Om indirekta gränssnitt till myndigheters webbtjänster

1 Problembeskrivning

Myndigheters webbtjänster har ett större användningsområde än bara den enskilda medborgaren som vill hitta information vid ett givet tillfälle. De informationsresurser som finns tillgängliga är också en plattform som andra aktörer kan vilja bygga vidare på. Därför bör myndigheter, vid webbpublicering, tänka på de indirekta gränssnitt man skapar vid sidan av de direkta, användarinriktade gränssnitten. Av dessa indirekta gränssnitt är URL-gränssnittet det viktigaste.

2 Exemplifiering

URL:en[1] till upphovsrättslagen, så som den publiceras i Regeringskansliets rättsdatabaser, ser ut såhär:

```
http://62.95.69.15/cgi-
bin/thw?%24%7BHTML%7D=sfst_lst&%24%7BOOHTML%7D=sfst_dok&%24%7BSNHT
ML%7D=sfst_err&%24%7BBASE%7D=SFST&%24%7BTRIPSHOW%7D=format%3DTHW&B
ET=1960%3A729%24
```

Den är 171 tecken lång, i praktiken oläslig, och i många fall kommer detta ställa till problem (om man exv vill skicka länken i ett epostmeddelande, eller läsa upp den för någon i telefon), men framförallt är den inte framtidssäker - inget talar för att någon har för avsikt att se till att den här adressen kommer fungera i fortsättningen. Det här är en **bräcklig** URL, eftersom den har stor sannolikhet att gå sönder,[2] och därmed bidra till problemet med länkröta.[3]

2.1 Att läsa en URL

Om vi utgår från en enklare URL (säg, Upphovsrättslagen som den publiceras på lagen.nu, `http://lagen.nu:80/1960:729#P49`) kan vi observera fem komponenter:

- protokoll (”`http`”)
- hostnamn (”`lagen.nu`”) – det här börjar av gammal hävd med ”www”, men det finns få anledningar till att ha detta.[4]
- Port (”`:80`”) - utelämnas nästan alltid, då det är underförstått genom protokollangivelsen vilken TCP-port som används
- Sökväg (”`1960:729`”) – det är i den här komponenten
- Fragment (”`P49`”) – för att adressera en viss **del** av en sida, i det aktuella fallet just 49 § i upphovsrättslagen

Om vi analyserar ovanstående URL enligt detta schema kan vi börja med att notera att webbservern inte har något namn i DNS – istället för ett begripligt datornamn används IP-numret (62.95.69.15). Detta får till effekt att om databastjänsten någon dag flyttas till en dator med ett annat IP-nummer – vilket det kan finnas många goda

[1] Uniform Resource Locator, även känt som ”webbaddress”. I det följande används begreppet URL då det har en tydligare teknisk definition (RFC 1738, se exv http://www.ietf.org/rfc/rfc1738.txt)

[2] Om undersökningar kring olika typer av URL:ers benägenhet att sluta fungera, se exv Spinellis, Diomidis, ”The decay and failure of web references”, Communications of the ACM, Vol 46, no 1, s71-77, http://portal.acm.org/citation.cfm?id=602422

[3] ”Link rot”, när länkar från en plats till en annan slutar att fungera. ”Even worse, linkrot contributes to **dissolving the very fabric of the Web**: there is a looming danger that the **Web will stop being an interconnected universal hypertext** and turn into a set of isolated info-islands. Anything that reduces the prevalence and usefulness of cross-site linking is a direct attack on the founding principle of the Web”, Jacob Nielsen, ” Fighting Linkrot”, http://www.useit.com/alertbox/980614.html

[4] De tre anledningar jag kan komma på är att det tekniskt sätt är (något) enklare att DNS-mässigt sätta upp en adress-till-IP-nummerkoppling för ”www.organisation.se”, snarare än för organisationsdomänen som helhet, att det ökar möjligheterna att i löpande text identifiera webadresser, samt att användarna förväntar sig att adresser ska börja med ”www.”

tekniska anledningar att göra – kommer alla länkar till lagtexter som gjorts att sluta fungera.

Dessutom så exponerar URL:en en massa oväsentlig information om hur tjänsten rent tekniskt sett är uppbyggd. Att veta att parametern `${SNHTML}` ska ha värdet `"sfst_err"` är i bästa fall ointressant, och i värsta fall ett potentiellt säkerhetshål.[5] Det är helt enkelt inte bra systemdesign att exponera interna parametrar eller övriga tekniska implementationsdetaljer utåt.

Den skulle kunna se ut såhär::

```
http://rattsdatabaser.regeringen.se/SFST/1960:729
```

- DNS-namnet speglar vem som är ansvarig för informationen/tjänsten
- Första delen av sökvägen (`"/SFST/"`) talar tydligt om vilken databas resursen finns i.
- Det sista ledet av sökvägen (`"1960:729"`) är en unik identifierare för den resurs vi är intresserade av.

Inga ovidkommande tekniska detaljer exponeras utåt, och länken är kort nog (49 tecken) att läsas upp över telefon eller skrivas på en servett.[6]

3 Att använda en myndighets webbtjänst som plattform

En myndighets webbtjänst kan, förutom att hjälpa den enskilde i stunden, i princip (åter-)användas som plattform på tre olika sätt. En myndighet bör underlätta framförallt det första av dessa sätt.

3.1 Extern informationsresurs att referera genom länkning

Det här är den enklaste kategorin av användning. En utomstående aktör, som vill underbygga sin egen text eller sina egna tjänster med hänvisningar till källmaterial, länkar direkt till sidor på myndighetens webbplats där informationen finns.

Att hänvisa till myndighetens förstasida (exv. `http://www.regeringen.se/`) är inte tillräckligt, eftersom den relevanta informationen kan ligga flera klick bort och vara svår att navigera till. Instruktioner i stil med ”Gå till Lagrummet, klicka på ’Författningar’, klicka på ’författningar i fulltext’, klicka på ’utökad sökning’ i det nya fönstret, skriv in ’1960:729’ i rutan för SFS-nummer, tryck ’Sök’, så dyker lagtexten upp i ett tredje fönster” är en drastiskt försämrad användbarhet jämfört med vad webben är kapabel till.

3.2 Informationsresurs att utvinna genom screenscraping

Nästa steg av återanvändning är att en utomstående aktör utvinner information från myndighetens webbplats genom att programmatiskt ladda ner sidorna och extrahera det relevanta materialet, s.k. screenscraping eller webscraping.[7] Man kan skilja på enkel screenscraping, där en enskild sida laddas ner och behandlas, och flerstegsscraping, där man går igenom en flerstegsprocess, exempelvis simulerar en webbläsare som fyller i ett sökformulär, trycker på sökknappen och sedan går igenom vart och ett av sökresultaten.

Screenscraping, till skillnad från länkning, bygger på två implicita gränssnitt: URL:er och HTML-struktur. Tekniken är till sin natur bräckligare, då minsta lilla förändring i den skrapade tjänstens HTML-kod kan leda till att processen slutar fungera. Denna struktur kan inte sägas vara en del av en utfästelse på samma sätt som en URL är, och en aktör som använder screenscraping måste därför vara beredd att övervaka den/de webbplatser han skrapar för att se att hans program fungerar över tiden.

---

[5] Exempel på möjligt säkerhetsproblem: är det en sökväg till en mallfil (template)? Vad händer om vi i så fall byter ut den mot `"../../../../etc/passwd"`?

[6] Se även ”Vägledningen 24-timmarswebben 2.0”, avsnitt 4.2.14-15.

[7] Denna användning kan väcka ett flertal juridiska frågor, exempelvis om materialet i fråga lyder under upphovsrätt, eller om gärningen kan ses som dataintrång, som vi bortser från här.

Det är denna teknik som lagen.nu använder för att hämta information från Regeringskansliets rättsdatabaser och domstolsverkets webbplatser.

3.3 Komponent att integrera genom väldefinerat API

Det blir allt vanligare att webbtjänster, framförallt de som vill profilera sig som varandes på framkanten av den tekniska utvecklingen, erbjuder ett väldefinierat programmeringsgränssnitt för att nå den information som man tidigare använde screenscraping för. Gränssnitten bygger vanligtvis på någon av standarderna XML-RPC eller SOAP.

ProgrammableWeb listar 129 olika sådana definerade API'er för många av de populäraste amerikanska webbtjänsterna.[8] Genom dessa kan en utvecklare lätt återanvända information och funktionalitet från webbtjänster. Ett av de tidigaste exemplen på vad detta möjliggör är busmonster.com, som använder funktioner från Google Maps och kopplar dessa till busslinjeinformation och trafikdata för att visa hur busstrafiken i Seattle flyter.

I Sverige är användandet av webbtjänster med APIer lågt, men för myndigheter, speciellt i relation till visionen om 24-timmarsmyndigheten,[9] har tekniken stor potential.[10]

4 Att utforma permanenta URL:er

Det viktigaste att förstå vid URL-utformning är att de URL:er som syns utåt inte behöver ha någon direkt relation till hur material är lagrat/strukturerat internt på webbservern.

För att ta ett exempel från lagen.nu: De två URL:erna http://lagen.nu/1960:729 respektive `http://lagen.nu/1960:729.xml` (där den sistnämnda pekar på en XML-representation av texten från den första) motsvaras i filsystemet av de två filerna `html/1960/729.html` respektive `xml/1960/729.xml`. Dvs, två resurser som utåt sett ser ut att ligga "bredvid" varandra är internt placerade i två helt separata kataloger.

Det andra viktiga att förstå är att det aldrig någonsin finns en godtagbar anledning att leverera statuskod 404. I samband med en teknisk eller strukturell omdaning av webbplatsen måste man sätta sig ner och fundera över vilka resurser man haft tillgängliga och under vilka URL:er dessa har legat. Man måste sedan tillse att dessa URL:er funkar även i framtiden, antigen genom att direkt ge rätt sida, eller att genom HTTP-statuskoden 301 ("Moved permanently") ange vilken den nya adressen är.

4.1 Funktionell utformning

För att utforma funktionella URL:er måste man tänka igenom de resurser man erbjuder. Det första steget är kanske just att börja tänka i termer av resurser, snarare än filer och dokument.

Man kan grovt göra en uppdelning mellan informationsresurser och funktionsresurser, även om gränsen dem emellan inte är knivskarp. Skatteverkets blankettsamling är en informationsresurs, innehållandes en samling av dokument.[11] Samma myndighets tjänst för att räkna ut inkomstskatt är en funktionsresurs.[12]

I det följande behandlar jag främst informationsresurser; även om funktionsresurser också bör ha permanenta URL:er så är avvägningarna där annorlunda (mängden funktioner tenderar att vara drastiskt mindre än mängden dokument, exempelvis).

---

[8] Se http://www.programmableweb.com/apis

[9] Se E-nämndens rapport 04:01, "Vägledningen 24-timmarswebben 2.0", avsnitt 2.2.10, http://www.e-namnden.se/enamnden/templates/Page____511.aspx

[10] Hedin, Carl & Uppström, Mattias: "Web services i 24H-myndigheten – potential och utmaningar", magisteruppsats, http://www.handels.gu.se/epc/archive/00004394/

[11] http://www.skatteverket.se/skatteverketsblanketter.4.18e1b10334ebe8bc800014.html

[12] http://www.skatteverket.se/webdav/files/servicetjanster/skatteutrakning2006/1prelskut05ink.html

4.1.1 Hierarkiska sökvägar och ”hackbara” URL:er

Om informationsresursmängden låter sig modelleras i en hierarkisk struktur bör denna reflekteras i URL:en.

Exempel: URL:en till Stockholms universitets information om IT-stödet i universitetsbiblioteket är `http://www.su.se/pub/jsp/polopoly.jsp?d=204&a=1983`. Den ”location breadcrumb trail”[13] som visas på sidan antyder att informationen om it-stöd i biblioteket sorteras under information om teknik och data, som i sin tur är en undergrupp av studentservicen i stort, osv. Denna hierarki borde reflekteras i sökvägen (`http://www.su.se/student/service/it/biblioteket`) . När man reflekterar en dokumenthierarki i sökvägen bör man göra det möjligt att ”hacka bort” slutled från denna, så att `/student/service/it` leder till en översikt över alla resurser som hör till IT-stödet för studenter, osv.

4.2 Teknisk implementation

Som tidigare sagt behöver inte URL-strukturen utåt sett ha något med hur filer eller databasposter faktiskt är ordnade. Det är till och med en fördel om man så långt som möjligt kan separera dessa två världar. Att tänka igenom skillnaderna gör det enklare att se vilka resurser det är man faktiskt erbjuder.

4.2.1 Något om webbpubliceringssystem och dess URL-hantering

Det är inte ovanligt att publiceringssystem för webbplatser förvärrar problemet med obegripliga och bräckliga URL:er.

Ett exempel är Stockholms universitets nya webbplats, där publiceringssystemet Polypoly används. URL:en för information om IT-stödet i universitetsbiblioteket refererades ovan. Varken parametern d med värdet 204 eller parametern a med värdet 1983 säger oss någonting. Förmodligen refererar värdet 204 på parametern d till en kategori av dokument (de som har med IT-service att göra), medan värdet 1983 refererar till ett visst givet dokument. Dessa värden är dock interna detaljer som det inte finns någon anledning att exponera utåt.

4.2.2 mod_rewrite

För webbtjänster som bygger på webbservern Apache kan man relativt enkelt designa en mappning mellan externa och interna URL’er genom modulen mod_rewrite.[14] Dvs, om man redan har en utvecklad webbplats, och vill införa ett system med persistenta URLar kan man göra det med en mod_rewrite-basererad nivå av indirektion. Detta skulle kunna användas på exemplet med informationen om IT-stödet i universitetsbiblioteket genom följande regler (otestat):

```
RewriteMap kat txt:/path/to/kategori.map
RewriteMap dok txt:/path/to/dokument.map
RewriteRule ^/([^/]+)/(.*) /pub/jsp/polopoly.jsp?d=${kat:$1}&a=${art:$2}
```

4.2.3 isapi_rewrite

isapi_rewrite är ungefär som mod_rewrite, fast för webbservern Internet Information Services (IIS).[15]

5 Summering

Länkröta är ett allvarligt problem med webben som den ser ut idag. Myndigheter, speciellt med tanke på visionen om 24-timmarsmyndigheten, bör dra sitt strå till stacken för att motverka problemet. Med en genomtänkt URL-design kan detta uppnås.

---

[13] Begrepp taget från Rogers, Bonnie Lida & Chaparro, Barbara: ”Breadcrumb Navigation: Further Investigation of Usage”, http://psychology.wichita.edu/surl/usabilitynews/52/breadcrumb.htm

[14] Se ”URL Rewriting Guide”, http://httpd.apache.org/docs/2.0/misc/rewriteguide.html

[15] http://www.isapirewrite.com/