blitt historisk.

Navigering og historikk

Det er kun objektene som blir endret som får en ny versjon ved endring, og ikke andre objekter. Det betyr at man ender opp i situasjoner der navigeringen av objektgrafen vil gi mange forskjellige oppdateringsdatoer på objekter som hører sammen. Man må da benytte seg av `oppdateringsdato` for én boble for å finne ut hvilken versjon av boblene den peker på vi skal ha tak i. For sanntidsnavigering blir dette enklere da vi alltid er interessert i den levende versjonen av bobler, men for historikk kan dette bli mer komplisert.

Det vil alltid kun være én versjon av en boble som gjelder for et gitt tidspunkt så hvis man har et startpunkt for navigering der man vet hvilket tidspunkt man vil se på vil systemet alltid gi ut de riktige versjonene av bobler som gjelder for det angitte tidspunktet.

Uthenting av historikk

Som skrevet i infoboksen over angir vi i `matrikkelContext`-parameteren (se "Tekniske detaljer" for mer informasjon om utfylling av denne) til alle tjenester på matrikkelAPI-et tidspunktet vi ønsker å se på data for. I de fleste tilfeller ønsker vi den levende versjonen og vi angir da følgende:

```
matrikkelContext
...
snapshotVersion = 01.01.9999 00:00:00.00
...
```

Dette viser for systemet at man ønsker sanntidsdata. Hvis man i stedet putter inn en historisk dato her vil man gjøre søk og hente ut historiske data. Dette gjelder for de fleste tjenestene, inkludert `StoreService` som brukes for å hente objekter.

- ❗ For å kunne hente ut historiske data kreves en spesiell rolle på brukeren. Denne rollen tildeles av Kartverket men det er foreløpig svært begrenset hvem som kan få denne rollen.

StoreService.getVersions(...)

For å se hvilke oppdateringsdatoer som eksisterer for en boble kan man benytte seg av metodene `getVersions` og `getVersionsForList` på `StoreService`. Disse metodene gir ut informasjon tidspunkt for alle oppdateringer gjort på en gitt boble innenfor et tidsrom.

Eksempel:

```
idsMedVersjon = storeService.getVersions(
    new Kommuneld(233),
    new Timestamp("01.01.2012"),
    new Timestamp("01.01.2017")
);

idsMedVersjon er da:
(
    ("27.10.2013T01:00:00.000+02:00", 233),
    ("16.02.2016T18:33:58.113+02:00", 233)
)
```

Vi ser at returen er en samling av nøkkel-verdi par der nøkkelen er tidspunktet kommunen ble oppdatert på og verdien er id-verdien.

Modellendringer

Over tid vil domenemodellen til matrikkelen endres, men for å sikre at de historiske versjonene fortsatt kan gjengis som de var på tidspunktet de ble opprettet er det lagt noen bånd på endringer som kan gjøres i modellen. Det er også innført et felt på alle bobler for å vise hvilke felt som inneholder data for denne versjonen av objektet.

Lovlige endringer

- Tillegg av felter
- Fjerning av felter (for framtidige data, ikke for gamle data)

Ulovlige endringer

- Endring av type for eksisterende felt
- Navneendring på felt

Modellering av endringer

Alle bobler har i API-et et eget felt, `metadata`, som inneholder hvilke feltnavn objektet har data i. Dette betyr at man kan se på dette for en enkelt versjon av en boble og se med en gang om det at et felt inneholder `null` er fordi dette er en gyldig verdi, eller fordi feltet ikke eksisterte på tidspunktet versjonen oppstod.

Det står over at fjerning av et felt er en lovlig endring, og det er sant hvis man ser på metadataene. Feltet vil ikke forsvinne fra modellen, men det vil ikke lengre være mulig å fylle data inn i. Sanntidsversjoner av objekter vil da ikke ha data i feltet, men historiske versjoner vil fortsatt kunne ha data der.
