

Dokumentation: DateOne

V0.6

Simon Williamsson

Sasa Ristic

2012-12-10

Examensarbete, WDK11

Innehållsförteckning

Dokumentation: DateOne.....	1
Vad är DateOne?.....	3
Affärsnyttan då?.....	3
Installation.....	3
Tekniska krav.....	3
Kända problem med installationen.....	3
Installation.....	4
Rent tekniskt ”mumbojumbo”.....	4
Versionshantering.....	4
Vad är DateOne byggt i?.....	4
Finns det något API?.....	4
Sajten är flerspråkig, hur lägger jag till ett till språk?.....	5
Säkerheten då?.....	5

Vad är DateOne?

DateOne är en dejtingapplikation på webben där användare kan fylla i information om sig själva för att matchas mot andra användare.

DateOne erbjuder även ett evenemangssystem som tillåter användare att skapa evenemang markerade på en google maps karta, som exempel:

Kalle vill gå ut och ta en öl i helgen men har ingen att gå med. Han lägger då upp ett evenemang i Stockholm i ungefär rätt område, lägger till en beskrivning och vilka datum och tider som gäller, och väljer sedan en "notifieringsdiameter" (namnet är ett work in progress).

Alla användare inom den diametern skall då få ett mail eller ett meddelande, beroende på deras inställningar, som berättar om evenemanget.

Observera att notifieringssystemet inte är helt implementerat, däremot går det att hämta ut alla användare inom den, ungefärliga, diametern.

Utöver detta är syftet att DateOne skall innehålla ett chattrum alla kan delta i, en sökmotor för att hitta användare inom sina preferenser, profilbilder, bildgallerier och eventuellt också videogallerier. Detta har tyvärr inte hunnits med, men det kommer byggas på senare.

Användare skall kunna uppgradera sitt konto till Premium, dessa skall ha exklusiv tillgång till:

- Skicka och läsa meddelanden
- Lägga till vänner
- Ladda upp bildgallerier
- Eventuellt kunna gå in i privata chattrum med en eller flera andra användare, istället för det publika chattrummet

Det har inte hunnits med, men applikationen är designad med detta i åtanke, och Simon har tidigare erfarenhet av att integrera PayPal på en webbsida, så detta kommer inte vara några problem att implementera.

Affärsnyttan då?

Lyckas man övertala personer att gå över till DateOne och att använda den kan man enkelt dra in pengar på premiumkonton, och det går alltid att lägga till fler betal- och tilläggstjänster i efterhand.

Installation

Tekniska krav

DateOne kräver:

- Curl, för att ta reda på en användares koordinater via Google Maps API
- Mod_rewrite, för att skriva om URLer på ett fint sätt. Skall i framtiden inte vara ett krav.
- PHP 5.2 eller bättre
- Användare måste tillåta javascriptkod.

Kända problem med installationen

Det verkar inte gå att köra DateOne via MAMP, varför har vi inte lyckats ta reda på.

Vi har inte haft möjlighet att försöka installera DateOne på en linux, dvs LAMP. Vi vet inte om det

fungerar.

DateOne går att installera på en WAMP server.

Installation

Skapa en databas kallad DateOne med koallitionering utf8_bin, välj den, och importera filen dateoneDB.sql till den, detta skapar alla tabeller som behövs, de som behöver information i sig har det från början.

Placera sedan DateOne mappen i din www folder, eller på din webbservers rootnivå, och gå in i ../DateOne/application/config/database.php och fyll i korrekta uppgifter för att ansluta till databasen på rad 51, 52 och 53.

Installationen är nu klar, och sajten är redo att användas!

Databasen kommer med ett fördefinierat konto, DateoneAdmin, med lösenord D2!v#5=1xi.

Detta konto kommer användas av systemet för att skicka meddelanden till användare, därför skall detta inte tas bort. Det går bra att logga in med detta för att se sig om dock.

Rent tekniskt "mumbojumbo"

Det här avsnittet är mest för framtida utvecklare på sajten.

Versionshantering

Ta en titt här: <https://github.com/Williamsson/DateOne>

Vad är DateOne byggt i?

För backend delen valde vi ramverket Codeigniter. Dels för att Simon hade vissa förkunskaper, dels för att det verkade fylla alla våra behov, vilket det gjorde.

I frontenden används JQuery, JQueryUi och ett par JQuery plugins.

För den flytande menyn använder vi

www.jtricks.com/javascript/navigation/floating.html

När det gäller datum/tidsväljaren använder vi

<http://trentrichardson.com/examples/timepicker/>

Anledningen till de två är först och främst åsikten att "varför ska man återuppfina hjulet?" men till viss del också tidsbrist.

Finns det något API?

Ja, det gör det. Det är ett RESTFUL API som finns i /application/controllers/api.php.

Tanken är att detta API endast skall användas internt av applikationen och inte utifrån, eventuellt kan man senare öppna upp för att hämta ut till exempel evenemangsinformation osv, men för tillfället skall det vara intern användning endast.

Det är för tillfället inte låst, så det kan nog gå att komma åt det utifrån. Det kommer det inte göra när tjänsten väl lanseras. Det finns redan nu möjligheter att ange användarnamn och lösenord för att nyttja APIet, det är bara inte gjort ännu för att kunna debugga enklare.

APIet är konstruerat med hjälp av <https://github.com/philsturgeon/codeigniter-restserver>

APIet används för att dynamiskt hämta, skicka, och presentera information för användaren via AJAX. Meddelandesystemet tillexempel använder sig av APIet, evenemangssystemet likaså.

APIet accepterar anrop med parametrarna lite hur som helst, t.ex. kan ett POST anrop till /api/event ha parametrarna /markerLong/18.65/description/massatext/eventLat/59.47, eller sorterade på ett helt annat sätt.

Det stödjer att skicka tillbaka information i JSON, XML, osv. Se <http://net.tutsplus.com/tutorials/php/working-with-restful-services-in-codeigniter-2/> för mer information om just APIet.

För att skicka anrop till APIet använder du URLen `dateone.se/se/api/`

APIet är inte helt implementerat, det lades till sent i utvecklingen tyvärr. Framöver kommer mer och mer av funktionaliteten flyttas till det.

Sajten är flerspråkig, hur lägger jag till ett till språk?

Först går du till `../application/languages/` och skapar en mapp, vi kallar den för Y, med namnet på språket, tillexempel "english" eller "swedish". Där skapar du en php fil som heter `X_lang.php`.

Gå sedan till metoden `getLanguageList()` i `../application/models/language_model.php` och lägg till X i den arrayen.

I metoden `loadLanguage()` lägger till en ny `elseif($lang == "X")` där X är detsamma som ovan.

Gå sedan till `getLanguage()` och lägg till en ny `elseif($this->uri->segment(1) == "X")`{

`$lang = X; $this->session->set_userdata('user_country_initials',$lang);`

Slutligen kopierar du innehållet i `../language/swedish/se_lang.php` till språkfilen du skapade i `languages/Y` mappen, och sätter sedan igång med att översätta.

Men hur hämtar jag text för att visa på sajten då?

Enkelt, använd bara `label('nyckel',$this);` för att hämta texten tillhörande nyckeln i den för tillfället laddade språkfilen.

Det här kommer från filen `lang_translate_helper` i `../application/helpers` mappen, som alltid laddas.

Och hur laddar man en språkfil?

I vanliga fall skickar du användarna till en ny sida via `$this->general_model->changeView($title, $view);` och den sköter språkladdningen av sig själv. Däremot finns det vissa fall då du inte vill använda den, utan behöver göra en egen lösning.

`$title` står för vad `<title>` taggen skall innehålla.

Då ropar du helt enkelt på `$this->language_model->loadLanguage();` så sköter den resten själv, och du kan återgå till din speciallösning.

Säkerheten då?

Alla lösenord hashas med `sha1`, och saltas med ett användarspecifikt salt på 30 tecken, dessa finns i `safety_model` men anropas från `user_model->register()` och `login` osv.

Vissa sidor får en användare inte titta på utan att vara inloggad, så det finns en metod för det med. I början på controllerns funktion som avgör vilken sida det är använder du

```
if(!$this->user_model->isLoggedIn()){  
    $this->redirect_model->redirect('gohomepage');  
}
```

Som helt enkelt kontrollerar om användaren som gjorde anropet är inloggad, om inte kastar den tillbaka den till startsidan. Vill man kan man lägga till ett felmeddelande med `$this->session->flashdata`.