

SmartElement

Automatiserad anpassning av webbinnehåll

Staffan Enberg

INNEHÅLL

1	Inledning	5
1.1	Målsättning och syfte	5
1.2	Metoder	5
1.3	Avgränsningar	5
2	Bakgrund	6
2.1	Anpassat innehåll	6
2.2	Behov	7
2.3	Befintliga lösningar	7
3	SmartElement	9
3.1	Beståndsdelar	9
3.1.1	<i>Webbsidan</i>	9
3.1.2	<i>Element</i>	10
3.1.3	<i>Filter</i>	10
3.1.4	<i>Innehåll</i>	10
4	Teknik	11
4.1	PHP	11
4.2	MySQL	11
4.3	MongoDB	11
4.4	Memcache	12
4.5	JavaScript	12
5	Processer	13
5.1	Information om användaren	13
5.1.1	<i>IP Geolokalisering</i>	13
5.1.2	<i>Besökarstatistik</i>	13
5.1.3	<i>Tid</i>	13
5.1.4	<i>Hänvisningsinformation</i>	13
5.1.5	<i>Användardefinierad information</i>	14
5.2	Filtrering	14
6	Arkitektur	15
6.1	Back-End system	15
6.1.1	<i>Matcher</i>	15
6.1.2	<i>Site objektet</i>	15
6.1.3	<i>Element objekten</i>	15
6.1.4	<i>Filter objekten</i>	15
6.1.5	<i>Content objekten</i>	16
6.2	Javascript tagen	16

7	Sammanfattning och Diskussion	17
8	För uppgiften Utvidgad disposition	17
	Källor	18

ORDLISTA

DOM Document Object Model. 3, 6

JSON JavaScript Object Notation. 3, 6

JSONP Json-with-padding. 3, 6

1 INLEDNING

Detta arbete dokumenterar systemet SmartElement som utvecklades för företaget Developer's Helsinki Ab. Systemets syfte är att möjliggöra enkel konfiguration av anpassat innehåll för webbsidor, vilket tillåter webbsidors ägare att själva kunna konfigurera regler som bestämmer vad för innehåll som skall lyftas fram för olika användare.

1.1 Målsättning och syfte

Målsättningen med arbetet är att kartlägga den arkitektur som bäst stöder målet av ett effektivt system för innehålls filtrering och leverering, som även är lätt att vidare utveckla och utvidga i form av nya filter och nya algoritmer för filtrering.

1.2 Metoder

Systemet är i sig bara en motor för identifikation av användare och filtrering av innehåll. Det ända gränssnitt som SmartElement har är ett JSON-gränssnitt genom vilket ett skilt grafiskt gränssnitt tillåter användare att konfigurera sitt innehåll.

1.3 Avgränsningar

Detta är en redogörelse för den tekniska implementeringen inte användningen av anpassat innehåll på webbplatser, arbetet tar därmed inte ställning till hur anpassat innehåll skall användas, endast hur det kan implementeras.

Även om en av målsättningarna är att skapa ett effektivt system så finns det ingen klar standard för vad som kan anses vara bra prestanda för ett liknande system, därav kommer arbetet inte att innehålla grundliga prestanda analyser.

2 BAKGRUND

Projektet började våren 2013 som ett internt verktyg på Developer's Helsinki efter ett behov uppstått för ett system som skulle tillåta snabb implementering av personligt innehåll på kunders webbsidor.

Eftersom det i den tiden inte fanns någon färdig implementation att använda, bestämdes det att det skulle utvecklas en mjukvara först för internt bruk men med målsättningen att lanseras som en SaaS produkt.

2.1 Anpassat innehåll

Your Amazon.co.uk

Computers & Accessories



New Release
Kensington Orbit ...
★★★★☆ (202)
£18.99 **£17.50**
Why recommended?



PCSL / Adafruit Clear ...
★★★★☆ (19)
£3.49
Why recommended?



Clear Transparent top ...
★★★★☆ (55)
£12.99 **£4.80**
Why recommended?



Raspberry Pi Camera ...
★★★★☆ (12)
£24.99 **£19.00**
Why recommended?

See all recommendations in Computers & Accessories

Books



New Release
Don't Make Me Think: ...
Steve Krug
★★★★☆ (4)
£22.99 **£20.99**
Why recommended?



Clean Code: A ...
Robert C. Martin
★★★★☆ (36)
£31.99 **£23.99**
Why recommended?



Java Concurrency in ...
Brian Goetz
★★★★☆ (20)
£38.99 **£29.24**
Why recommended?



Growing ...
Steve Freeman
★★★★☆ (22)
£36.99 **£27.74**
Why recommended?

See all recommendations in Books

Figur 1. Amazons användarsida med anpassat innehåll

Anpassat innehåll, eller personligt innehåll (eng. *Personalized content*), är innehåll som väljs ut på basis av egenskaper hos användaren. I fallet av webbsidor kan det hand-

la om information som användarens geografiska läge, användarens språk, användarens webbläsare, antalet besök som användaren gjort till sidan, med mera. Tanken är att förse användaren med det den behöver eller vill ha utan att denne behöver be om det. (Mulvenna et al. 2000)

Anpassat innehåll har bland annat användts inom nätbutiker för att visa reklam anpassad för kunden på basis av dennes beställningshistorik, bilden 1 visar användarsidan som visas då man loggar in på amazons webb-butik. Inom social media har används anpassat innehåll för att lyfta fram innehåll som antas vara intressant för användaren (Uppdateringar från vänner som användaren ofta är i kontakt med, reklam från företag användaren har gillat m.m.). (Guy et al. 2010)

2.2 Behov

I takt med att stora aktörer införde allt mer anpassat innehåll så började även mindre kunder fråga om möjligheten att använda tekniken på sina webbsidor. Det visade sig att det inte fanns något företag på finska marknaden som erbjöd ett tillräckligt flexibelt system för anpassning av innehåll och därmed bestämdes det att Developer's Helsinki skulle utveckla ett sådant.

Systemet skulle kunna leverera innehåll dels i form av text (HTML eller rå text) för direkt injicering i webbsidans Document Object Model (DOM), dels i form av data i JavaScript Object Notation (JSON) format och dels som Json-with-padding (JSONP) svar vilket tillåter exekvering av javascript vid svaret. För filtrering skulle det samlas in data om beökares webbläsare och dator men det skulle även vara möjligt att tillägga egen data som sedan kunde användas för filtrering.

2.3 Befintliga lösningar

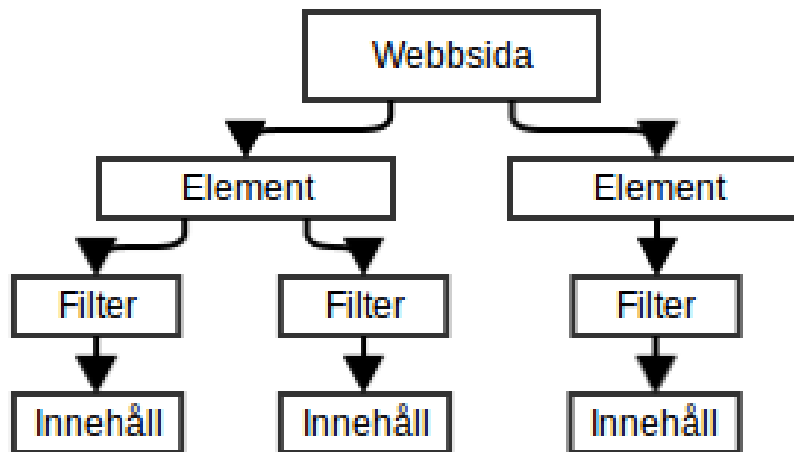
Vid tillfället för beslutet att utveckla produkten var den enda liknande lösningen på finska marknaden nosto.com som levererar personaliserat innehåll med inriktning på webbuti-

ker.(Nosto Website 2014) På grund av produktens klara vinkling passade den inte in i alla de användningsområden var SmartElement skulle användas.

3 SMARTELEMENT

SmartElement är ett back-end system som sköter filtrering av innehåll på basis av information som samlas in om besökare på en webbsida. Upprätthållare av en webbsida kan genom att integrera SmartElements JavaScript tag i sin sida, låta back-end systemet fylla i specificerade element med personligt innehåll.

3.1 Beståndsdelar



Figur 2. Exempel på en hierarki i SmartElement

SmartElement bygger på några grundläggande koncept; webbsidor, element, filter och innehåll. Med dessa fyra byggstenar konfigureras det objekt, visualiserat i bild 2, som back-end systemet använder för att filtrera och leverera innehåll.

3.1.1 Webbsidan

Webbsidan är det högsta elementet som SmartElement handskas med. Det representerar en hel webbplats, under vilken man kan definiera element och innehåll. Varje sida har en unik identifikationsnummer som används för att hämta ladda SmartElement tagen.

3.1.2 Element

Elementen är hållare för den data som returneras från back-end systemet och det som knyter innehållet till webbsidan. Elementet har i sig flere innehåll och filter, filtren används för att välja ut det innehåll som skall visas i elementet. Element har en inom sidan unik kod som används för att specificera vilka element som behöver processeras när man kallar på back-end systemet.

3.1.3 Filter

Filter består i SmartElement av flere olika regler som ställer en fråga om besökaren samt en länk till ett innehåll, vilket visas om filtret matchar. För att ett filter skall matcha krävs det att besökaren passar alla regler i filtret. För att hantera fall var två eller flere filter passar för en besökare har filter en prioritetsordning inom elementet, valbar av användaren.

3.1.4 Innehåll

Innehåll är den konkreta data som levereras för ett element efter att ett filter valts som vinnare. Innehållet kan bestå av text, HTML kod eller JSON data, det är upp till användaren att definiera vad för data som sparas.

4 TEKNIK

SmartElement bygger i stort sett på fem centrala tekniker, PHP som användts för att programmera programlogiken, MySQL som databaslager var data lagras, Memcache som caching lager vilket tillåter att information som är tung att beräkna kan sparas mellan förfrågningar, dokument databasen MongoDB för lagring av information om besökare och till sist JavaScript som används för att skapa det informationspaket om användaren som skickas till servern för användning vid filtreringen.

4.1 PHP

PHP är ett skriptspråk som utvecklats med målet att snabbt kunna skapa dynamiska webbsidor. (PHP Manual 2013) Språket valdes för att det är enkelt att arbeta med och det var språket som företagets andra produkter va skrivna med. Det finns också en mängd information och färdiga kodpaket till förfogande för programmerare, vilket underlättar och försnabbar utvecklingen.

4.2 MySQL

För lagring av data valdes databasmjukvaran MySQL. En relationell databas passar sig bra för lagring av den information som systemet behandlar eftersom det handlar om entiteter som är starkt länkade till varandra.

4.3 MongoDB

Dokumentdatabasen MongoDB används för lagring av användarinformation.

4.4 Memcache

Eftersom SmartElements datamodeller är relativt tunga att bygga så bestämdes det att ett caching lager skulle användas för att hålla dessa datamodeller och på så vis låta mjukvaran gå direkt till filtrerings logiken då en förfrågan behandlas.

Målet med cache lagret är att hålla sidorna i minnet på servern, och endast läsa från databasen när ett objekt byggs om i samband med en uppdatering. För detta ändamål valdes Memcache, en väletablerad mjukvara för caching.

4.5 JavaScript

För att kunna identifiera så mycket information som möjligt om användaren är det nödvändigt att köra kod på klienten innan innehållet kan presenteras. I dagens läge finns det endast ett skriptspråk som kan köras på godtycklig klient, och det är javascript, vilket valdes även för detta projekt.

5 PROCESSER

Processer för identifikation av användare och filtrering av innehåll.

5.1 Information om användaren

En genomgång av de olika typerna av information man kan samla in om användare samt fördelar och svårigheter med dessa.

5.1.1 IP Geolokalisering

Geolokalisering på basis av användarens IP-address. Vilket land kommer användaren ifrån, vilken stad, vilken region.

5.1.2 Besökarstatistik

Information om hur många gången besökaren återvänder, hur ofta denne sett sidan, allmän information om beteende.

5.1.3 Tid

Vad är den lokala tiden för sidan, vad är den lokala tiden för användaren, hur stor tidskillnad är det mellan de två, hur länge har användaren varit på sidan.

5.1.4 Hänvisningsinformation

Varifrån kom användaren, vilka sökord använde den eventuellt. Vilken sida visades först. Varifrån har användaren tidigare anlänt till sidan.

5.1.5 Användardefinierad information

Information som websidan själv tillhandahåller.

5.2 Filtrering

Hur kan man filtrera på basis av den insamlade informationen?

6 ARKITEKTUR

Förklaring av arkitekturen

6.1 Back-End system

Back-end systemet är det som gör den egentliga processeringen av information och väljer ut innehållet som levereras.

6.1.1 Matcher

Matchern är den kod som tar emot ett Visitor objekt som innehåller information om användaren och kör sedan igenom filtren för sidan med detta objekt.

6.1.2 Site objektet

Objektet som innehåller all information om webbsidan, elementen, filtren och innehållen.

6.1.3 Element objekten

Element objekten motsvarar ett HTML element på webbsidan, dessa har filter med vilka det kan definiera vilket innehåll det skall visa.

6.1.4 Filter objekten

Filter objekten innehåller regler som definierar en målgrupp och ett innehåll som det returnerar.

6.1.5 Content objekten

Content objekten är innehåll som returneras, detta kan vara dels HTML innehåll eller javascript kod.

6.2 Javascript tagen

Javascript tagen är en kort kod som laddas i samband med sidan och samlar ihop information om användaren och sedan skickar denna information till back-end systemet för processering.

7 SAMMANFATTNING OCH DISKUSSION

Sammanfattning av arbetet, tillbakatitt på produkten. Diskussion om förbättringar som skulle kunna göras i systemet. Diskussion om tekniker som kunde bytas ut för att eventuellt förbättra prestanda. Diskussion om svårigheter med produkten.

8 FÖR UPPGIFTEN UTVIDGAD DISPOSITION

Istället för att skapa ett skilt dokument valde jag att skriva använda mig av dokumentet som kommer att bli mitt slutarbete. Här kan man se en grund för arbetet och ett preliminärt skelett för dokumentet.

Första kapitlet är inte riktigt klar än men jag har försökt skriva ned några korta meningar under alla rubriker för att ge en bild av vad jag tänkt skriva om.

Har ännu inte hunnit skriva bygga en god källförteckning så jag valde att lägga till ett översatt citat från PHP manualen, närmast för att se att min källförteckning fungerar. Flera källor kommer att läggas till, till stor del handlar det om websidor eftersom en stor del av informationen om de tekniker som används finns tillgänglig på webben. Mer etablerad teknik har dock litteratur skriven om sig, så några böcker har även används, så som Schwartz et al. (2008) High Performance MySQL.

KÄLLOR

Guy, Ido; Zwerdling, Naama; Ronen, Inbal; Carmel, David & Uziel, Erel. 2010, Social Media Recommendation based on People and Tags, I: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, ACM New York, s. 194–201.

Mulvenna, Maurice D.; Anand, Sarabjot S. & Büchner, Alex G. 2000, Personalization on the Net using Web mining: introduction, *Communications of the ACM*, årg. 43, , s. 123–125.

2014, *Nosto Website*. Tillgänglig: <http://www.nosto.com>, Hämtad: 23.3.2014.

2013, *PHP Manual*. Tillgänglig: <http://fi2.php.net/manual/en/>, Hämtad: 2.3.2014.

Schwartz, Baron; Zaitsev, Peter; Vadim Tkachenko, Jeremy D. Zawodny; Lentz, Arjen & Balling, Derek J. 2008, *High performance MySQL*, 2 uppl., O'Reilly Media, Ltd.