Myndigheten
för civilt försvar



Teknisk dokumentation

API för datasynkronisering av RIB-databaserna –

Bibliotek, Farliga ämnen & Beräkningsmoduler

API för datasynkronisering av RIB-databaserna – Bibliotek, Farliga ämnen & Beräkningsmoduler

Myndigheten för civilt försvar
651 81 Karlstad

Enhet: Enheten för geodata och beslutsstödssystem

Publikationsnummer: MCF0214 – november 2026
ISBN-nummer: 978-91-7927-800-7
Tidigare utgiven: MSB2199 – april 2023, MSB1649 – december 2020

Förord

När Myndigheten för samhällsskydd och beredskap (MSB) nu stöpts om till Myndigheten för civilt försvar passar vi på att justera några detaljer i vårt API för synkronisering av data till applikationerna RIB Bibliotek, RIB Farliga ämnen och RIB Beräkningsmoduler. Men i grunden är det den tekniska plattformen som driftsattes 2023 som vi fortsätter att använda.

Versionen vi lanserar nu är byggd för vår nya grafiska profil, vilket framför allt märks i HTML- och CSS-data för Farliga ämnen. Vi publicerar den därför under nytt versionsnummer: API-version 3. Den tidigare MSB-versionen kommer att finnas kvar under en övergångsperiod, men med begränsningar i vilka uppdateringar vi kan leverera den vägen.

Vi uppmanar alla API-användare att snarast växla över till API-version 3. Vi kommer i närtid att aktivera ett ”du behöver uppdatera din programvara”-meddelande i databaserna som levereras via det gamla API:et.

Med detta sagt följer här en kort sammanfattning av de ändringar vi gjort:

- Det nya API:et har en ny URL.
- De grafiska element som finns i databaserna (generiska omslagsbilder i Bibliotek; HTML- och CSS-data i Farliga ämnen och Beräkningsmoduler) är anpassade till Myndigheten för civilt försvars grafiska profil.
- Stöd för byte av kundkategori, om klientsidan skickar med nuvarande kategori.
- Stoppordlistan som används i Bibliotek är justerad, vilket innebär att filerna för fulltextindexering i Bibliotek (ftix) nu inkluderar fler ord än förut.
- Farliga ämnen är på konceptuell nivå förberett för att kunna visa riskområde på karta, med hjälp av iframe-teknik och anslutning till vår webbserver. För detta reserveras parametern *RiskomradeCSV*.
- Andra nya parametrar i Farliga ämnen är *SkyddLivraddning* och *AvtalKemsamverkan*.
- Beräkningsmoduler har delats upp i standard- och pro-utgåva, vilket hanteras tekniskt genom att **tpl/index_pro.html** införts.

Karlstad i november 2026

Magnus Levein

Enheten för geodata och beslutsstödssystem

Innehåll

1. Inledning 5
1.1 Bakomliggande filosofi 5
1.2 Översikt över API:ets olika nivåer 6
1.3 Kundkategorier 6
1.4 Koppling till sökmotor 7
1.5 Eventuella framtida förändringar 7

2. Synkronisering med servern vid Myndigheten för civilt försvar 8
2.1 Den synkroniserade databasens struktur 8
2.2 Arbetsflöde när databasen uppdateras på en klient 9
2.3 Anrop till Myndigheten för civilt försvars server 10
2.3.1 Autentisering 11
2.3.2 LastUpdateCount? LastUpdated=<tidsstämpel>& Edition={standard|pro|mixed} 11
2.3.3 GetUpdatedAfter? LastUpdated=<tidsstämpel>& Edition={standard|pro|mixed} 11
2.4 Möjlighet till kontroll och felsökning 12

3. Databasens dokumenttyper 13
3.1 Dokumenttypen framgår av nyckelnamnet 13
3.2 Globala data ”global” 14
3.3 Markering av pro-utgåva ”pro” 14
3.4 Globala data ”topical” 15
3.5 Metadatadokumentet ”version” 15
3.6 Beräkningsmoduler som ”app/” 15
3.7 Omslagsbilder som ”cov/” 16
3.8 Stilmallar som ”css/” 16
3.9 Biblioteks- och ämnesposter som ”doc/” 16
3.10 Fulltextindex som ”ftix/” 16
3.11 Hjälpfiler som ”help/” 17
3.12 Bilder + alternativtext som ”img/” 17
3.13 Mediafiler som ”media/” 18
3.14 Malldokument som ”tpl/” 18
3.14.1 Malldokument i Farliga ämnen 18
3.14.2 Malldokument i Beräkningsmoduler 18
3.15 Typsnittsfiler inkluderas inte 19
3.16 Hantering av skyddade dokument i Bibliotek 19

4. Json-strukturer i Bibliotek 20
4.1 Utbrutna datamängder – ”global” 20
4.2 Utvalda aktuella data – ”topical” 21

4.3 Biblioteksposterna 21

5. Json-struktur i Farliga ämnen 25
5.1 Översiktlig struktur 25
5.2 Unik nyckel (”ID”) 25
5.3 Referenser till sektionsmallarna (”Templates”) 25
5.4 Ämnespostens egentliga data (”data”) 26
5.4.1 Data med datatypen *värde* 26
5.4.2 Data med datatypen *array* 29
5.4.3 Data med datatypen *nyckelförsedd array* 30
5.4.4 Specifik struktur för synonymer 30
5.4.5 Specifik struktur för hygieniska gränsvärden 31

6. Malldokumenten i Farliga ämnen 33
6.1 Varför malldokument? 33
6.2 Malldokumentens tredubbla funktion 34
6.2.1 Funktion som strukturell grund 34
6.2.2 Funktion som faktaartikel 35
6.2.3 Funktion som bärare av hjälptexter 36
6.3 Hur ämnespostens data görs tillgänglig för malldokumenten 36
6.3.1 Mappning av datatyper 36
6.3.2 Särskilda filter 37
6.4 Komplettering med HTML-header 37

Bilaga A – Exempel på komplett bibliotekspost 38

Bilaga B – Språkbehandling i indexeringsalgoritmen 39

Bilaga C – Exempel på komplett ämnespost 41

Bilaga D – Presentation av molekylformler 44

1. Inledning

Detta dokument vänder sig till utvecklare som vill hämta data från våra databaser *RIB Bibliotek* och *RIB Farliga ämnen* på maskinell väg, på samma sätt som våra egna applikationer uppdaterar sin data genom att ansluta till vår server. Eftersom även delsystemet *RIB Beräknings-moduler* använder samma synkroniseringsmetod behandlas den också, om än inte lika ingående.

Lösningen bygger på att mottagaren (klienten) skapar sig en lokal databas, och kommunikationen med servern hos Myndigheten för civilt försvar syftar till att hålla den lokala databasen uppdaterad.

Kommunikationsprotokollet som används för att kommunicera med servern är inte lämpligt för att direkt hämta information om en enskild databaspost (biblioteksdokument eller ämnespost). API:ets[1] syfte är i stället att möjliggöra effektiv synkronisering för nedkopplad åtkomst (”off line”) till vår databas.

1.1 Bakomliggande filosofi

I vår administrativa förvaltning av databaserna Bibliotek och Farliga ämnen använder vi klassiska SQL-databaser med ibland ganska komplicerad struktur. Strukturen är byggd för att göra administrationen effektiv, men den är samtidigt – i synnerhet för Farliga ämnen – svår att hämta data ur på ett smidigt sätt.

Vi har därför valt att använda ett mer ”färdigtuggat” överföringsformat i form av en Json-struktur. Det gör vi för vår egen skull. Vi hanterar ju själva flera olika programvaror som läser ur databasen: den installerade Windows-versionen, serverprogramvaran för webbsidan samt mobilappen Farliga ämnen. Under tidigare år har vi råkat ut för buggar som berodde på att en av de olika ”läsarna” läste fel i något sammanhang. Nu vill vi eliminera den möjligheten genom att använda ett centralt ”läsprogram” som hämtar uppgifterna från administrations-databasen och distribuerar dem vidare på ett mer lättillgängligt sätt.

En annan viktig aspekt av ”databasen” Farliga ämnen, är att mycket av informationen faktiskt inte administreras i någon databas, utan som fristående filer. Det rör sig framför allt om de textintensiva delarna *Information för räddningstjänsten* och *Akut omhändertagande på olycksplats*, som administreras i form av textfiler och bearbetas till ett slags HTML-liknande filer. Andra exempel på fristående data är bildfilerna som illustrerar olika varningsetiketter, samt förstås hela systemet med hjälptexter.

[1] API = Application Progamming Interface

På motsvarande sätt består ”databasen” Bibliotek också av flera olika typer av fristående filer vid sidan av SQL-databasen, bland annat omslagsbilderna och själva dokumentfilerna.

För att kunna hantera hela vidden av olika typer av data på ett enhetligt sätt, från Json-strukturen som beskriver en ämnespost till pdf-filen för ett biblioteks-dokument, har vi valt att bygga den distribuerade databasen som en generell dokumentdatabas. För Farliga ämnen innebär den valda lösningen att vi skulle kunna göra ganska stora förändringar i ämnesposternas uppbyggnad, utan att för den skull behöva ändra i klientprogramvaran. För Bibliotek innebär lösningen att vi på klientsidan inte behöver hantera tiotusentals filer, utan klarar oss med en handfull filer (för databas och fulltextindex).

1.2 Översikt över API:ets olika nivåer

Beskrivningen av API:et är indelad i tre nivåer:

1. **Synkroniseringsnivån** – På denna nivå hanteras all information som tids-stämplade *dokumentposter* som synkroniseras i en generell dokumentdatabas med enkel struktur. Kommunikationen med servern vid Myndigheten för civilt försvar sker på denna nivå.
2. **Dokumentnivån** – Här förklaras vilka olika dokumenttyper som hanteras i databasen. De enklare dokumenttyperna beskrivs fullständigt här, medan de mer komplicerade dokumenttyperna enbart beskrivs översiktligt.
3. **Detaljnivån** – De dokumenttyper som kräver mer detaljerad förklaring förklaras i flera separata kapitel.

Rekommenderad läsordning är att läsa kapitlen i den ordning de kommer. Letar man efter specifika detaljer kan man slå upp dem i respektive kapitel, men för att förstå detaljbeskrivningen på en nivå behöver man vara insatt i förklaringen av de överliggande nivåerna.

1.3 Kundkategorier

Vi ger ut beslutsstödet RIB i två olika utgåvor: *standard*, som är tillgänglig för vem som helst, och *pro*, som enbart är tillgänglig för blåljusaktörer och relevanta myndigheter. Denna åtskillnad finns inte i delsystemet Farliga ämnen, men i Bibliotek och Beräkningsmoduler finns det däremot dokument respektive moduler som enbart pro-kunder kommer åt. Att uppdelningen görs också i Beräknings-moduler är en nyhet i och med API-version 3.

Utöver detta finns också kundkategorin *utvecklare*, för tredjepartstillverkare av mjukvara som vill ansluta till våra API:er utan ett bakomliggande abonnemang. Konto som utvecklare är kostnadsfritt och förfrågan sänds till rib@mcf.se. Utvecklare får tillgång till API:et (med behörighetsnivån standard), men kan däremot inte ladda ner programvara från vår nedladdningssida. Den licensnyckel vi ger till utvecklare är inte enbart avsedd för testning, utan vi tillåter att den distribueras inbäddad i de programvaror som utvecklas.

Hur synkroniseringsmekanismen hanterar kundkategorierna beskrivs i kapitel 2, och berörs även i kapitel 3.

1.4 Koppling till sökmotor

Den typ av synkroniserad databas som detta dokument beskriver behöver kopplas till en sökmotor för att användaren ska kunna söka i datamängden. Myndigheten för civilt försvar använder sökmotorn Lucene.Net, men sökmekanismen kan lösas på valfritt sätt.

Vi matar sökmotorn med utvalda data från biblioteksposter respektive ämnesposter. Men för att kunna implementera fulltextsökning i Bibliotek på ett bra sätt använder vi dessutom en särskild dokumenttyp för just fulltextindexeringen. Detta förklaras närmare i kapitel 3.10.

1.5 Eventuella framtida förändringar

Vi hoppas att detta API kommer att vara stabilt under många år, åtminstone i de övergripande delarna. På detaljnivån i Json-strukturen för biblioteksposter och ämnesposter kan vi däremot vänta oss att ytterligare värden tillförs, förmodligen redan under nästa år. Denna typ av förändringar bör dock inte leda till problem med bakåtkompatibiliteten.

Om vi skulle behöva göra så stora förändringar i API:et att bakåtkompabiliteten bryts, är tanken att vi ska publicera det nya API:et under en ny adress och låta både gammalt och nytt API leva parallellt under en övergångsperiod. Detta ska då annonseras genom metadatadokumentet ”version” i databasen, se kapitel 3.5.

Bytet av grafisk profil, från MSB till Myndigheten för civilt försvar, är en förändring som bryter bakåtkompatibiliteten för våra egna applikationer, och det är därför som vi nu lanserar API-version 3.

2. Synkronisering med servern vid Myndigheten för civilt försvar

Synkroniseringsnivån – Kapitlet beskriver hur mottagaren (klienten) skapar sig en lokal databas för respektive delsystem i RIB (Bibliotek, Farliga ämnen eller Beräkningsmoduler), och sedan håller den uppdaterad i förhållande till vår centrala databas.

2.1 Den synkroniserade databasens struktur

De olika delsystemen inom RIB betraktas som oberoende av varandra.

All data i ett delsystem hanteras som nyckel-värde-par, och hela datamängden hanteras i en enda SQL-tabell. SQL-tabellens struktur för att hantera vår nyckel-värde-data ser likadan ut på serversidan och på klientsidan:

kolumn ”key”	**kolumn ”hidden”**	**kolumn ”value”**	**kolumn ”timestamp”**
Strängvärde (max 40 tecken) som fungerar som primärnyckel i tabellen.	Synlighetsvärde för denna rad (8 bit integer): 0 = synlig, 1 = synlig för pro-kunder, 2 = raderad (För synkronise-ringens skull måste alla ”key” som någonsin funnits finnas kvar.)	Binär data (BLOB) beroende på synlighetsvärdet i kolumn ”hidden”: 0 = innehåll, 1 = innehåll om pro-kund, annars tom 2 = tom (0 bytes)	Tidsstämpel som anger när senaste uppdateringen av denna rad gjordes. 64 bit signed integer, samma värde som DateTime.Ticks enligt UTC.
Denna kolumn ska vara indexerad.			Denna kolumn ska vara indexerad.
Ingen av kolumnerna kan anta NULL-värden.			

Varje rad i datatabellen motsvarar en *dokumentpost*. Som vi ska se i nästa kapitel finns några olika typer av dokumentposter, men initialt väljer vi att betrakta dem på samma sätt.

Vid uppdatering på server-sidan kommer endast de dokumentposter som får nya värden att uppdateras, och deras ”timestamp” sätts då till aktuell tidpunkt. Alla poster som uppdateras i en och samma transaktion på serversidan får samma ”timestamp”-värde, i syfte att underlätta eventuell felsökning.

Ett ”key”-värde som någon gång lagts till i tabellen kan inte tas bort, eftersom det skulle förvilla synkroniseringen. I stället kommer kolumnen ”hidden” att sättas till 2 och ”value” att tömmas (till 0 bytes). Naturligtvis uppdateras också ”timestamp” till aktuell tidpunkt.

De poster som har begränsad synlighet, och endast är tillgängliga för pro-kunder, har värde 1 i kolumnen ”hidden”. För dessa poster returnerar servern olika värden i kolumnen ”value” beroende på om klienten är en pro-kund eller inte. (För pro-kunder returneras det faktiska innehållet, och för övriga returneras tomt värde, 0 bytes.)

Detta upplägg möjliggör inkrementell synkronisering av ändringar i databasen. Det möjliggör också att ett dokuments eventuella begränsning till pro-användare kan förändras över tid i båda riktningar. En nyhet är att API:et nu även stödjer scenariot att en kund av någon anledning byter kundkategori, från standard till pro eller tvärtom.

2.2 Arbetsflöde när databasen uppdateras på en klient

1. Klienten undersöker om den lokala databasen är av standard- eller pro-version, eller om den befinner sig i ett ”mixed”-tillstånd (vilket kan ha uppstått om kundkategorin tidigare har ändrats utan att databasen tömts). Det kan göras genom att till exempel köra följande SQL-fråga mot den lokala databasen:[2]

   ```
   SELECT
     COUNT(CASE WHEN hidden = 1 AND LENGTH(value) = 0 THEN 1 END) AS Empty,
     COUNT(CASE WHEN hidden = 1 AND LENGTH(value) > 0 THEN 1 END) AS Filled
   FROM KeyValueDocument;
   ```

   Poängen är att räkna hur många poster med hidden-värde 1 som har BLOB-data och hur många som har tom BLOB. Resultatet tolkas sedan enligt följande:

   - Om Filled = 0 ⇒ **standard**
   - Annars, om Empty = 0 ⇒ **pro**
   - Annars ⇒ **mixed**

[2] SQL-koden fungerar i SQLite. För att köra på SQL Server behöver LENGTH() ändras till LEN().

2. Klienten frågar servern ”Hur många nya dokumentposter finns det, givet att min nyaste dokumentpost har tidsstämpel X, och givet att min databas är standard/pro/mixed? – Och vilket är förresten din nyaste tidsstämpel?” (Om klientens databas är tom skickar den 0 som tidsstämpel.)
3. Servern svarar med antal nya dokumentposter, samt tidsstämpel för den nyaste.
4. Klienten avgör (eventuellt genom att fråga användaren) om den vill hämta de nya posterna eller avvakta. Vilken logik som används här kan bero på sammanhanget.
5. Klienten ber servern ”Skicka mig alla nya dokumentposter, givet att min nyaste dokumentpost har tidsstämpel X, och givet att min databas är standard/pro/mixed”.
6. Servern skickar ett MessagePack-dokument som innehåller en array med de dokumentposter som behöver uppdateras. (Uttaget från servern görs med en enda SELECT-sats, isolerad så att den inte påverkas av en eventuell samtidig uppdaterings-transaktion på servern.)
7. Klienten påbörjar uppdateringen av den lokala databasen, med de uppdaterade eller nytillkomna dokumentposter som kommer. Uppdateringen ska göras som en transaktion, och om det visar sig att svaret inte är komplett (d.v.s. om den skickade arrayen inte kommer fram som helhet, t.ex. p.g.a. ett avbrott i förbindelsen) ska uppdateringen avbrytas och rullas tillbaka.

2.3 Anrop till Myndigheten för civilt försvars server

Synkroniseringsmekanismen bygger på två olika typer av anrop till vår server: **LastUpdateCount** och **GetUpdatedAfter**. Anropen görs som vanliga http GET-anrop, och basadressen till tjänsterna är URL
https://ribdataapi.mcf.se/library/v3/documents,
https://ribdataapi.mcf.se/hazmat/v3/documents respektive
https://ribdataapi.mcf.se/calculations/v3/documents.

Vid felaktiga anrop returneras en felsignalerande http-status (typiskt 400 Bad Request eller liknande) samt en Json-kodad felbeskrivning.

De tidsstämplar som används som parametrar är samma heltal (64 bit signed integer) som används i databaskolumnen ”timestamp”. Det är det ”ticks”-värde som C#-funktionen DateTime.UtcNow ger. (Se Microsofts .Net-dokumentation för närmare detaljer.)

2.3.1 Autentisering

För att ansluta till vårt API krävs antingen ett RIB-abonnemang (som standard- eller pro-kund), eller ett konto som utvecklare (vilket kan erhållas utan kostnad, se kapitel 1.3). Inloggningen används för att avgöra om data som är begränsad till pro-användare ska tillhandahållas eller inte, samt för att kunna föra statistik över hur mycket tjänsten utnyttjas.

Som autentiseringsmekanism används ”Basic access authentication” (specificerad i RFC 7617) på detta sätt:

Användarnamn	**Lösenord**
det kundnummer som tilldelats organisationen	den licensnyckel som tilldelats organisationen

2.3.2 LastUpdateCount? LastUpdated=<tidsstämpel>& Edition={standard|pro|mixed}

Funktionen LastUpdateCount anropas med dels en parameter som anger tidsstämpeln för klientens nyaste dokumentpost (om klientens databas är tom anger man 0), och dels en parameter som anger om klientdatabasen är av standard- eller pro-version – eller om den är skadad och är en blandning av standard- och pro (”mixed”). Edition-parametern är frivillig, men för Bibliotek och Beräkningsmoduler bör den användas för att stödja scenariot med ändrad kundkategori.

Servern svarar dels med hur många poster som behöver uppdateras hos klienten, och dels vilken tidsstämpel som den nyaste dokumentposten har. Svaret ges som en Json-sträng.

Exempel på fråga

```
https://ribdataapi.mcf.se/hazmat/v3/documents/LastUpdateCount?LastUpdated=
639275008568259729&Edition=standard
```

Exempel på svar

```
{"lastUpdated":639275836937010431,"items":63}
```

2.3.3 GetUpdatedAfter? LastUpdated=<tidsstämpel>& Edition={standard|pro|mixed}

Funktionen *GetUpdatedAfter* anropas med dels en parameter som anger tidsstämpeln för klientens nyaste dokumentpost (om klientens databas är tom anger man 0), och dels en parameter som anger om klientdatabasen är av standard- eller pro-version – eller om den är skadad och är en blandning av standard- och pro (”mixed”). Edition-parametern är frivillig, men för Bibliotek och

Beräkningsmoduler bör den användas för att stödja scenariot med ändrad kundkategori.

Servern returnerar alla dokumentposter som behöver uppdateras hos klienten. Svaret ges som ett MessagePack-dokument som innehåller en array med de efterfrågade dokumentposterna.

MessagePack-dokumentets första beståndsdel är alltså ”array format family”, och den första byten är således 0x9*n* eller 0xdc eller 0xdd beroende på hur lång arrayen är. Jämför specifikationen för MessagePack.[3] Varje tabellrad kodas sedan i sin tur också som ”array format family” med fyra enheter (en för varje kolumn); det blir således byte 0x94 som inleder varje ny tabellrad.

Kolumnerna skickas alltså utan sina kolumnnamn för att spara plats. De skickas i ordningen ”key”, ”value”, ”timestamp”, ”hidden”.

(För att kunna hantera stora datamängder rekommenderas att dataposterna hanteras löpande.)

Exempel på fråga

```
https://ribdataapi.mcf.se/hazmat/v3/documents/GetUpdatedAfter?LastUpdated=
638169720384141010&Edition=standard
```

Exempel på svar[4]

```
\x91
  \x94
    \xa7version
    \xc4\x28{"UseUntil":"2027-06-30","UpdateApp":""}
    \xcf\x08\xd8\x7a\x6f\xa7\x99\x18\x04
    \x00
```

(Texten är raduppdelad och indragen för att lättare kunna läsas. Binär data visas med traditionell C-notation som \x*nn*, där *nn* är bytens hexadecimala värde.)

2.4 Möjlighet till kontroll och felsökning

Ett enkelt sätt att kontrollera om databasen synkroniserats korrekt är att göra en sammanställning över vilka tidsstämplar som finns, och hur många dokument-poster som har respektive tidsstämpel.

Uttryckt med SQL kan det till exempel se ut på detta sätt:

```
SELECT timestamp, count(*) FROM KeyValueDocument GROUP BY timestamp ORDER BY 1
DESC;
```

(I vår app Farliga ämnen skrivs just denna information ut för de tio senaste tidsstämplarna, om man väljer Övrigt > Databas > Detaljer för felsökning. Tidsstämplarna skrivs där ut som datum och klockslag i UTC.)

---

[3] https://github.com/msgpack/msgpack/blob/master/spec.md

[4] Exemplet är MessagePack-motsvarigheten till detta Json-uttryck: [["version", "{\"UseUntil\":\"2027-06-30\",\"UpdateApp\":\"\"}", 637393967244711940, 0]]

3. Databasens dokumenttyper

Dokumentnivån – Kapitlet beskriver de olika dokumenttyper som används i synkroniseringsdatabaserna för Bibliotek, Farliga ämnen och Beräkningsmoduler.

3.1 Dokumenttypen framgår av nyckelnamnet

Som vi såg i föregående kapitel är den synkroniserade databasen uppbyggd som en nyckel-värde-lagring, där varje rad i databasen har en nyckel (kolumnen ”key”) och ett värde (kolumnen ”value”).

För att på ett enkelt sätt särskilja olika typer av dokumentposter inleds nyckelnamnen med prefix, på ett sätt som påminner om en katalogstruktur. De speciella dokumenten ”global”, ”pro”, ”topical” och ”version” har dock inget egentligt prefix, eftersom det alltid bara förekommer (som mest) ett sådant dokument.

Följande prefix används:

Prefix	**Fullständigt nyckelnamn (exempel)**	**Förklaring** ***All text är kodad i UTF-8.***	**Bibliotek**	**Farliga ämnen**	**Beräkn-moduler**
global	global	En singulär post med globala data som ändras sällan. Json.	X	-	-
pro	pro	En singulär post som indikerar om det är pro-utgåva eller inte.	X	-	-
topical	topical	En singulär post med globala data som ändras regelbundet. Json	X	-	-
version	version	En singulär post med versions-information för hela databasen. Json.	X	X	X
app/	app/ riskavstand.html	Komplett webbapplikation sparad som HTML (med CSS, Javascript och bilder inbäddade).	-	-	X*
cov/	cov/4321.png	Omslagsbild (png) för bibliotekspost.	X	-	-
css/	css/app.css	Stilmallar (CSS) sparade som text.	-	X	-
doc/	doc/4321	Biblioteks- respektive ämnes poster i Json-format.	X	X	-
ftix/	ftix/4321	Fulltextindex för bibliotekspost. MessagePack.	X	-	-

help/	help/fahjalp.html	Hjälpfil sparad som HTML.	-	X	-
img/	img/etk_2_3.png	Småbilder med tillhörande förklaringstext, sparade i en Json-struktur med base64-kodad binärdata (png).	-	X	-
media/	media/4321.pdf	Mediafil för bibliotekspost (pdf, mp4, etc).	X	-	-
tpl/	tpl/sirib45.htm	Malldokument sparade som HTML i *Liquid Templating Language* (för Farliga ämnen) eller som ”naken HTML” (för Beräkningsmoduler)*.	-	X	X*

*) För Beräkningsmoduler utgörs startfilen (tpl/index.html eller tpl/index_pro.html) av ett mall-dokument i form av löst HTML-innehåll som behöver omgärdas av bl.a. sidhuvud och sidfot. De enskilda modulerna är dock kompletta webbapplikationer sparade under ”app/”.

Som nämns i tabellrubriken är alla texter (inklusive CSS, HTML och Json) kodade i UTF-8.

De olika dokumenttyperna förklaras närmare i de följande underkapitlen.

3.2 Globala data ”global”

Precis som beskrivs i kapitel 1.1 är den bakomliggande filosofin att databasens distribuerade format ska vara så ”färdigtuggat” som möjligt, utan tabeller som refererar till varandra och annan komplex logik. För Bibliotek behöver dock några datamängder brytas ut från det ”färdigtuggade” formatet, för att möjliggöra sökningar på ett effektivt sätt. Det handlar om delarna *utgivare*, *serie, mediatyp* och *ämneshylla*.

Dokumentposten ”global” definierar de här tabellerna med hjälp av en Json-struktur, och innehållet förväntas inte ändras särskilt ofta. Närmare detaljer ges i kapitel 4.

3.3 Markering av pro-utgåva ”pro”

Databasen för Bibliotek innehåller ett särskilt flaggdokument i form av dokumentposten ”pro”: på server-sidan innehåller det texten ”PRO” med kolumnen ”hidden” = 1, d.v.s. endast synlig för pro-kunder.

På klientsidan kommer dokumentposten ”pro” således att ha ett tomt innehåll för standard-kunder, men texten ”PRO” för pro-kunder. Vitsen med detta är att klientapplikationen på så sätt enkelt kan läsa av om databasen är en standard-databas eller en pro-databas, vilket Bibliotek-mjukvaran behöver veta för att kunna visa partiellt skyddade dokument på rätt sätt i träfflistan, se kapitel 3.16.

Det här sättet att flagga pro-utgåvan underlättar också felsökning.

3.4 Globala data ”topical”

I Bibliotek har vi behov av att hantera de tips som våra bibliotekarier ger: ett aktuellt utvalt tema, särskilt utvalda ämneslänkar, populära serier och populära utgivare. Det är inga stora datamängder det handlar om, men det kan bli ganska frekventa uppdateringar.

Dokumentposten ”topical” hanterar detta med en Json-struktur. Närmare detaljer ges i kapitel 4.

3.5 Metadatadokumentet ”version”

Alla tre databaser har ett särskilt metadatadokument med nyckelnamnet ”version”. Syftet är att vi ska kunna hantera två särskilda situationer:

a) Användaren har en databas som aldrig blir uppdaterad, t.ex. för att enheten inte är ansluten till internet. Då ska ett varningsmeddelande visas när databasen passerat sitt bäst före-datum.

b) Vi vill även kunna varna användare för att de måste uppdatera sin programvara. Det kan till exempel vara så att vi bygger om datastrukturen så att kompatibiliteten bryts. Även då ska ett varningsmeddelande visas när användaren har kontakt med vår server.

Metadatadokumentet är en Json-struktur med två värden, *UseUntil* och *UpdateApp*. Innehållet ser typiskt ut på detta sätt:

```
{"UseUntil":"2027-06-30","UpdateApp":""}
```

Värdet **UseUntil** är databasens bäst före-datum. Klienten förväntas jämföra detta med dagens datum enligt lokal tid. Om bäst före-datumet har passerats finns det risk att allt som står inte längre gäller. Klienten ska då påtala för användaren att **databasen är för gammal och behöver uppdateras** eftersom regelverk med mera kan ha ändrats. Användaren ska dock tillåtas att fortsätta använda sin databas på egen risk.

Värdet **UpdateApp** är i normalfallet tomt. Om det innehåller någon text förväntas klienten visa denna text för användaren som ett varningsmeddelande. Om vi behöver göra så stora förändringar i API:et att bakåtkompabiliteten bryts lägger vi ett meddelande i detta fält innan vi upphör med uppdateringarna. Användarna kan på det sättet uppmanas att uppdatera sin programvara. Den nya versionen av programvaran läser från en annan URL och därmed ur en annan datatabell. Jämför kapitel 1.5.

3.6 Beräkningsmoduler som ”app/”

De enskilda beäkningsmodulerna i programdelen Beräkningsmoduler distribueras som webbapplikationer i HTML-kod, med inbäddad CSS, Javascript och base64-kodade inbäddade bilder. De lagras som dokumenttyp ”app”.

De typsnittsfiler som den inbäddade CSS-koden hänvisar till finns dock inte med i den distribuerade databasen, se kapitel 3.15.

3.7 Omslagsbilder som ”cov/”

Omslagsbilder till böcker, filmer och annat som finns i Bibliotek lagras som dokumenttyp ”cov”. Varje biblioteksposti nkluderar information om vilken omslagsbild som ska användas.

I de flesta fall används en unik omslagsbild för varje biblioteksposti, men det förekommer även generella omslagsbilder.

Något oegentligt används dokumenttypen ”cov” också för omslagsbilden till det aktuella temat. I sådant fall är nyckelnamnet ”cov/topical.jpg”. (Vi vill använda ett statiskt namn för att inte fylla på med nya nycklar i onödan, eftersom inlagda nyckelnamn aldrig får tas bort.)

3.8 Stilmallar som ”css/”

Dokumenttypen ”css” används för stilmallar i CSS som används i Farliga ämnen, nämligen ”css/app.css” och ”css/hazmat.css”.

3.9 Biblioteks- och ämnesposter som ”doc/”

Kärnan i Biblioteks- och Farliga ämnes-databaserna är själva biblioteksposterna (med titel, författare, o.s.v.) respektive ämnesposterna (med alltifrån ämnesnamn till kokpunkt). De lagras som dokumenttyp ”doc”. Dokumentposternas fullständiga nyckelnamn utgörs av prefixet ”doc/” följt av biblioteksposti ens unika dokid respektive ämnespostens unika ämnes-id.

Dokid är primärnyckel i Biblioteks administrationsdatabas, och är alltså ett internt nummer som unikt identifierar en biblioteksposti; just nu 1~31 500 (och passerar snart 15 bitars-gränsen 32 767). I Farliga ämnes-databasen kallas primärnyckeln ämnes-id och är just nu 1~10 000.

Dokumenttypen ”doc” utgörs av en ganska lång Json-struktur, som innehåller flera olika typer av data. En detaljerad beskrivning ges i kapitel 4 för Bibliotek respektive i kapitel 5 för Farliga ämnen. Kompletta exempel ges i bilaga A respektive C.

3.10 Fulltextindex som ”ftix/”

För att möjliggöra fulltextsökning av (så gott som) alla dokument i Bibliotek, även de som bara är tillgängliga on-line, distribueras färdigpreparerade ordfrekvensfiler för dokumenten som dokumenttyp ”ftix”. Formatet som används är MessagePack, d.v.s. en binär motsvarighet till Json (jfr kapitel 2). Översatt till Json ser strukturen ut på detta sätt:

```
{
  "civilt": 2,
  "försv": 2,
  "målet": 2,
  "beredskap": 1,
  "bidr": 1,
```

```
  "förberet": 1,
  "höjd": 1,
  "krig": 1,
  "kris": 1,
  "krisberedskap": 1,
  "mot": 1,
  "mynd": 1,
  "mål": 1,
  "olyck": 1,
  "samhället": 1,
  "samt": 1,
  "skydd": 1,
  "säkerställ": 1,
  "totalförsvaret": 1,
  "uppfyll": 1,
  "verksam": 1,
  "övergrip": 1
}
```


Observera att exemplet ovan är en Json-visualisering av strukturen. Rent faktiskt är det en binär MessagePack-struktur som används.

Exemplet visar indexeringen av texten ”Myndigheten för civilt försvar ska säkerställa att samhället är förberett för kris, höjd beredskap och krig. Vår verksamhet bidrar till att uppfylla målet för civilt försvar och det övergripande målet för totalförsvaret samt målen för skydd mot olyckor och krisberedskap”. För varje ord anges antal förekomster i texten. Så kallade stoppord (som är så vanliga att de inte tillför något värde i söksammanhang) är helt bortplockade. Dessutom är orden trunkerade med en så kallad *stemmer*, som klipper av alla ordändelser för att böjningsvarianter av ord ska indexeras som samma ord. (Så länge samma stemmer används vid sökningen som vid indexeringen får man träff även om algoritmen råkar klippa för hårt ibland.)

I bilaga B beskrivs närmare detaljer om hur vi tillverkar ftix-posterna.

3.11 Hjälpfiler som ”help/”

Dokumenttypen ”help” används för den övergripande hjälpen i Farliga ämnen, dokumentposten ”help/fahjalp.html”. Dess exakta innehåll och funktion ingår inte i API-definitionen och kan ändras utan förvarning.

3.12 Bilder + alternativtext som ”img/”

Småbilder i Farliga ämnen lagras som dokumenttyp ”img”; det rör sig framför allt om varningsetiketter. Eftersom varje bild behöver ha en alternativtext associerad till sig används en enkel Json-struktur som inkluderar både alternativtext och bilddata:


```
{"title":"Transportetikett 2.3: Giftiga gaser",
"data":"iVBORw0KGgoAAAANSUhEUgAAAFQAAABUCAYAAAFrbCBXAAAACXBIWXMAAAsTAAALEwEAmp
/.../
IgMfxMwAkqtdfMqw9kQAAAABJRU5ErkJggg=="}
```


Exemplet visar dokumentposten ”img/etk_2_3.png”. Bildfilen är en PNG-fil kodad som Base64. (För närvarande är alla bildfiler PNG-filer, vilket återspeglas i att dokumentposternas nyckelnamn slutar på ”.png”, men även andra filformat är tänkbara i framtiden.)

3.13 Mediafiler som ”media/”

Dokumenttypen ”media” används för själva mediafilerna i biblioteket, i huvudsak pdf-dokument men även andra filformat kan förekomma. Varje bibliotekspost inkluderar information om vilken mediafil som ska användas, men många bibliotekposter använder webb-länkad media som alltså inte finns med i synkroniseringsdatabasen.

3.14 Malldokument som ”tpl/”

3.14.1 Malldokument i Farliga ämnen

Varje innehållsflik i Farliga ämnen (som t.ex. Akutvård, Fysdata eller Miljö) är byggd på en mall. Vissa flikar använder en och samma statiska mall oavsett ämnespost, medan andra flikar använder olika mallar för olika ämnesposter.

Speciellt kan nämnas att flikarna Räddning och Akutvård innehåller faktatexter i mallarna, anpassade på så sätt att olika mallar används för olika grupper av ämnen. (Detta sätt att använda malldokumenten på skulle mycket väl kunna byggas ut till att omfatta fler flikar i framtiden, t.ex. fliken Miljö.)

Varje ämnespost inkluderar information om vilka malldokument som ska användas för just den ämnesposten. Klientprogramvaran behöver därför inte ha någon närmare kunskap om ifall det är fråga om en statisk mall eller en mall som bara används av en specifik ämnesgrupp.

Malldokumenten är HTML-filer berikade med *Liquid Templating Language*, vilket innebär att de innehåller platshållare (referenser) till ämnespostens olika datavärden. De innehåller också viss logik med villkor för att t.ex. dölja eller visa vissa rubriker eller statiska texter beroende på den aktuella ämnespostens olika datavärden.

Ämnesposterna (”doc/”) och malldokumenten (”tpl/”) fungerar alltså i symbios med varandra: ämnesposterna hänvisar till vilka malldokument som ska användas för det aktuella ämnet, och malldokumenten ger närmare instruktioner om vilka av ämnespostens datavärden som ska presenteras var, när och hur.

Denna konstruktion gör systemet mycket flexibelt, och det är möjligt att t.ex. införa helt nya datavärden i ämnesposten och få dem korrekt presenterade för användarna utan att behöva göra några ändringar i klientprogramvaran.

I kapitel 6 ges närmare detaljer om malldokumenten.

3.14.2 Malldokument i Beräkningsmoduler

Beräkningsmoduler använder dokumentposterna ”tpl/index.html” och ”tpl/index_pro.html” för sin startsida. De innehåller dels en ingress och dels länkar till de olika beräkningsmodulerna. HTML-koden är ”naken” och behöver kompletteras med bl.a. sidhuvud och sidfot.

Klientprogramvaran ska först se om ”tpl/index_pro.html” har något innehåll. Om den finns och har innehåll är det den som ska användas. I annat fall ska ”tpl/index.html” användas.

3.15 Typsnittsfiler inkluderas inte

Som framgår av ovanstående genomgång inkluderas ganska många typer av data i de distribuerade databaserna. Däremot inkluderas inte de typsnittsfiler som CSS-stilmallarna (i Farliga ämnen och Beräkningsmoduler) refererar till.

Exakt vilka typsnitt som de inkluderade CSS-mallarna hänvisar till, och som behövs för att stilsättningen ska fungera som avsett, ingår inte i API-definitionen utan kan ändras från tid till annan.

Dock kan nämnas att det i skrivande stund är typsnitten *Inter* och *DS-Icons* som utnyttjas, varav det sistnämnda är vår egen reducerade version av Font Awesome Free Solid (med pilar och andra symboler). DS-Icons går att extrahera från rib.mcf.se med webbläsarens utvecklarverktyg.

3.16 Hantering av skyddade dokument i Bibliotek

Bibliotek hanterar tre nivåer av skydd:

- **Publika dokument.** Kolumnen ”hidden” har värde 0 för alla dokumentposter som rör denna bibliotekspost.
- **Partiellt skyddade dokument.** Detta är dokument som endast pro-kunder kan läsa, men vars metadata tillhandahålls öppet. Man kan alltså vid en sökning få träff på dem även i standard-versionen och på webben. För denna skyddsnivå är dokumentposterna i kategori *ftix/* och *media/* markerade med ”hidden”-värde 1, medan *cov/* och *doc/* har ”hidden”-värde 0. (Denna kategori är tänkt för insatskort för tåg, som vi vill marknadsföra förekomsten av.)
- **Helt skyddade dokument.** Detta är dokument som endast pro-kunder kommer åt. De dyker inte alls upp vid sökning i standard-versionen eller på webben. För denna skyddsnivå är alla fyra kategorierna (*cov/*, *doc/*, *ftix/* och *media/*) markerade med ”hidden”-värde 1. (Vid en manuell inspektion av synkroniseringsdatabasen på en standard-klient kommer man att kunna se nyckelnamnen på dokumentposterna, men eftersom de bara är löpnummer är de inte särskilt intressanta.)

4. Json-strukturer i Bibliotek

Detaljnivån – Kapitlet beskriver Json-strukturerna som används av Bibliotek: ”global”, ”topical” och biblioteksposterna. (Hur dessa lagras i den synkroniserade databasen förklarades i kapitel 3.)

4.1 Utbrutna datamängder – ”global”

I biblioteksposterna lagras *utgivare*, *serie, mediatyp* och *ämneshylla* med referenstal i stället för i klartext. Sammankopplingen mellan dessa referenstal görs i dokumentposten ”global”, i form av en Json-struktur byggd på detta sätt:

```
{
  "Utgivare":
  [
    {"93": "Försvarets forskningsanstalt (FOA)"},
    {"334": "Räddningsverket (SRV)"},
    {"515": "Lunds tekniska högskola (LTH)"},
    {"733": "Major Accident Hazards Bureau (MAHB)"},
    {"1294": "Myndigheten för samhällsskydd och beredskap (MSB)"},
    ...
  ],
  "Serie":
  [
    {"20": "FOA Rapport"},
    {"28": "LUTVDG/TVBB"},
    {"142": "Brandutredning/SRV"},
    {"320": "Utökade olycksundersökningar"},
    ...
  ],
  "Mediatyp":
  [
    {"1": "CD"},
    {"8": "Häfte"},
    {"9": "Videofilm"},
    {"11": "Tidskriftsartikel"},
    ...
  ],
  "Hylla":
  [
    {"0": "Allmänt"},
    {"0-0": "Uppslagsverk"},
    {"0-1": "Språkvetenskap"},
    {"0-1-0": "Ordböcker"},
    {"0-1-0-0": "Svenska"},
    {"0-1-0-1": "Engelska"},
    {"0-1-0-2": "Tyska"},
    ...
    {"3": "Räddningstjänst" },
    ...
    {"3-4":"Brand" },
    {"3-4-0":"Allmän brandkunskap" },
    {"3-4-1":"Brandförlopp" },
    ...
  ]
}
```

Som synes är *Hylla* i sig en trädstruktur, med som mest fyra nivåer. I Json-representationen lagras den dock tillplattad, med vanligt bindestreck (ASCII-kod 45) som avgränsare i nyckelnamnet.

4.2 Utvalda aktuella data – ”topical”

Dokumentposten ”topical” beskriver aktuellt utvalt tema och utvalda ämneslänkar, samt populära serier och populära utgivare, i form av en Json-struktur, byggd enligt nedanstående exempel.

```
{
  "Tema":
  {
    "Rubrik": "Översvämning",
    "Omslag": "cov/topical.jpg",
    "Ingress": "Läs om förstärkningsresurser, erfarenheter, ...",
    "URL": "?h=3-7-0-4&rs=N"
  },
  "Utvalda":
  [
    {"Namn": "Insatskort tåg",
      "URL": "?s=587&h=3-5-1" },
    {"Namn": "Hemberedskap",
      "URL": "?nyck=Hemskydd" },
    {"Namn": "Utökade olycksundersökningar",
      "URL": "?s=320" }
  ],
  "PopSerier": [ 333, 592, 320, 582, 586 ],
  "PopUtgivare": [ 1545, 1294, 919, 473, 1398 ]
}
```

Temats omslag är i regel just "cov/topical.jpg", men det skulle kunna vara *null* om ingen omslagsbild ska användas. Ett tema används för att kunna lyfta fram ett ämne som förmodas vara särskilt intressant för våra användare just nu. Typiskt kopplat till årstidsvariationer eller pågående större händelser.

Både temat och de utvalda sökningarna är helt enkelt relativa webbadresser till Bibliotek, d.v.s. vilken kombination som helst av sökvillkor, så som den kodas i adressraden. De ska alltid inledas med frågetecken, ’?’.

”Populära Serier” och ”Populära Utgivare” är förpreparerade listor på serier respektive utgivare som vi bedömer vara intressanta. (Det är alltså inget automatiserat mått, utan vår manuella bedömning.)

4.3 Biblioteksposterna

Varje biblioteksposts uttrycks med ett Json-objekt, alltså en uppsättning av nyckel/värde-par. De olika typerna av värden som lagras är här indelade i:

- **Dokumentlänk:** en textsträng som utgör nyckel till en annan dokument-post i den synkroniserade databasen, jfr kapitel 2.1.
- **Heltal:** ett positivt heltal; den närmare beskrivningen anger hur stort talet kan vara (eller hur mycket plats som mest behövs för att lagra det).

- **Referens:** en referens till *utgivare*, *serie, mediatyp* eller *ämneshylla*. För utgivare, serie och mediatyp används heltal, för ämneshylla används en sträng (heltal och bindestreck). Jfr kapitel 4.1.
- **Text:** vanlig textsträng.
- **URL:** en webblänk.
- Symbolen **[]** indikerar att det är en array, som alltså kan innehålla flera värden (t.ex. flera författarnamn).

I nedanstående tabell redovisas vilka nyckelnamn som är definierade:

Nyckel	Obligatorisk	Typ av värde	Beskrivning
dokid	X	heltal	Biblioteksposten unika id (kräver 16 bitars teckenlös lagring).
skydd		heltal	0 (eller saknat värde): publik post, 1: partiellt skyddad post, 2: helt skyddad post Se kapitel 3.16.
ftix		dokument-länk	Referens till bibliotekspostens fulltextindex-post.
omsl	X	dokument-länk	Referens till bibliotekspostens omslagsbild (vilken kan vara gemensam för många poster), se kapitel 3.7.
media	*	dokument-länk	Referens till bibliotekspostens mediafil, se kapitel 3.13.
mediaURL	*	URL	Webblänk till bibliotekspostens mediafil, eller till en betalvägg eller motsvarande bakom vilken resursen befinner sig (aktuellt för 90 sekunder-filmer).
mediatyp	X	referens	Mediatyp, givet som referenstal, se kapitel 4.1.
datatyp	X	heltal	0: fysisk bibliotekspost utan digital media, 1: fysisk bibliotekspost med sammanfattnings-pdf, 2: vanligt dokument (default), 3: film (ev. som webblänk), 4: länk till webbsida
Titel	X	text	Biblioteksposten titel.
Forf		text[]	Förteckning över verkets författare.
Omf		text	Textfältet ”omfång”.

UtgTid		heltal	Utgivningstidpunkt i ett av tre format: - Utgivningsår (fyra siffror, ÅÅÅÅ) - Utgivningsår och -månad (sex siffor, ÅÅÅÅMM) - Utgivningsdag (åtta siffror, ÅÅÅÅMMDD).
DateCreated		heltal	Datum då posten registrerats i vår databas, som ÅÅÅÅMMDD.
Utg		referens[]	Förteckning över utgivare, givna som referenstal, se kapitel 4.1.
Forl		text	Utgivande förlag. (En publikation kan ha flera utgivare, men bara ett förlag.)
Serie		referens	Eventuell serie som verket ingår i, givet som referenstal, se kapitel 4.1.
UtgSerie		text	Av utgivaren tilldelat nummer enligt deras eget nummersystem, t.ex. ”P30-194/97” för en specifik bok från Räddningsverket. Kan innehålla mellanslag.
ISSN		text	ISSN givet som åtta ”siffror”, där den sista positionen kan vara bokstaven X. (Vid presentation ska de grupperas fyra-bindestreck-fyra.)
ISBN		text[]	ISBN, ett eller två (d.v.s. både gamla formatet med 10 siffror och nya formatet med 13 siffror). Oftast med bindestreck angivna, men inte alltid.
Spr		text	Språkkod med tre bokstäver, eller ”mul” för flerspråkiga verk.
NyckO	X	text[]	Biblioteksposten ämnesord (från en kontrollerad vokabulär), vilket tidigare också kallats nyckelord.
Hylla	X	referens[]	Referens till en eller flera ämneshyllor, givet som siffror med bindestreck, se kapitel 4.1.
Samfatt	X	text	Sammanfattningstext. (Nytt stycke är kodat som två nyradstecken efter varandra, ”\n\n”.)
Anm		text	Anmärkningstext, kan användas för varierande typer av anmärkningar.
EjDigital		-	Värdet *true* anger att biblioteksposten ursprungligen saknar digital media, även om det förvisso kan finnas en pdf-fil, som i så fall bara är en sammanfattning. Om digital media finns är värdet utelämnat.

*) För normala poster ska ett av fälten *media* och *mediaURL* ingå (men inte båda samtidigt). Dock finns även poster som enbart refererar till fysiska böcker i Myndigheten för civilt försvars boksamling. För de är inget av dessa båda fält angivet.

I bilaga A återfinns ett exempel på en komplett bibliotekspost.

5. Json-struktur i Farliga ämnen

Detaljnivån – Kapitlet beskriver ämnesposternas Json-struktur, och de olika datatyper som den består av. (Hur denna Json-struktur lagras i dokumentdatabasen förklarades i kapitel 3.9.)

5.1 Översiktlig struktur

Nedanstående strukturbeskrivning avser hur datamängden för en (1) ämnespost uttrycks i Json, inklusive hänvisningarna till sektionsspecifika HTML-mallar.

När ämnesposterna visas för användaren är de tematiskt indelade i sektioner (”flikar”) vilket dock inte påverkar ämnespostens datastruktur, mer än att varje sektion behöver en hänvisning till sin sektionsspecifika HTML-mall.

Datastrukturen är i huvudsak generisk. De generiska variablerna har en av tre tillgängliga datatyper: *värde*, *array* eller *nyckelförsedd array*. För att hantera dels **synonymer** och dels **hygieniska gränsvärden** används dock specifika strukturer. Tanken med att använda så mycket generiska variabler som möjligt är att vi i ett senare skede ska kunna lägga till en ny variabel utan att behöva göra någon ändring i applikationen (så länge någon av de tre generiska datatyperna används).

Ämnespostens grova struktur är följande:

```
{
     "ID": …,
     "Templates": { … },
     "data" : { … }
}
```

Under ”data” döljer sig både generiska ämnesdata och de två specifika strukturerna för synonymer respektive hygieniska gränsvärden.

I bilaga C redovisas ett verkligt exempel på en komplett ämnespost: Ammoniak, vattenfri (UN 1005).

5.2 Unik nyckel (”ID”)

Ämnespostens ämnes-id är utbruten ur den generiska strukturen, eftersom den utgör ämnestabellens unika nyckel. Ämnes-id är ett heltal, just nu 1~10 000 men det kan förstås växa uppåt i rimliga proportioner. Jfr kapitel 3.9.

5.3 Referenser till sektionsmallarna (”Templates”)

Ämnespostens referenser till HTML-mallarna är också utbruten ur den generiska datastrukturen. Referensobjektet ser ut på detta sätt:

```
"Templates": {
     "Identitet":"…",
     "Raddning":"…",
     "Akutvard":"…",
     "Fysdata":"…",
```

```
    "Miljo":"…"
}
```

Värdena som anges är mallarnas namn utan prefix. För Akutvård kan värdet t.ex. vara `"sirib45.htm"`, vilket betyder att den ifrågavarande mallen lagras i dokument-databasen under nyckeln ”tpl/sirib45.htm”.

På sikt är det möjligt att fler sektioner (”flikar”) tillförs enligt samma mönster.

5.4 Ämnespostens egentliga data (”data”)

5.4.1 Data med datatypen *värde*

Datatypen *värde* omfattar data som uttrycks med ett enstaka värde. Värdet kan vara ett heltal, ett flyttal eller en sträng. De flesta fält kan också vara tomma, vilket uttrycks som en tom sträng. Några av fälten kan uteslutas helt, vilket anges särskilt nedan.

Bokstäverna som förekommer i kolumnen Anmärkning förklaras direkt efter tabellen.

Parameternamn	**Anm.**	**Beskrivning**
Namn		Ämnespostens huvudnamn. Fältet har alltid ett värde.
Fortydligande		Ämnespostens förtydligandetext. Förtydligar vad som menas med ämnespostens namn. Huvudsyftet är att hjälpa användaren att välja rätt ämnespost.
LosningAvId		Om ämnesposten är en lösning av en annan ämnespost anges det lösta ämnets id här. I annat fall är fältet tomt.
Beskrivning		Koncentrerad beskrivning av ämnets egenskaper grundat på dess transportklassificering.
FarlNr		Farlighetsnummer enligt ADR eller RID. Kan utöver siffror innehålla bokstav och punkt.
FarlNrTxt		Textförklaring av vad farlighetsnumret betyder. (Fältet kan i teorin vara tomt även om FarlNr är angivet.)
FarlNrADR		Farlighetsnummer enligt ADR. Kan utöver siffror innehålla bokstaven X. (Detta värde är för flera ämnesposter tomt även om *FarlNr* är angivet.)
FgADR		Anger för respektive regelverk (ADR, RID, IMDG, IATA-DGR) om ämnesposten omfattas av reglerna: 1: Ej bedömt 2: Omfattas ej av regelverket 3: Farligt gods 4: Transport ej tillåten
FgRID		
FgIMDG		
FgIATADGR		

Klass		Klass enligt ADR.[5]
Klassif		Klassificeringskod enligt ADR.[6]
Fpg		Förpackningsgrupp enligt ADR.[7]
UN	A	Om ämnesposten motsvarar en rad i transport-regelverket anges UN-numret här, t.ex. ”0034”. I annat fall är fältet utelämnat.
UNutg	A	Om ämnesposten motsvarar ett UN-nummer som tidigare fanns i transportregelverket, men som tagits bort, anges numret här, t.ex. ”3050”. I annat fall är fältet utelämnat.
Anm		Anmärkning om ämnesposten. Ibland är det ett förtydligande till någon uppgift, ibland är det mer allmän information.
AggTill		Aggregationstillstånd.
Farg		Färg.
Lukt		Lukt.
SmaltP	B	Smältpunkt.
KokP	B	Kokpunkt.
BrbOmr		Brännbarhetsområde, t.ex. ”från 15 till 28 vol-%” eller ”Oxiderande (brandunderstödjande)”.
FlamP	B	Flampunkt.
TermTP	B	Termisk tändpunkt.
Dens		Densitet. Avser flytande eller fast form av ämnet.
DensTal		Densitetstal, det vill säga hur tunga gaser eller ångor från ämnet är i förhållande till vanlig luft, som har densitetstal 1,0.
Visk		Viskositet (kinematisk viskositet).
VtnLosl		Vattenlöslighet. Anges oftast både med ord och med siffror (viktprocent).
Molekyl		Molekylformel i textformat. Klientapplikationen bör snygga till den innan den visas för användaren. Se bilaga D.
MolVikt		Molekylvikt i g/mol.

[5] Alla ämnesposter har likalydande ADR-klass och RID-klass.
[6] Klassificeringskoden i RID är densamma som klassificeringskoden i ADR.
[7] Det finns två ämnesposter där förpackningsgruppen skiljer sig åt mellan ADR och RID: UN 3533 och 3534 saknar förpackningsgrupp i RID. Det beror på att järnvägstransport av dessa ämnen är helt förbjuden.

JonPot		Jonisationspotential.
KritT		Kritisk temperatur i grader Celsius.
AngtryckA		Konstanterna A [N/m²] och B [K] i formeln som beskriver ämnets ångtryckskurva: $$P_S(T) = A \cdot e^{\frac{-B}{T}}$$ P_S är ångtrycket [Pa] vid given temperatur T [K].
AngtryckB		
AngtryckPkt	B	Ångtrycksangivelse som punktvärde (så som det är inmatat i databasen).
Riskomrade		Rekommenderat initialt riskområde. Om värdet är tomt beror det antingen på att rekommendation saknas, eller på att en riskområdestabell visas från HTML-mallen i stället (jfr kapitel 6.2.2).
RiskomradeCSV		Fältet planeras att införas i framtiden som en maskinläsbar riskområdesangivelse, som kan ligga till grund för en visualisering på karta.
Skyddsutrustning		Rekommenderad skyddsutrustning vid läckageplats.
SkyddLivraddning		Rekommenderad skyddsutrustning vid livräddning.
Slackmedel		Släckmedelsrekommendation.
AvtalKemsamverkan		Om ämnet omfattas av avtalet om Företagens Kemberedskap anges ämnets namn här. I annat fall är fältet utelämnat.
GhsSignord		Signalord enligt harmoniserad CLP-märkning. Antingen `FARA` eller `VARNING`. Fältet kan vara tomt.
GhsFinns		Om ämnet har information om harmoniserad CLP-märkning anges `X` (som representerar ”sant”). Annars tom sträng.
GhsAnm		Eventuell anmärkningstext rörande informationen om harmoniserad CLP-märkning; CLP-informationen kan bli missvisande om inte anmärkningen visas. Fältet kan vara tomt. Exempelämne: UN 3318.
SpridningLuft		Om ämnesposten kan hanteras av Spridning Luft anges en startparameter som kan skickas till programmet för att starta med rätt ämne. I annat fall är fältet utelämnat.
SpridningMark		Om ämnesposten kan hanteras av Spridning Mark anges en startparameter som kan skickas till programmet för att starta med rätt ämne. I annat fall är fältet utelämnat.

A) Jämför **UNovr** under datatypen *array*. Endast ett av **UN**, **UNutg** och **UNovr** kan ha värde satt.

B) Vissa fysikaliska uppgifter kan ges med mindre än- eller större än-tecken (<, >). I Json-strukturen är de lagrade just så, även om det innebär att de måste kodas om till `<` respektive `>` om de ska gå att använda i HTML-kod.

5.4.2 Data med datatypen *array*

Datatypen *array* omfattar de data som kan uttryckas med en sträng-array (inklusive specialfallet att strängarna utgör filnamn till bilder). Skillnaden mot enkla värden är att i arrayer kan det vara flera värden för en och samma parameter. Värdena är strängar, men fält kan också utelämnas helt om värde saknas.

Parameternamn	Anm.	Beskrivning
UNovr	A	Om ämnesposten inte motsvarar någon rad i transportregelverket, men ändå måste transporteras under samlingsbenämning, anges vilken eller vilka samlingsbenämningar som kan komma ifråga som UN-nummer här, t.ex. [”2780”] eller [”2779”, ”2780”, ”3013”]. I annat fall är fältet utelämnat.
CAS		Ämnespostens samtliga CAS-nummer[8], t.ex. [”64742-47-8”, ”68334-30-5”, ”68476-30-2”]. Fältet utelämnas om CAS-nummer saknas.
EG		Ämnespostens samtliga EG-nummer, t.ex. [”265-149-8”, ”269-822-7”, ”270-671-4”]. Fältet utelämnas om EG-nummer saknas.
Reaktioner		Ämnesspecifika riskfaktortexter (reaktionstexter). Fältet utelämnas om texter saknas.
AkutvardInfo		Ämnesspecifika akutvårdstexter.[9] Fältet utelämnas om texter saknas.
TEtiketter		Filnamn på transportetiketter för etikettering av kollin enligt ADR och RID. Fältet utelämnas om transportetiketter saknas.
GhsPiktogram		Filnamn på faropiktogram enligt harmoniserad CLP-märkning. Fältet utelämnas om faropiktogram saknas.
Anvomr		Ämnets användningsområden. Fältet utelämnas om texter saknas.

A) Jämför **UN** och **UNutg** under datatypen *values*. Endast ett av **UN**, **UNutg** och **UNovr** kan ha värde satt.

[8] (Intern kommentar: detta omfattar både nummertyp *CAS* och *CAS_X* i vår administrationsdatabas.)
[9] (Intern kommentar: egenskaper med kryssrutan ”sjukvård” ikryssad i vårt administrationssystem.)

5.4.3 Data med datatypen *nyckelförsedd array*

Datatypen *nyckelförsedd array* är en utökad array. Varje element i arrayen består av ett beskrivning-värde-par. Man kan betrakta det som en enkel tabell med två kolumner: den vänstra kolumnen är en slags beskrivning eller kod, och den högra kolumnen är ett värde eller en uttydning av koden.

I Json representerar vi detta som en array av objekt, där varje tabellrad utgör ett objekt:

```
"ToxTolkning":
[
     { "Akut giftighet": 1 },
     { "Anrikas i naturen": 3 },
     { "Cancerframkallande": 2 },
     { "Långsiktiga skador på organ/nervsystem": 2 }
]
```

I beskrivnings-exemplen nedan ges (av utrymmesskäl) bara en tabellrad, även om tabellerna ofta har fler rader än så.

Parameternamn	Anm.	Beskrivning
ToxTolkning		Tabellen ”Tolkning av toxikologiska data” på Miljö-fliken. 1: ”Ja” 2: ”Nej” 3: ”Okänt” `[ {"Akut giftighet": 1 } ]`
ToxData	B	Tabellen ”Toxikologiska data” på Miljö-fliken. `[ {"Giftighet vid inandning: LC50 inhalerat mus, 1 h": "> 0,4 mg/l" } ]`
GhsFaror		Faroangivelser enligt harmoniserad CLP-märkning. `[ {"H400": "Mycket giftigt för vattenlevande organismer." } ]`

B) Vissa fysikaliska uppgifter kan ges med mindre än- eller större än-tecken (<, >). I Json-strukturen är de lagrade just så, även om det innebär att de måste kodas om till `<` respektive `>` om de ska gå att använda i HTML-kod.

5.4.4 Specifik struktur för synonymer

Ämnespostens information om synonymer är tyvärr för komplicerad för att kunna redovisa med en generisk datastruktur. Därför används i stället följande specifika struktur: en array med synonym-objekt.

Den obligatoriska beståndsdelen i synonym-objektet är ett nyckel-värde-par med språkkod som nyckel och själva synonymen som värde. Utöver detta kan även en typ-angivelse ges: 1=ADR, 2=Handelsnamn, 3=ADR och Handelsnamn.

Språkkoderna väljs enligt ISO 639-1, vilket i vårt fall blir dessa fyra: en, de, fr, sv.

Ett enkelt exempel visar hur det skulle kunna se ut:

```
"Synonymer":
[
    {"en": "Ammonia"},
    {"de": "Ammoniak, wasserfrei", "typ": 1},
    {"sv": "R 717", "typ": 2}
]
```

5.4.5 Specifik struktur för hygieniska gränsvärden

Ämnespostens information om hygieniska gränsvärden och lukttrösklar hanteras i ett specialdesignat objekt. Objektet består av upp till 20 nyckel-värde-par. Många av värdena gäller för en särskild exponeringstid, och då anges tiden i minuter som ett suffix i nyckelnamnet (efter '_').

Studera gärna hur tabellen som presenterar hygieniska gränsvärden ser ut i RIB Farliga ämnen, men använd en tillräckligt bred skärm för att se den fullständiga tabellen. (På telefoner kommer appen Farliga ämnen att visa en omgjord layout.)

För närmare förklaring om de olika gränsvärdena hänvisas till den inbyggda hjälpen i RIB Farliga ämnen.

Följande nycklar är definierade för de tre raderna "Risk för …". Dock är det bara en av grupperna *a–c* som används i varje enskilt fall:

a) AEGL1_10, AEGL1_30, AEGL1_60, AEGL1_240, AEGL1_480, AEGL2_10, AEGL2_30, AEGL2_60, AEGL2_240, AEGL2_480, AEGL3_10, AEGL3_30, AEGL3_60, AEGL3_240, AEGL3_480

b) ERPG1_60, ERPG2_60, ERPG3_60

c) TEEL1_15, TEEL2_15, TEEL3_15

För raden "Arbetsmiljö" används dessa nycklar; maximalt kombineras en KGV-variant med NGV-värdet:

- KGV(V)_5, KGV(V)_15, KGV_5, KGV_15, NGV_480

Notera skillnaden på KGV = bindande korttidsgränsvärde som inte får överskridas, och KGV(V) = vägledande korttidsgränsvärde som är ett rekommenderat högsta värde.

De återstående raderna "Filtermask ej lämplig" (= IDLH), "Uttalad lukt" och "Förnimbarhet" – som alla saknar tidsangivelse – har dessa nycklar:

- IDLH, Lukt, Fornim

Värdena anges alltid i ppm som siffervärden (inte som strängar). När de presenteras för användaren skrivs däremot alltid enheten "ppm" ut.

Ett enkelt exempel hämtat från kolmonoxid visar hur det kan se ut:

```
"Gransvarden":
{
    "AEGL3_10": 1700,  "AEGL3_30": 600,  "AEGL3_60": 330,
    "AEGL3_240": 150,  "AEGL3_480": 130,  "AEGL2_10": 420,
    "AEGL2_30": 150,  "AEGL2_60": 83,  "AEGL2_240": 33,
    "AEGL2_480": 27,  "KGV_15": 100,  "NGV_480": 20,
    "IDLH": 1200
}
```

6. Malldokumenten i Farliga ämnen

Detaljnivån – Kapitlet ger fördjupad information om hur malldokumenten i Farliga ämnen-databasen fungerar.

6.1 Varför malldokument?

RIB Farliga ämnen innehåller olika typer av information som inte kan presenteras på ett bra sätt i ren textform:

- Orange farligt gods-skylt och transportmärkning (etiketter) för farligt gods, samt faropiktogram enligt CLP.
- Åtgärdsschema för räddningstjänsten, som kan grena upp sig beroende på om det är brand eller läckage.
- Interaktiv ångtryckskurva med möjlighet att välja ämnets temperatur, och då få temperaturspecifika uträkningar av ångtryck och mättnads-koncentration.
- För vissa ämnen behöver det rekommenderade initiala riskområdet ges som en tabell. (Men det vanligaste är att det består av en enkel text.)

Dessutom finns längre, mer resonerande, texter på vissa av innehållsflikarna,[10] och de texterna behöver kunna ha formatering som t.ex. fet stil eller hyperlänkar till webben.

Eftersom vi vill kunna visa innehållet i Farliga ämnen-databasen på flera olika plattformar, med så lite dubbelarbete som möjligt, har vi valt att använda HTML med CSS och Javascript som presentationsteknik. Utifrån ett utvecklarperspektiv är det en enkel textbaserad lösning, men den är ändå kraftfull nog att hantera de olika informationstyperna vi behöver hantera (enligt ovan). HTML fungerar rakt av på webben, och det kan läsas in i webbvyer i såväl Windows-applikationer som mobilappar.

För att kunna presentera med HTML behöver HTML-koden genereras på något sätt. Enklaste sättet att göra det är genom att definiera någon slags skelett som vår programvara får fylla på med anpassat innehåll för det aktuella tillfället. Mall-dokumenten utgör just detta skelett. Som vi ska se i nästa avsnitt kan skelettet vara mer eller mindre innehållsrikt.

[10] I nuläget är det flikarna Räddning och Akutvård som inkluderar längre texter på detta sätt.

6.2 Malldokumentens tredubbla funktion

Alla malldokument fungerar som en strukturell grund för att definiera vilka uppgifter som ska visas var. Men vissa malldokument innehåller också faktatexter i sig själva. Utöver detta är dessutom hjälptexter inbäddade i malldokumenten.

6.2.1 Funktion som strukturell grund

Utifrån resonemanget i 6.1 ser vi att ett viktigt syfte med malldokumenten är att tjäna som strukturell grund för den slutliga HTML-sida som ska presentera information om ett valt ämne. Malldokumentet ”tpl/fysdata.htm” fungerar på just detta sätt (här återgivet i förkortad och förenklad form, för att belysa logiken bättre):

```
<h2>Fysikaliska data</h2>
  {% if Anm %}
    <section class="anmarkning">
       <h3>Anmärkning</h3>
       <p>{{ Anm }}</p>
    </section>
  {% endif %}

  <h3>Allmänt</h3>
  <section>
    <h4>Tillstånd:</h4><p><span data-rib="AggTill">{{ AggTill }}</span></p>
    <h4>Färg:</h4><p><span data-rib="Farg">{{ Farg }}</span></p>
    <h4>Lukt:</h4><p><span data-rib="Lukt">{{ Lukt }}</span></p>
    <h4>Smältpunkt:</h4><p><span data-rib="SmaltP">{{ SmaltP }}</span></p>
     <!-- ... -->
  </section>

  <h2>Förångnings&shy;egenskaper</h2>
  <section>
    {% if AngtryckA and AngtryckB %}
       <!-- ... -->
    {% endif %}
  </section>

  <h2>Gränsvärden</h2>
  <section>
    {% assign AeglErpgTeel = nil %}
    {% assign Col_5 = GRV_KGVV_5 or GRV_KGV_5 %}
    {% assign Col_10 = false %}
    <!-- ... -->
  </section>
  <!-- ... -->

<script>
  /* ... */
</script>
```

Vi kan se att sidan inleds med en H2-rubrik ”Fysikaliska data”. Därefter visas eventuellt en anmärkningstext, och sedan kommer första delsektionen ”Allmänt”. Den börjar med rubriken ”Tillstånd”, och där visas ämnets aggregationstillstånd (som enligt 5.4.1 har parameternamnet *AggTill*). Och så fortsätter det. Sist på sidan finns det javascript som ritar ut ångtryckskurvan och som låter användaren välja ämnets temperatur.

Malldokumenten använder *Liquid Templating Language* för att bädda in logiska villkor och referera till ämnespostens datavärden. Vi återkommer till det i kapitel 6.3.

6.2.2 Funktion som faktaartikel

För flikarna Räddning och Akutvård fungerar malldokumenten inte bara som strukturell grund, utan även som en gemensam faktagrund för en specifik ämnesgrupp. Sålunda finns, med ungefärliga siffor, 50 malldokument för Räddning och 40 malldokument för Akutvård. Samma teknik kan mycket väl komma att användas för fler flikar i framtiden, t.ex. fliken Miljö.

Malldokumenten för Räddning är indelade i kategorier baserat på ämnenas transportklassificering (t.ex. ”Brandfarliga gaser”, ”Giftiga ämnen”, ”Självantändande ämnen”), medan malldokumenten för Akutvård är indelade efter deras förgiftningsegenskaper (t.ex. ”Aromatiska föreningar”, ”Retande gaser”, ”Sexvärda kromföreningar”).

För dessa två flikar utgör faktatexterna i mallarna sidans huvudsakliga innehåll. Mallarnas lite mer övergripande faktatexter kompletteras med ämnesspecifika detaljer (med hjälp Liquid Templating Language, precis som ovan).

Som exempel tittar vi närmare på ett utdrag ur mallen ”tpl/r_f.htm” som avser fliken Räddning för brandfarliga ämnen (förenklat för att bli tydligare):

```
<h3>Initial information</h3>
<section>
  <h4>Ämnesbeskrivning:</h4><p>{{ Beskrivning }}</p>
  <h4>Initialt riskområde:</h4><p>{{ Riskomrade }}</p>
  <h4>Skydd – livräddning:</h4><p>{{ SkyddLivraddning }}</p>
  <h4>Skydd – läckageplats:</h4><p>{{ Skyddsutrustning }}</p>
  <h4>Släckmedel:</h4><p>{{ Slackmedel }}</p>
</section>

<h2>Riskfaktorer – Brandfarliga ämnen</h2>
<section>
  <p>Brandrisk. En brand kan sprida sig snabbt. Risk för återantändning
  efter släckt brand.</p>
  <p>För <strong>vätskor</strong> gäller att brännbara ångor kan sprida sig,
  och ge explosionsrisk vid läckage inomhus eller i avlopp. Ångorna är i
  många fall tyngre än luften (kontrollera på Fysdata-fliken!) och kan då
  samlas i lågt belägna utrymmen, t.ex. källare. [...]</p>
  <!-- ... -->

{% if Reaktioner %}
  <h3>Särskilt för <em>{{ Namn }}:</em></h3>
    <ul>
      {% for text in Reaktioner %}
        <li>{{ text }}</li>
      {% endfor %}
    </ul>
{% endif %}
```

Lägg särskilt märke till:

- Texten under **Riskfaktorer** börjar med en flera stycken lång text som står direkt i mallen, men kompletteras sedan med en uppräkning av ämnespostens specifika riskfaktortexter (reaktionstexter), om sådana finns.

- På just fliken Räddning påverkar valet av malldokument även om **Initialt riskområde** ska visas som en text från ämnesposten (med parameternamn *Riskomrade*, som i exempelkoden), eller om det ska visas som en tabell – vilken i så fall är kodad direkt i malldokumentet.

6.2.3 Funktion som bärare av hjälptexter

Malldokumenten fungerar dessutom som bärare av hjälptexter. I gränssnittets högerkant visas en liten i-symbol som användaren kan klicka på, och då visas en hjälptext för aktuellt avsnitt.

Hjälpsystemet är i huvudsak statiskt, baserat enbart på HTML, men på några ställen har det varit nödvändigt att bygga in en dynamisk hantering, som då görs med Javascript.

Referenserna till det inbyggda hjälpsystemet har tvättats bort från kodexemplen i 6.2.1 och 6.2.2 för att inte ta fokus från det som exemplen vill förmedla.

6.3 Hur ämnespostens data görs tillgänglig för malldokumenten

Vi använder templating-motorn *Fluid.Core* för att göra om malldokumenten från mallar skrivna i *Liquid Templating Language* till färdig HTML-kod.[11]

6.3.1 Mappning av datatyper

De olika datavärdena tillgängliggörs för templating-motorn under samma namn som de definieras med i Json-strukturen. Detta sker på ett generiskt sätt, så att strukturen kan byggas ut med fler värden utan att någon omprogrammering behöver göras:

- Datatypen *värde* tillgängliggörs som Liquid-datatypen ”number” om det är möjligt utan att innehållet förvanskas, annars som ”string”.
- Datatypen *array* tillgängliggörs som Liquid-datatypen ”array”.
- Datatypen *nyckelförsedd array* tillgängliggörs som Liquid-datatypen ”array av array”, där nyckeldelen av varje rad har index [0] och värdedelen av varje rad har index [1].

För att förenkla testvillkoren i Liquid-koden tilldelas dock inga tomma värden: om Json-strukturen innehåller ett tomt värde, så kommer motsvarande variabel i Liquid *inte* att tilldelas.[12] Detta medför att villkoret `{% if variabelnamn %}` kan användas för att testa om en variabel är definierad med värde eller inte. (I annat fall hade varje sådant test behövt kontrollera både variabelns existens och om den är tom.)

---

[11] Se https://github.com/sebastienros/fluid för mer information om Fluid.Core, och https://shopify.github.io/liquid/ för mer information om själva skriptspråket.
[12] Även specialvärdet `null` i Json behandlas på detta sätt, d.v.s. variabeln tilldelas inte i Liquid.

De två specialstrukturerna för synonymer och hygieniska gränsvärden hanteras så här:

- **Synonymer** hanteras med ett generiskt stöd för tabeller med fyra kolumner, på ett liknande sätt som *nyckelförsedd array*, men med indexering från [0] till [3] för de olika kolumnerna. (Detta medför att kolumn [2] blir ointressant: antingen finns den inte, eller så innehåller den texten ”typ”.)
- **Hygieniska gränsvärden** tillgängliggörs som enskilda värden med namn GRV_*xx*, så att de på ett tydligt sätt kan tas omhand av logiken i Liquid-koden. Dock kan inte parentes användas i variabelnamnet, så gränsvärdena KGV(V)_5 och KGV(V)_15 får variabelnamnen GRV_KGVV_5 och GRV_KGVV_15. Variabelvärdena kompletteras dessutom med enheten ”ppm” (som ju inte ingår i Json-strukturen).

6.3.2 Särskilda filter

Templating-motorn Fluid.Core ger möjlighet att definiera egna filter. Vi använder följande specialbyggda filter:

- **imageurl** – Hämtar base64-kodad bild som data-URL för en given bildfil.[13] Jfr kapitel 3.12.
- **imagetext** – Hämtar beskrivningstext för en given bildfil (metadata för *alt*- och *title*-taggarna). Jfr kapitel 3.12.
- **molecule** – Gör om en ren textformel till HTML-kodad textformel med `<sub>`-taggar för siffror som ska vara nedsänkta etc. Se bilaga D.

6.4 Komplettering med HTML-header

Den HTML-kod som genereras när malldokumenten körs genom templating-motorn Fluid.Core är det som ska stå innanför HTML-sidans `<body>`-taggar. Koden för varje sida behöver alltså kompletteras med sidhuvud och sidfot, samt med CSS. (Om en tredjepartsutvecklare vill göra på detta sätt bör både ”css/app.css” och ”css/hazmat.css” inkluderas.)

[13] Vi använder alltså data-URL för att infoga transport- och CLP-etiketter i HTML-sidan, i stället för referenser till fysiska filer. Jfr https://css-tricks.com/data-uris/.

Bilaga A – Exempel på komplett bibliotekspost

Här återges ett exempel på en komplett bibliotekspost: ”Insatskort för acetylengasflaskor”, dokid 28223. Biblioteksposten lagras i den synkroniserade Bibliotek-databasen under nyckelnamnet ”doc/28223”.


```
{
  "dokid": 28223,
  "ftix": "ftix/28223",
  "omsl": "cov/28223.png",
  "media": "media/28223.pdf",
  "mediatyp": 15,
  "datatyp": 2,
  "Titel": "Insatskort för acetylengasflaskor",
  "Forf": [ "Egardt Erik", "Levein Magnus" ],
  "Omf": "4 s. : ill.",
  "UtgTid": 2017,
  "DateCreated": 20170116,
  "Utg": [ 1294 ],
  "Forl": "Myndigheten för samhällsskydd och beredskap",
  "Serie": 587,
  "UtgSerie": "MSB1074 - januari 2017",
  "Spr": "Swe",
  "NyckO": [ "Räddningstjänst", "Gaser", "Räddningskemi", "Acetylen" ],
  "Hylla": [ "3-6-9" ],
  "Samfatt": "Acetylen, eller etyn som den också heter, är en brandfarlig,
färglös gas. Vid förbränning med ren syrgas fås en flamma på över 3 000 °C som
lämpar sig väl för svetsning, skärning eller hårdlödning. Acetylen kan även
sönderfalla utan tillgång till syre, och bildar då kol (sot) och vätgas. Om
acetylengasflaskan blir utsatt för en temperatur över 300 °C startar
sönderfallet, med tryck och värmestegring som följd. Detta kan leda till att
flaskan exploderar även efter att värmepåverkan upphört. Acetylengasflaskor
behandlas därför annorlunda vid brand än andra gasflaskor.",
  "Anm": ""
}
```

Bilaga B – Språkbehandling i indexeringsalgoritmen

I kapitel 3.10 beskrivs hur Bibliotek använder färdigpreparerade fulltextindex-poster, hanterade som dokumenttyp ”ftix” i synkroniseringsdatabasen. I denna bilaga beskrivs kort vilken metod vi använder vid framställningen av dessa indexposter.

Precis som beskrevs i kapitel 1.4 använder vi Lucene.Net som sökmotor, och indexeringsprocessen bygger till stor del på Lucenes funktionalitet.

Bibliotek är en flerspråkig databas. Största språken är förstås svenska (75 %) och engelska (15 %). Men det finns flera andra språk också, där norska och tyska är störst med vardera knappt 1 %.[14]

Den specifikt språkkänsliga delen av indexeringen är trunkeringen med stemmer (som alltså görs för att böjningsvarianter av ord ska indexeras lika). Eftersom det är en så pass stor andel engelska texter, har vi valt att köra både svensk och engelsk stemming-algoritm – och för varje ord välja den variant som blev kortast. Att göra på detta sätt innebär att sökfrågan kan trunkeras på samma sätt, vilket ju är en förutsättning för att sökning alls ska vara möjlig.

Även stoppordlistan, som används för att exkludera vanliga ord från sökningen, är tvåspråkig och inkluderar både svenska och engelska ord.

Vi använder stemmern Annytab som har stöd för tolv språk. I vårt fall stemmar vi orden på svenska och engelska för att sedan välja det kortaste.

Dokumenttexten, hämtad från pdf-fil, förbereds på följande sätt innan ftix-filen skapas:

- Överflödiga mellanslag och mjuka radbryt rensas.
- Alla bokstäver görs om till gemener.
- Allt förutom latinska bokstäver, siffror, bindestreck, snedstreck samt lika-med-tecken rensas.
- Bindestreck rensas i ord som inleds och avslutas med siffra, där den avslutande siffran kan vara bokstaven X. (Avsikten med detta är att underlätta sökningar på ISBN- och ISSN-nummer.)
- Snedstreck ersätts med mellanslag för att separera exempelvis ”risk/säkerhet”.
- Inledande och avslutande bindestreck rensas för att snygga till ord som ”risk- och miljöavdelning” samt ”-totaldemokrati”.
- Tecknet lika-med ersätts med mellanslag.
- Rensar bort alla stoppord och alla korta ord (ord med mindre än tre bokstäver).

[14] Procentangivelserna är något osäkra, men det saknar praktisk betydelse för vårt resonemang.

- Ersätter norska och danska tecken (æ, ø) med den svenska motsvarigheten (ä, ö), samt tyska ”ß” med ”ss”.

Stoppordslistan gjordes kortare i och med lanseringen av API-version 3. Den ser nu ut som följer:

```
"att", "av", "blev", "blir", "de", "dem", "den", "denna", "deras", "det",
"detta", "dig", "din", "dina", "ditt", "där", "då", "eller", "en", "er",
"era", "ert", "ett", "från", "för", "ha", "hade", "har", "här", "i", "in",
"kan", "kunde", "man", "med", "men", "mig", "min", "mina", "mitt", "ner",
"ni", "när", "och", "om", "oss", "på", "sig", "sin", "sina", "sitt", "ska",
"skulle", "som", "så", "till", "under", "upp", "ut", "var", "varit", "vi",
"vid", "vår", "våra", "vårt", "är", "över",


"a", "an", "and", "as", "at", "be", "been", "being", "but", "by", "did", "do",
"does", "for", "from", "had", "has", "have", "he", "here", "in", "is", "it",
"of", "on", "or", "our", "she", "that", "the", "their", "them", "there",
"these", "they", "this", "those", "to", "was", "we", "were", "when", "where",
"will", "with", "you", "your"
```

Bilaga C – Exempel på komplett ämnespost

Här återges ett verkligt exempel på en komplett ämnespost: Ammoniak, vattenfri (UN 1005). Internt ämnes-id är 448 och ämnesposten lagras i den synkroniserade Farliga ämnen-databasen under nyckelnamnet ”doc/448”.

```
{
  "ID": 448,
  "Templates": {
    "Identitet": "id.htm",
    "Raddning": "r2t-m.htm",
    "Akutvard": "sirib45.htm",
    "Fysdata": "fysdata.htm",
    "Miljo": "miljo.htm"
  },
  "data": {
    "CAS": [
      "7664-41-7"
    ],
    "EG": [
      "231-635-3"
    ],
    "Reaktioner": [
      "Löser sig i vatten och bildar en basisk vattenlösning.",
      "Reagerar häftigt med oxidationsmedel.",
      "Reagerar häftigt med syror.",
      "Vid brand/upphettning bildas nitrösa gaser.",
      "Vid läckage inomhus kan brännbara koncentrationer uppnås."
    ],
    "AkutvardInfo": [
      "Verkar även frätande.",
      "Vid brand/upphettning bildas nitrösa gaser."
    ],
    "TEtiketter": [
      "etk_2_3.png",
      "etk_8.png",
      "etk_13_klass2.png"
    ],
    "GhsPiktogram": [
      "ghs04_bottle.png",
      "ghs05_acid.png",
      "ghs06_skull.png",
      "ghs09_aquaticpollut.png"
    ],
    "Anvomr": [
      "Kylmedium.",
      "Framkallare.",
      "Fönsterputsmedel.",
      "Metallytbehandlingsmedel.",
      "Komponent i färg och lack.",
      "Råvara vid plasttillverkning.",
      "Träskyddsmedel."
    ],
    "ToxTolkning": [
      { "Akut giftighet": 1 },
      { "Anrikas i naturen": 2 },
      { "Cancerframkallande": 2 },
      { "Giftigt för akvatiska system": 1 },
      { "Hormon/Reproduktionsstörande": 3 },
      { "Långsiktiga skador på organ/nervsystem": 1 },
      { "Ozonnedbrytande": 2 }
    ],
    "ToxData": [
```

```
  { "logPow, bioackumulation vid logPow lika med eller över 3.0":
    "0,23 enhetslös" },
  { "EC50 Daphnia magna, 48 h":
    "25,4 mg/l" },
  { "IC50 alger, 72 h":
    "5 mg/l" },
  { "LC50 Daphnia magna, 48 h":
    "0,66 mg/l" },
  { "LC50 fisk, 96 h":
    "0,02 mg/l" },
  { "Giftighet vid inandning: LC50 inhalerat kanin, 1 h":
    "7 mg/l" },
  { "Giftighet vid inandning: LC50 inhalerat mus, 1 h":
    "2,9 mg/l" },
  { "Giftighet vid inandning: LC50 inhalerat råtta, 4 h":
    "1,4 mg/l" },
  { "Giftighet vid förtäring: LD50 oralt råtta":
    "350 mg/kg" }
],
"GhsFaror": [
  { "H221": "Brandfarlig gas." },
  { "H314": "Orsakar allvarliga frätskador på hud och ögon." },
  { "H331": "Giftigt vid inandning." },
  { "H400": "Mycket giftigt för vattenlevande organismer." }
],
"Synonymer": [
  { "en": "Ammonia anhydrous",            "typ": "1" },
  { "en": "Ammonia anhydrous, liquefied"            },
  { "fr": "Ammoniac, anhydre",            "typ": "1" },
  { "sv": "Ammoniak"                                },
  { "sv": "Ammoniak, vattenfri",          "typ": "1" },
  { "sv": "R 717",                        "typ": "2" },
  { "sv": "R717",                         "typ": "2" },
  { "de": "Ammoniak, wasserfrei",         "typ": "1" }
],
"Gransvarden": {
  "AEGL3_10": 2700.0,
  "AEGL3_30": 1600.0,
  "AEGL3_60": 1100.0,
  "AEGL3_240": 550.0,
  "AEGL3_480": 390.0,
  "AEGL2_10": 220.0,
  "AEGL2_30": 220.0,
  "AEGL2_60": 160.0,
  "AEGL2_240": 110.0,
  "AEGL2_480": 110.0,
  "AEGL1_10": 30.0,
  "AEGL1_30": 30.0,
  "AEGL1_60": 30.0,
  "AEGL1_240": 30.0,
  "AEGL1_480": 30.0,
  "KGV_5": 50.0,
  "NGV_480": 20.0,
  "IDLH": 300.0,
  "Lukt": 50.0,
  "Fornim": 5.0
},
"Namn": "Ammoniak, vattenfri",
"Fortydligande": "",
"LosningAvId": "",
"Beskrivning":
 "Giftig och frätande gas (tryckkondenserad). Vid höga koncentrationer
 inomhus även brandfarlig.",
"FarlNr": "268",
"FarlNrTxt": "Giftig gas, frätande",
"FarlNrADR": "268",
"FgADR": 3,
```

```
    "FgRID": 3,
    "FgIMDG": 3,
    "FgIATADGR": 4,
    "Klass": "2",
    "Klassif": "2TC",
    "Fpg": "",
    "UN": "1005",
    "Anm": "",
    "AggTill": "Gas; kondenserad",
    "Farg": "Färglös",
    "Lukt": "Skarp; stickande",
    "SmaltP": "-78 °C",
    "KokP": "-33 °C",
    "BrbOmr": "från 15 till 28 vol-%",
    "FlamP": "",
    "TermTP": "630 °C",
    "Dens": "682 kg/m³ vid -33 °C",
    "DensTal": "0,6",
    "Visk": "",
    "VtnLosl": "Lättlöslig i vatten (34 vikt-% vid 20 °C)",
    "Molekyl": "NH3",
    "MolVikt": "17 g/mol",
    "JonPot": "10,18 eV",
    "KritT": "133 °C",
    "AngtryckA": 13631304649.6458,
    "AngtryckB": 2836.06255488474,
    "AngtryckPkt": "857 kPa vid 20 °C",
    "Riskomrade": "",
    "Skyddsutrustning":
     "Gastät kemskyddsdräkt (typ 1) och andningsskydd, kompletterad med
     branddräkt vid brandfara.",
    "SkyddLivraddning": "Branddräkt och andningsskydd.",
    "Slackmedel":
     "Stoppa i första hand gasflödet. Måste branden släckas, använd tät
     vattenspray.",
    "AvtalKemsamverkan": "Ammoniak",
    "GhsSignord": "FARA",
    "GhsFinns": "X",
    "GhsAnm": "Ytterligare märkning kan behövas. Se hjälpen.",
    "SpridningLuft": "NH3"
  }
}
```

Bilaga D – Presentation av molekylformler

I databasen lagras molekylformler som vanlig text, t.ex. ”H2O” för vatten, trots att det korrekta sättet att skriva vattens molekylformel är med nedsänkt tvåa: H_2O. Det är klientapplikationens ansvar att snygga till de råa textmolekylformlerna till snygga, korrekta, molekylformler.

RIB Farliga ämnen använder följande logik för att bearbeta molekylformlerna:

- Alla siffror som står direkt efter en **bokstav**, en **högerparentes** ’)’ eller en **högerklammer** ’]’ visas nedsänkta.
- Efterföljande siffror visas på samma sätt som den inledande siffran, d.v.s. står det C60 så ska både 6 och 0 vara nedsänkta: ”C_{60}”.
- Övriga siffror visas normala, d.v.s. inte nedsänkta: ”U-235”.
- Punkt (.) visas som liten prick i stället (·): ”$CaSO_4 \cdot 2H_2O$”.
- Gement ’x’ behandlas som om det är en siffra: ”Pb_xAsO4”.
- Gement ’n’ behandlas som siffra om det står direkt efter **C**, **H**, en **högerparentes,** en **högerklammer** eller en annan siffra. I övriga fall behandlas det som en bokstav.
- Ett parentesuttryck som enbart innehåller siffror, gement ’n’, plus- eller minustecken, görs om till en nedsänkt formel utan parenteser: (2n+2) blir $_{2n+2}$.

Här följer några exempel för att belysa olika möjliga varianter på molekylformler. De är sorterade efter ökande ”svårighetsgrad”:

Databaslagrad molekylformel	**Molekylformeln presenteras som**
Xe	Xe
In	In
AgCN	AgCN
Li(ClO) (aq)	Li(ClO) (aq)
H2O	H_2O
C88H126O10Mn	$C_{88}H_{126}O_{10}Mn$
Zn3P2	Zn_3P_2
Zr[(NO2)2C6H2(NH2)O]4	$Zr[(NO_2)_2C_6H_2(NH_2)O]_4$
Na3[N(CH2COO)3].1H2O	$Na_3[N(CH_2COO)_3] \cdot 1H_2O$
Na2(NH4)4(V2O7).xH2O	$Na_2(NH_4)_4(V_2O_7) \cdot xH_2O$
(-CH2O-)n	$(-CH_2O-)_n$
CnH(2n+2)	C_nH_{2n+2}

Myndigheten
för civilt försvar