Denne guide er oprindeligt
udgivet på Eksperten.dk

COMPUTERWORLD

Korrekt visning af æøå - character encoding

Udviklere bøvler ikke kun med DOCTYPE og HTML-standarder - også tegnsætsproblemer ses ofte så det forsøger vi at gøre noget ved nu.

Skrevet den **31. Jul 2009** af **keysersoze** I kategorien **Programmering / Generelt** | ★★★☆☆

I en tidligere artikel her på siden forsøgte jeg at komme med et opråb om korrekt brug af DOCTYPE og valid kode da det så ud til, at mange havde deres egen definition af, hvordan HTML kode kunne struktureres - og lige som om at det ser ud til at problemer med DOCTYPE og invalid HTML er lidt et "mode-problem" i øjeblikket lader det også til, at mange har problemer med deres encoding (tegnsæt) selvom dette også er en udfordring, der har eksisteret så længe man har skrevet web - så nu må vi hellere forsøge at rettet op på det.

Hvad består fejlen i og hvornår

Kort fortalt handler problemet med encoding om, at hvis man ikke har styr på encodingen i sin applikation og fx kører med flere forskellige i de forskellige lag i applikationen, så er der stor risiko for problemer med æøå og andre special karakterer på sin hjemmeside. Problemet omhandler ikke kun dynamiske applikationer - også simple statiske html-sider er ramt. Værst af alt, så er der modsat valid HTML-kodning reelt meget få ting at holde styr på her, men heldigvis er løsningerne langt hen ad vejen også rimelig simple.

Selvom det ikke bliver til meget kode, vil jeg i artiklen tage udgangspunkt i ASP.NET bare for at have et fast holdepunkt, men udfordringerne med korrekt tegnsæt kan lige så vel ramme klassisk ASP, PHP og altså selv simpel HTML.

Valg af encoding

Der findes ikke et endeligt svar på hvilken encoding man skal gå efter - men jeg kan ikke komme på en grund til ikke at vælge UTF-8 (Unicode Transformation Format 8-bit) hvorimod der er flere grunde til ikke at vælge det i Danmark nok mest kendte og brugte, ISO-8859-1, da det fx må betragtes som værende forældet da flere JavaScript-funktioner er udgået til fordel for tilsvarende Unicode-funktioner. Så selvom ISO-8859-1 flere steder anses som dét man benytter på hjemmesider i Danmark og vesteuropa så er min mening, at UTF-8 er det eneste rigtige valg i dag.

Encoding på filen

Det første sted at starte med det korrekte tegnsæt er allerede når vi laver vores filer, og det er nok her de fleste går fejl da de færreste er opmærksom på, at det har noget at sige og måske endda slet ikke er opmærksom på, at filer kan overhovedet gemmes med forskellig encoding. Visual Studio opretter som udgangspunkt filer i UTF-8, så i bund og grund behøver vi ikke bekymre os om det store her efter vores valg af netop UTF-8, men vil vi gerne være sikre kan vi åbne vores fil og klikke på Advanced Save Options i File-menuen for at se filens nuværende format og eventuelt gemme den med en ny encoding. Notepad er er også et godt lille program til at se filens encoding i, nemlig ved at klikke på Gem som og se hvilket format der står under Kodning - det "farlige" ved Notepad er, at de fleste vil opleve, at den som standard gemmer nye filer i ANSI og ikke UTF-8, og genkender Notepad ikke formatet på den fil man åbner vil det derfor også være ANSI, der gemmes i hvis man gemmer filen igen og dermed overskrive man det eksisterende format.

Encoding i HTML koden

Dette punkt er det nok de færreste der er i tvivl om så det bliver meget kort - i HTML kan vi benytte meta til at sende informationer om vores dokument til klienten, og en af de informationer vi bør sende med er Content-Type;

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Encoding på data(basen)

Langt de fleste applikationer i dag er dynamisk opbygget ud fra data i et lager - mest oplagt selvfølgelig en database, som er det jeg vil tage udgangspunkt i selvom det bliver relativt kort da korrekt database-opsætning kan være en videnskab i sig selv - og her er encoding altså bestemt heller ikke uden betydning.

Præcis hvordan encodingen, og dataene generelt, håndteres her er lidt forskelligt fra den ene database til den anden så sæt dig godt ind i den database du arbejder med. Taler vi MySQL kan man sætte tegnsættet direkte på databasen og i MSSQL kan lidt af styringen ligge i valg af datatype, hvor nchar, nvarchar og ntext til unicode (hvilket reelt vil sige, at den valgte collation kun styrer sortering og ikke tegnsæt) og char, nvarchar og text til ikke-unicode (som altså styres af collation så æøå kun kan gemmes ved korrekt valg af collation), men her er det også værd at overveje performance ind i valget da unicode datatyperne er tungere at arbejde.

Men ellers ligger det vigtigste tiltag omkring encoding-opsætningen i valget af ens collation - collations handler ikke kun om encoding men også om sortering og sammenligning af data - et simpelt eksempel på dette kunne være en numeriske collation, der sorterer 1 før 2 eller en karakter collation, der bestemmer om hvorvidt a er det samme som á eller ej, men det er ikke desto mindre vigtigt at tage stilling til.

Encoding fra serveren

Det sidste sted det kan gå galt er i forhold til hvilken encoding serveren sender dokumentet afsted i - og her ved jeg at flere webhoteller vælger at sender ISO-8859-1 som standard da det stadig er det format langt de fleste vælger og dette betyder altså, at vi har noget at skal tage forhold for med vores valg af UTF-8. Heldigvis er det ret simpel at styre da vi i vores web-config kan tilføje følgende til at overskrive webhotellets valg;

```
<system.web>
  <globalization responseEncoding="utf-8" requestEncoding="utf-8" fileEncoding="utf-8" />
</system.web>
```

Tilsvarende muligheder har vi selvfølgelig til rådighed i fx PHP gennem .htaccess.

Vil du se hvad serveren sender af sted sammenlignet med din meta kan du gå ind på http://validator.w3.org/, vælge More Options og sætte hak i Verbose Output før du validerer dit dokument - er der uoverenstemmelser mellem dokumentets og serverens format vil du her få besked om det.

Ikke flere problemer med æøå

Nu har vi endelig fået styr på hvilke 4 flaskehalse der er for at få æøå vist korrekt på vores hjemmeside - så dem ser vi selvfølgelig ikke flere af fremover :)