

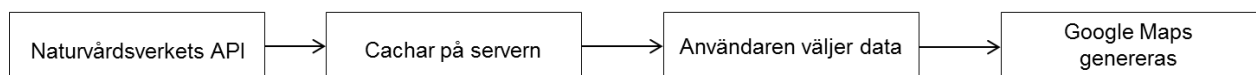
# Rapport Miljöutsläpp

## Inledning

Jag har skapat en applikation där användaren kan se om det finns något miljöfarligt utsläpp presenterat på en karta. Man kanske vill se om det finns något utsläpp i närheten av där man bor, eller på sin sommarstuga. Datat kommer från naturvårdsverket, som har samlat alla miljörapporter från de företag i Sverige som har tillstånd att släppa ut olika ämnen. De visas sedan på en karta från Google Maps. Förhoppningen från min sida är att öka medvetenheten av de miljöfarliga ämnen som sprids i naturen, även fast det finns tillstånd, och att det på sikt kan bilda en opinion för att minska användningen av dessa ämnen över huvud taget.

Projektet finns att besöka på <http://fridaholmstrom.se/php/wt2/projekt/>  
Presentationsfilm: <http://fridaholmstrom.se/php/wt2/projekt/presentation.mp4>

## Schematisk bild



## Serversida

När applikationen anropas presenteras en karta och en meny för användaren, som kan välja vad hon vill visa på kartan. I bakgrunden körs en koll om cachat material är aktuellt. Cachningen sker i php, all annan hantering med javascript och jQuery.

Jag har valt att cacha alla data från naturvårdsverket i filer med json. Vad jag förstår ger det ett snabbare svar än att spara dem i en databas som man ska fråga. Naturvårdsverket uppdaterar sin data endast två gånger om året. Jag har valt att cacha en gång i månaden då det är något osäkert när deras uppdatering sker. Jag har en lösning för cachning som gör att man inte behöver installera något tillägg på servern, vilket kan vara svårt på ett webshotell, som jämför timestampen för filerna varje gång en användare kommer in på sidan. Endast om filerna är äldre än en månad hämtas de på nytt.

För att få tag på datat från naturvårdsverket använder jag curl. Naturvårdsverket har valt formatet oData som jag inte alls förstod mig på en lång tid, men har nu förstått att det är ett bra format för öppen data då det går att använda flera olika format, och det finns redan massa parametrar man kan använda sig av för att göra förfrågningar i själva formatet i sig. Jag har valt json för att lätt kunna jobba vidare med objekten på klienten.

Felhanteringen på serversidan består av en enkel try-catch med ett felmeddelande till användaren. Eftersom jag jobbar med enbart cachad data minskar det risken betydligt för att applikationen ska påverkas av att Naturvårdsverkets api ligger nere. Man kan fortsätta använda min applikation med den sedan tidigare cachelade datan istället. Eftersom det endast kommer ny data två gånger per år, så är sannolikheten för att cachen ska uppdateras samtidigt som api:et möjligen ligger nere väldigt liten.

När jag sparar ner data har jag delat upp det utifrån de preferenser som användaren kan välja på för att det ska gå snabbare att hämta från cachan på servern när en användare efterfrågar något. Då slipper datan sållas ut allt för mycket en gång till.

## Klientsida

Vad gäller utseendet har jag försökt tänka på att ha lite dova färger som för tankarna till naturen. Jag har stilat om kartan i lite annan färgskala också för att den ska matcha bättre med det övriga, och komma ifrån "google-maps looken" lite. Därför har jag också valt egna markers. Menyn har jag hållit så enkel som möjligt då jag vill att fokus ska ligga på kartan.

Utifrån användarens val i menyn sker ajax-anrop som hämtar data att presentera på kartan utifrån den cachade datan. Varje val användaren gör genererar ett ajax-anrop som sedan skickar data vidare till Google Maps Api för att kunna generera upp markers och infowindows med information om varje utsläpp.

För de ämnen som släpps ut som jag hittat info om på wikipedia har jag valt att ge en länk i infowindow:et så användaren kan läsa vidare. Jag försökte titta på att göra detta med wikipedias api, men de sökfraser jag skulle kunna använda mig av med hjälp av naturvårdsverkets data var dels på engelska och med både förkortningar och skrivet i plural, så det gav inget bra resultat. Därför valde jag att manuellt översätta och leta reda på en artikel för att ge bättre info till användaren.

Felhanteringen på klientsidan har jag löst genom att presentera ett meddelande till användaren ovanför kartan om det inte finns något i svaret från Naturvårdsverket. Har man skrivit en felaktig fråga till api:et kommer en tom array tillbaka, och i detta fall fångas det inte av något error i ajaxanropet.

För caching på klienten har jag valt att ändra på "Cache-Control" för js och css-filer. De behöver inte hämtas om så snabbt som är standardvärde. Övriga resurser som hämtas från min server har jag inte ändrat på, då webbläsarna cachar dem på ett tillfredställande sätt för min applikation.

## Reflektion

Jag började ganska tidigt med att klura ut en idé att jobba med. Sen försökte jag förstå mig på formatet oData, vilket jag trodde skulle gå snabbt och enkelt. Men det var en så omfattande dokumentation kring det att jag missade de delar jag hade användning för enda fram tills mellandagarna. Jag har gett mig själv jul- och nyårsledigt från kodandet, men tankarna har ändå varit igång för hur jag ska lägga upp projektet. Det har gjort det till en enklare process under själva kodskrivandet. Det är ofta förarbetet som är mest tidskrävande, jämfört med själva implementationen i sig. Att få upp en arkitektur på projektet är svårt. Det har alltid lönat sig för mig att tänka och skissa på den innan jag börjar koda, än att bara "sätta igång".

Jag hade problem i början med den enorma mängd data som naturvårdsverkets api ger. Det gick inte att hämta all data jag ville ha på en gång i en förfrågan, utan att det kraschade. Det har gjort att jag delat upp datat per län eller kategori som jag fått jobba vidare från. Då har det blivit ett par fler ajaxanrop än vad jag skulle klara mig med om alla data hämtats i en klump, men samtidigt kanske det ger en snabbare applikation då varje ajaxanrop blir mindre.

Cachning har varit ett nytt område för mig som jag fått försöka sätta mig in i så gott som möjligt. Jag tycker att jag kommit fram till en rätt så bra lösning på serversidan, men klientsidan kan man säkert

jobba mer med. Dock har inte min applikation mer än en bild tex som brukar vara lämpliga att cacha på klienten, sedan är det många förfrågningar till google för kartans del, som jag inte tror blir något bra att cacha. Jag har i alla fall försökt mig på att ändra cachtiden för js och css-filer. Men jag känner att detta är ett område jag behöver lära mig mer inom.

Jag skulle vilja lägga till funktion för att kunna dela med sina vänner på olika sociala nätverk, för att enkelt kunna sprida informationen om miljöutsläpp på ett enkelt sätt framöver.

## **Risker**

En risk som alltid finns är att någon av producenterna av de apier jag arbetar med slutar att underhålla dem, eller tar bort dem helt. Det skulle leda till att hela användarupplevelsen försvinner. Etiskt kan säkert vissa av de företag som visas upp i applikationen känna sig uthängda, trots att det är en offentlig handling uppgifterna bygger på. Jag har dock gjort bedömningen att det är helt ok att visa denna data, mycket baserat på att Naturvårdsverket valt att göra ett api för just utsläpp från företag. Jag skulle själv önska att datat uppdaterades lite oftare för att kunna se mer färska utsläpp.