Engineer

# Feed

en sides oppdateringer, i et format du kan abonnere på

## Hva det er

På en fast adresse hos mange nettsteder ligger en feed: en fil som lister opp de siste innleggene i en form programvare kan lese, hvert med tittel, dato, en identifikator som aldri endres, og som regel selve teksten. Et program kan hente filen etter en fast plan og vise en person bare det som er nytt. Andre navn er webfeed, syndikeringsfeed og nyhetsstrøm, og mange sier «RSS-feed» om alle sammen, uansett hvilket format som ligger under.

Det som skiller en feed fra andre kanaler, er retningen. Et nyhetsbrev blir dyttet ut: utgiveren fører en liste over adresser og sender til hver av dem. En feed blir hentet: utgiveren legger én fil på serveren, og feedleseren til hver abonnent henter den, sammenligner den med kopien den så sist, og beholder elementene den ikke har møtt før. Utgiveren får aldri vite hvem som abonnerer, abonnenten gir aldri fra seg en e-postadresse, og å slutte er et spørsmål om å slette én linje i en feedleser.

Tre formater bærer nesten alle feeder på weben. RSS 2.0 er XML, daterer elementene sine i stilen fra RFC 822 ( `Sat, 07 Sep 2002 00:00:01 GMT` ) og er erklært frosset av sin egen spesifikasjon. Atom, publisert av IETF som RFC 4287 i desember 2005, er også XML og strengere: hver feed og hver oppføring har en permanent `id` og et `updated` -tidsstempel på formen fra RFC 3339, og en forfatter må være oppgitt et sted. JSON Feed, i versjon 1.1 siden august 2020, sier det samme i JSON, for utviklere som helst slipper å parse XML.

En side forteller programvare hvor feedene dens ligger, med `<link rel="alternate">` -elementer i hodet, og hvert av dem oppgir en medietype: `application/rss+xml` , `application/atom+xml` eller `application/feed+json` . Slik finner en feedleser feeden ut fra ingenting annet enn den vanlige adressen til nettstedet.

### Bruk begrepet i en beslutning

Bruk «Hva er en feed?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Et nettsted som er raskt, lett å finne og kan redigeres av dine egne ansatte, uten en regning hver gang en formulering endres.

**Se tjenestene våre** Ta kontakt →

### Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Utviklet 100 webapplikasjoner med HTML/HTML5 — 100 webapplikasjoner bygd

**Utviklet 100 webapplikasjoner med HTML/HTML5** Ta kontakt →

## Hvorfor det er viktig

Den første avgjørelsen er om du skal publisere en feed i det hele tatt. En feed er ett utdataformat til som skal ha mal, valideres og holdes stabilt i årevis, for et publikum som er mindre enn et sosialt nettverks. På den andre siden er det den eneste abonnementskanalen som ikke trenger noen plattform, noen konto eller noen e-postliste, og det står ingen rangeringsalgoritme mellom et innlegg og menneskene som har bedt om å få se det.

Den andre avgjørelsen er hvilke formater. Noen feedlesere tar bare RSS og noen bare Atom, og du kan ikke vite på forhånd hvilket den enkelte abonnentens leser vil ha. Å publisere begge dekker hullet til prisen av en mal til, og JSON Feed legger til et tredje format for programvare skrevet av folk som heller håndterer JSON. Hvert ekstra format er enda en fil å holde i takt: de samme elementene, de samme identifikatorene, de samme datoene.

Den tredje er hvor mye tekst hvert element skal ha. En feed med hele teksten lar folk lese hele stykket i feedleseren sin, og det er mye av poenget med en feedleser, men den koster utgiveren sidevisninger. En feed med bare sammendrag sender hver leser til nettstedet for hvert element. Begge deler er greit; valget står mellom hvor langt teksten reiser, og hvor ofte selve siden får besøk.

Uansett svarene er adressen et løfte. Feedlesere spør etter den URL-en de fikk, etter en plan hver leser velger selv (RSS lar en feed foreslå et intervall i minutter med `ttl`, og leseren kan se bort fra det), og mange fortsetter å spørre i årevis. Når adressen til en feed først er ute, blir den værende.

# Slik ser det ut i praksis

Dette nettstedet publiserer alle tre formatene fra den norske forsiden sin, på `/nb/index.xml` (RSS), `/nb/atom.xml` (Atom) og `/nb/feed.json` (JSON Feed). Kolofonen lister dem opp sammen med tjenestene som sjekker dem: alle tre har hele teksten, og det er to XML-feeder i stedet for én fordi noen feedlesere bare tar imot ett av de to formatene, og hvilket det er, varierer. W3C Feed Validator sjekker Atom- og RSS-feeden, og JSON Feed Validator sjekker den tredje, som W3C-tjenesten ikke dekker fordi den er eldre enn formatet. W3C-tjenesten advarer om at de fleste oppføringene har samme endringsdato. Siden lar advarselen stå i stedet for å skjule den: datoene kommer fra git-historikken, og de fleste sidene ble skrevet i de samme få commitene.

Kontaktsiden tilbyr feedene som en måte å følge selskapet på uten konto, ved siden av de sosiale profilene, og gjentar at alle tre inneholder hele teksten.

Alle tre lages ved byggetid, ikke for hånd og ikke når noen ber om dem: prosjektet som bygde nettstedet genererer nettstedet i sin helhet ved byggetid, uten noen backend ved kjøretid. Hver side publiseres i flere representasjoner fra ett innholdstre, og forsiden legger i tillegg til tre syndikeringsfeeder og et manifest for installerbar applikasjon.

# Hvor det går galt

**Hvert gammelt element dukker opp igjen som nytt.** Etter et nytt design, en flytting til et annet domene eller en endret mal ser abonnentene dusinvis av elementer de leste for måneder siden. En feedleser kjenner igjen et element på identifikatoren, `id` i Atom, `guid` i RSS eller `id` i JSON Feed, og når den strengen endres, er elementet nytt for leseren. RFC 4287 sier uttrykkelig at en oppførings `id` skal overleve flytting, migrering og ny publisering. En permalenke er bare en god identifikator så lenge nettstedet lover å beholde den gamle strengen etter en flytting. I RSS bør du også si hva `guid` er: spesifikasjonen setter `isPermaLink` til true som standard, og RSS-malen på dette nettstedet skriver `isPermaLink="true"` likevel, fordi en leser som gjetter motsatt, importerer hvert element på nytt som et duplikat.

**Feeden påstår at den er endret ved hvert bygg.** Datoen på feednivå flytter seg hver gang nettstedet bygges på nytt, enten noe er publisert eller ikke, så programvare som følger med på den datoen for å oppdage nyheter, slår ut på ingenting. Årsaken er en mal som stempler klokkeslettet ved byggetid. RSS

definerer `lastBuildDate` som siste gang innholdet i kanalen ble endret, og `updated` på feednivå i Atom betyr det samme, så datoen bør hentes fra det nyeste elementet. Atom-malen på dette nettstedet tar den fra den nyeste oppføringen av akkurat den grunnen.

**Feeden lar seg ikke lenger parse etter en typografisk endring.** Feedleserne melder at feeden er ødelagt, og en validator peker på en udefinert entitet i den første artikkelen med et krøllete anførselstegn. XML forhåndsdefinerer fem navngitte entiteter, `lt` , `gt` , `amp` , `apos` og `quot` , mens HTML definerer hundrevis flere. En Markdown-renderer som skriver anførselstegn og tankestreker som navngitte entiteter fra HTML, lager kode som er grei i en side og fatal inne i et XML-element. Escape HTML-koden før den settes inn, slik innhold med `type="html"` i Atom forventer, eller pakk den inn i en CDATA-seksjon, som RSS Best Practices Profile tillater for `content:encoded` . Atom-malen på dette nettstedet escaper innholdet sitt, og kommentaren ved siden av den skriver at feeden ikke lot seg parse i det hele tatt da innholdet ble skrevet rått.

**Abonnentene forsvinner etter en flytting, og det merkes bare som stillhet.** En feed har ingen avmeldingsknapp og ingen liste som krymper, så det eneste tegnet er at det blir stille. Adressen til feeden ble endret, eller filen ble fjernet, og hver feedleser som spør etter den gamle URL-en, får nå en feil. La den gamle adressen fortsette å svare, enten med selve feeden eller med en permanent omdirigering. Dette nettstedet beholder RSS-feeden sin på `index.xml` , standardstien i den statiske nettstedsgeneratoren det er bygd med, fordi det er den adressen eksisterende lesere spør etter. På et språk det ikke annonserer, skriver det likevel velformede feeder uten elementer, så en leser får et gyldig dokument i stedet for en 404.

**To feeder smelter sammen til én i en abonnementsdialog.** En nettleserutvidelse eller en feedleser tilbyr én feed der siden deklarerer to. To alternative lenker med samme type og samme tittel ser for et slikt verktøy ut som én feed, og det viser den det så sist. Gi hver sin egen tittel. Sidene på dette nettstedet navngir sine med nettstedets navn fulgt av «(RSS)», «(Atom)» eller «(JSON Feed)».

## Les mer

- RFC 4287: The Atom Syndication Format — IETF-standarden, med de påkrevde elementene, permanente identifikatorer og datoer etter RFC 3339.

- RSS 2.0 Specification — teksten til det frosne formatet fra RSS Advisory Board, element for element.
- JSON Feed Version 1.1 — den samme ideen uttrykt i JSON, og hva som er endret siden versjon 1.0.
- HTML Standard: link type “alternate” — hvordan en side viser programvare veien til feedene sine.
- W3C Feed Validation Service — sjekker en RSS- eller Atom-feed ut fra adressen og forklarer hver advarsel.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

**Se hva det koster** Ta kontakt →

## Gjør begrepene praktiske

Spør hvordan «Hva er en feed?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

**Se tjenestene våre**

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

**Laget med kjærlighet av D.**

**CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N**

**Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.**

**Skrevet ut fra https://engineer.company/nb/glossary/feed/**