

## 4. Tekniske detaljer

### MatrikkelContext

Alle tjenestekall mot API-et har en MatrikkelContext som siste parameter. Denne setter opp en del innstillinger for kallet og dataene som skal returneres. Blant disse er versjon av systemet klienten er laget for, identifikasjon av klienten, koordinatsystem man ønsker data returnert i og hvilket tidspunkt man ønsker data for. Under følger eksempel og litt mer detaljer for noen av feltene.

### Eksempel

```
MatrikkelContext context = new MatrikkelContext();
context.setLocale("no_NO_B"); //Norsk bokmål
context.setBrukOriginaleKoordinater(false);
context.setKoordinatsystemKodeId(new KoordinatsystemKodeId(10L)); //Denne bør hentes via kodeliste for ønsket koordinatsystem
context.setKlientIdentifikasjon("minKlient");
context.setSystemVersion("3.10"); //Det er denne versjonen jeg har hentet ut wsdl-er for og bygget min lokale kode for.
context.setSnapshotVersion(new Timestamp("9999-01-01T00:00:00+01:00")); //Siste versjon av objekter (trenger ikke lenger angi noe hvis det er det man vi ha)
```

### Klientidentifikasjon

En streng som identifiserer klienten. For noen kall er det påkrevd med en godkjent klient, men for de fleste innsynskall må denne bare være utfyllt med noe.

### SystemVersion

Hvilken versjon av matrikkelen man har laget klienten basert på. Ved å angi denne sørger API-et for at man ikke får ut felter og datainnhold man ikke kjenner til. API-et er bakoverkompatibelt og sørger for å mappe inn nye subtyper av objekter i kjente supertyper slik at man får ut alt man kan.

## SnapshotVersion

Angir om man ønsker å gå mot sanntidsversjonen av matrikkeldata eller en historisk versjon. For systemer som har historikk er det bestemt at sanntids-SnapshotVersion skal være 01-01-9999 klokken 00:00:00. Alle andre tidspunkt enn dette vil føre til at man prøver å hente ut historiske data, og dette krever en egen rolle for brukeren som skal gjøre kallet. Det er derfor viktig å passe på at man skriver tidspunktet som beskrevet i eksemplet over, eller ikke angir snapshotVersion i det hele tatt (angitt, men tom, er ikke lov).

Om man ser på selve requesten (med Fiddler) eller tester fra SoapUI skal altså snapshotVersion være slik (dersom man ønsker sanntids-data):

```
<dom:snapshotVersion>  
  <dom:timestamp>9999-01-01T00:00:00+01:00</dom:timestamp>  
</dom:snapshotVersion>
```

## Bruk av StoreService

Første parameter i storeService.getObject(s) er av typen MatrikkelBubbleId. Den kalles imidlertid alltid med en av subtypene som da angir hvilken retur-type man forventer. Eksempel i kode under:

```
Kommune kommune = (Kommune) storeService.getObject(kommuneId, matrikkelContext); //Vi må caste retur-verdien til rett bobleobjekt
```

Dersom man tester direkte på XML-nivå vil det sånn ut. Her er det altså xsi:type="kom:KommuneId" som angir at vi ønsker å hente ut en kommune

```
<stor:getObject>  
  <stor:id xsi:type="kom:KommuneId" xmlns:kom="http://matrikkel.statkart.no/matrikkelapi/wsapi/v1/domain/kommune" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <dom:value>3007</dom:value>  
  </stor:id>  
  <stor:matrikkelContext>...</stor:matrikkelContext>  
</stor:getObject>
```

# Koder

En del felter i modellen har en fast liste av mulige verdier, en kodeliste. Det er definert opp et sett av regler for koder, bruk av disse og måten vi behandler dem.

1. Id for en kode skal aldri gjenbrukes
2. Kodeverdi for en kode skal aldri gjenbrukes
3. Beskrivelse for en kode kan endres, men ikke slik at betydningen endres, bare skrivefeil og lignende.
4. Koder kan oppstå og forsvinne når som helst. Det er derfor viktig å synkronisere lokale koder mot systemet ofte.

Kodene i matrikkelen er bygget opp av id, kodeverdi, beskrivelse og ekstra felter. Id-ene er unike innenfor kodelisten, men ikke unike for hele systemet. Kodeverdien og beskrivelsen er ofte bestemt av en standard og beskrivelsen er lokalisert.

Det er laget en egen tjeneste for synkronisering av kodelister, **KodelisteService**. Denne har en metode for å hente ut alle kodelister.

## Forskjellige typer bobler

Det finnes flere forskjellige typer bobler i matrikkelsystemet. Noen har ekstra felter knyttet til seg, mens andre er enklere. Her følger en beskrivelse av de forskjellige typene.

### MatrikkelBubbleObject

Grunnobjektet for alle bobler. Alle bobler arver fra denne klassen. Ved å arve fra denne klassen viser objektene at de er bobler, at de har en unik numerisk id som identifiserer dem, og at de har et felt av **metadata** som viser hvilke felter på objektet som inneholder informasjon.

### Kode

Se **koder** over.

### MatrikkelBubbleObjectWithHistory

De fleste boblene er av denne typen. Typen tilbyr følgende felter:

- oppdateringsdato

- sluttdato
- versjonId
- oppdatertAv
- avsluttetAv
- versjon

Dette er versjoneringsfelter som beskriver når et objekt har blitt endret, hvem som har endret det og hvor mange ganger objektet har blitt endret. For sanntidsversjoner av objekter vil ikke `sluttdato` eller `avsluttetAv` ha noen verdi, men for historiske versjoner av objekter vil disse være fylt inn for å vise hvem som avsluttet en versjon av objektet. `Sluttdato` for ett objekt vil være `oppdateringsdato` for den neste versjonen av objektet hvis dette har blitt oppdatert.

## Domainklasse

Noen tjenester (`NedlastningService` og `EndringsloggService`) tar inn en parameter som angir hvilke bobler man er interessert i. Denne enumerasjonen inneholder ganske mange flere boblerklasser enn det som faktisk er støttet eller relevant når man ser begge tjenestene under ett. Følgende bobleklasser er aktuelle for bruk med disse to tjenestene:

- `MatrikkelBubbleObject` (kun for endringslogg, betyr "alt")
  - Kode (for endringslogg betyr dette kun koder som vi kan endre på uten å legge ut nye versjon)
  - Fylke
  - Kommune
  - Krets
  - Kretsflate
  - Veg
  - Adresse
    - Vegadresse
    - Matrikkeladresse
  - Matrikkelenhet
  - Teig
  - Teiggrense
  - Teiggrensepunkt

- Anleggsprosjeksjonsflate
- Anleggsprosjeksjongrense
- Anleggsprosjeksjonspunkt
- Forretning
- Bygg
  - Bygning
  - Bygningsendring
- Bruksenhet
- Person
  - FysiskPerson
  - AnnenPerson
  - JuridiskPerson
- Kulturminne
- Grunnforurensing
- Grunnerverv
- JordskifteKrevd
- Klage
- SamlaFastEiendom
- SefrakMinne
- Konsesjonsforhold
- KonsesjonsforholdUtskrift
- AvtaleGrensePunktfeste
- AvtaleStedbundenRettighet

Noen klasser er subklasser av andre klasser, som det fremkommer av hierarkiet. Her gir det kun mening å enten bruke superklassen eller kun bruke subklassen. Det gir for eksempel ingen mening å både spørre etter Adresse og Vegadresse.

---