

Engineer

WordPress

et publiseringssystem med åpen kildekode for nettsteder



Hva det er

WordPress, ofte forkortet til WP, er et system for innholdsstyring (CMS) med åpen kildekode: PHP-programvare som holder innleggene, sidene, opplysningene om medier, brukerne og innstillingene til et nettsted i en MySQL- eller MariaDB-database, og som setter sammen sider av dem når noen ber om dem. Opplastede mediefiler ligger vanligvis utenfor databasen. WP er det samme prefikset som programmets egne filer og mapper bærer (`wp-config.php` , `wp-content` , `wp-admin`). Programvaren er utgitt under GPLv2 (eller senere), så den koster ingenting å laste ned, kjøre eller endre.

Et WordPress-nettsted består av tre lag. Kjernen (*core*) er den felles koden alle laster ned og oppdaterer. Et tema styrer presentasjonen, og utvidelser (*plugins*) legger til oppførsel; det er der prosjektets egen håndbok for temaer trekker grensen. Begge griper inn i kjernen gjennom kroker (*hooks*), navngitte punkter i WordPress-koden der annen kode får lov til å kjøre, og krokene finnes i to slag: handlinger (*actions*) og filtre (*filters*). En handling kjører i et bestemt øyeblikk, f.eks. når et innlegg lagres, og returnerer ingenting. Et filter tar imot en verdi, endrer den og gir den tilbake. Dette filteret korter ned de automatiske utdragene av innlegg fra standardlengden på 55 ord til 30:

```
add_filter( 'excerpt_length', function ( $length ) {
    return 30;
} );
```

Temaene har skiftet form gjennom årene. Klassiske temaer bruker malsystemet i PHP som kom med WordPress 1.5 i 2005, og de er fortsatt mye i bruk. Blokktemaer er dagens tilnærming: malene deres er HTML-filer bygd av blokker,

som kan redigeres i nettstedsredigereren (*Site Editor*), mens farger, skrifter og avstander ligger i en `theme.json` -fil. Hybridtemaer er klassiske temaer som har tatt i bruk noen av blokkfunksjonene.

Innholdet kan også forlate nettstedet som JSON gjennom REST-API-et, som blokkredigereren selv er bygd på. En forespørsel til adressen `/wp-json/` på nettstedet returnerer et kart over hver rute det tilbyr. Offentlig innhold kan leses uten innlogging, og alt som er privat, krever autentisering. Ethvert program kan hente de tre siste innleggene slik:

```
curl -s 'https://example.com/wp-json/wp/v2/posts?per_page=3'
```

For å kjøre det anbefaler prosjektet PHP 8.3 eller nyere, MySQL 8.0 eller MariaDB 10.11 eller nyere, og HTTPS, helst på Apache eller Nginx. Joomla og Drupal er systemer for innholdsstyring av samme slag, og de to andre veiene å veie det mot, er en skreddersydd applikasjon på et rammeverk som Django og en statisk nettstedsgenerator.

Bruk begrepet i en beslutning

Bruk «Hva er WordPress?» 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 egentlige beslutningen bak WordPress er hvem som endrer nettstedet etter lansering, og hvem som holder det trygt mens de gjør det. WordPress gir ansatte uten teknisk bakgrunn en komplett redigerer: de kan publisere en side, bytte et bilde, legge til en bruker eller bygge om en meny uten å opprette en sak. På et nettsted som endres hver uke, er den billigste endringen den som ikke trenger en utvikler.

Lisensen er gratis, men nettstedet er det ikke. GPL dekker programvaren, og wordpress.org legger til grunn at utvidelser og temaer er avledede verk som arver den samme lisensen. Utgiftene flytter seg andre steder: et webhotell som kjører PHP og en database, timene det tar å velge, sette opp og teste et tema, og det løpende arbeidet med å holde kjernen, utvidelsene, temaene og PHP-versjonen oppdatert. Den siste kostnaden løper så lenge nettstedet lever, og et tilbud bør navngi hvem som bærer den.

Vedlikeholdet følger av at WordPress er en levende applikasjon. Generering av sider som ikke er mellomlagret, kjører PHP mot en database; en helsidecache kan levere en lagret side. Utvidelseskode kjører med samme tilgang som kjernen, og innloggingsskjermen står åpent ut mot internett med mindre noen setter en ekstra lås foran den. Bakgrunnsoppdateringer av kjernen kom i versjon 3.7. Nye installasjoner fra 5.6 aktiverer som standard både mindre og større kjerneoppdateringer, avhengig av konfigurasjonen og om installasjonen oppfyller vilkårene for oppdatering. Eldre installasjoner beholder tidligere innstillinger med mindre de endres. PHP følger en egen tidsplan for oppdateringer.

Et skreddersydd bygg passer data og arbeidsflyter som ikke ligner sider og innlegg i det hele tatt, og redigeringsfunksjonene må bygges og vedlikeholdes. Et statisk bygg fjerner PHP-applikasjonen og databasen fra behandlingen av forespørsler, mens serveren og byggeverktøyene fortsatt trenger vedlikehold. Redaktørene jobber som regel gjennom filer og et byggesteg. WordPress passer den brede midten: et ganske standard nettsted som eierne skal drive selv, der redigeringsskjermene er verdt vedlikeholdet de fører med seg. Innlåsingen holder seg beskjeden, fordi koden er GPL, innholdet ligger i en vanlig SQL-database, og REST-API-et serverer innlegg, sider og taksonomier som JSON.

Slik ser det ut i praksis

I selskapets arbeid er WordPress ett svar blant flere, valgt med vilje. Oppsummeringen av hundre webapplikasjoner dekker 2003–2014 og sier rett ut regelen bak valget. Der en kunde trengte noe skreddersydd, ble det HTML/HTML5, CSS/SCSS og Django. Der en kunde trengte et mer standard nettsted som den også kunne administrere selv, kom WordPress eller Joomla raskere i mål og var det mer fornuftige svaret.

Det siden verdsetter, er skjønnet, ikke verktøyet. En del av arbeidet den beskriver, var å skille de to tilfellene fra hverandre: å få en kunde til å droppe en skreddersydd løsning den ikke trengte, og til å velge en når behovet var reelt.

Stilt som spørsmål en kjøper kan svare på før det bes om noe tilbud, ser skjønnet slik ut:

- Skal folk uten en utvikler redigere nettstedet hver uke, eller endres det noen få ganger i året?
- Passer innholdet i sider, innlegg, kategorier og medier, eller trenger det en egen datamodell og egne regler?
- Hvem skal holde kjernen, utvidelsene og PHP oppdatert så lenge nettstedet er i drift?
- Skal nettstedet lene seg på en håndfull godt vedlikeholdte utvidelser, eller på dusinvis av dem sydd sammen for å etterligne en applikasjon?

Ukentlige endringer, innhold som passer, en navngitt vedlikeholder og en kort liste over utvidelser peker mot WordPress eller et CMS av samme slag. Uvanlige data, eller ingen til å holde det oppdatert, peker mot et skreddersydd bygg eller et statisk.

Hvor det går galt

En rutinemessig oppdatering ødelegger nettstedet

Noen oppdaterer en utvidelse, og en side begynner å vise en feilmelding, eller administrasjonspanelet nekter å laste. Nettstedet hadde samlet utvidelser i årevis, noen av dem forlatt av forfatterne, hver av dem tredjeparts PHP som kjørte med full tilgang, og to av dem kolliderte. Hold listen kort, og slett hver utvidelse nettstedet har sluttet å bruke, slik herdingshåndboken ber om. Ta en sikkerhetskopi før hver oppdatering, prøv de store på en kopi først, og velg utvidelser der forfatterne fortsatt gir ut nye versjoner.

Egne endringer forsvinner ved en oppdatering eller et redesign

En justering som virket i månedsvis, forsvinner den dagen temaet oppdateres, eller en funksjon virksomheten er avhengig av, blir borte når nettstedet får et nytt utseende. I det første tilfellet redigerte noen filene i et tema de ikke hadde skrevet selv, og oppdateringen erstattet dem; temahåndboken advarer om at en funksjon som legges rett inn i `functions.php` i et tema, forsvinner ved neste oppdatering av temaet. I det andre tilfellet lå oppførsel som nettstedet var avhengig av, i temaet, så den forsvant sammen med temaet. Legg endringer i andres temaer i et undertema (*child theme*), og legg funksjoner i en liten egen utvidelse som kobler seg på kjernen, så temaet bærer presentasjonen og ingenting annet.

Et administratorpassord blir til kode på serveren

Den som stjeler passordet til én administrator, kan gå fra å redigere innlegg til å kjøre sin egen PHP, fordi administrasjonspanelet leveres med en redigerer for filene i temaer og utvidelser. Én linje i `wp-config.php` slår den redigereren av, noe som har samme virkning som å fjerne rettighetene `edit_themes`, `edit_plugins` og `edit_files` fra hver bruker:

```
define( 'DISALLOW_FILE_EDIT', true );
```

Linjen stenger én vei blant flere, og håndboken sier rett ut at den ikke stopper en angriper som kan laste opp ondsinnede filer. Kombiner den med sterke passord, totrinnsautentisering for hver administrator og så få administratorkontoer som arbeidet tillater.

Nettstedet sakker akterut

Administrasjonspanelet viser et banner om en ny versjon i månedsvis, mens nettstedet kjører en hovedversjon eller en PHP-versjon godt bak den gjeldende. Et nettsted som bare er satt opp for mindre kjerneoppdateringer, kan fortsette å lappe seg selv mens hovedversjonen og PHP sakker akterut, helt til en utvidelse det er avhengig av, krever noe nyere. Nye installasjoner fra 5.6 aktiverer også større kjerneoppdateringer som standard, men konfigurasjonen kan endre det. Sett de store oppdateringene i kalenderen, sjekk PHP-versjonen mot prosjektets side med krav samtidig, og test begge på en kopi før nettstedet i drift.

Les mer

- WordPress Documentation — prosjektets håndbok for å drive et nettsted, fra publisering og medier til installasjon og vedlikehold.
- Requirements — versjonene av PHP, MySQL og MariaDB som prosjektet anbefaler i dag.
- Upgrading WordPress — oppdateringer i bakgrunnen, med ett klikk og for hånd, og sikkerhetskopien som skal tas først.
- Hardening WordPress — filrettigheter, databaserettigheter, bryteren for filredigereren og hygiene for utvidelser.
- Plugin Handbook: Hooks — hvordan handlinger og filtre lar kode endre WordPress uten å røre kjernen.
- REST API Handbook — ruter, endepunkter og autentisering for API-et som blokkredigereren er bygd på.

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 WordPress?» 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/wordpress/