

Webbteknik II - 1DV449

Projekthandledning

John Häggerud

john.haggerud@lnu.se

Johan Leitet

johan.leitet@lnu.se

December 2012

Inledning

Som avslutning av denna kurs ska Du skriva en komplett applikation som ska sammanfatta vad du lärt dig under kursens gång samt förhoppningsvis ge dig ytterligare kunskap och erfarenhet av att jobba med webb-API:er, egna som andras. Projektet är också tänkt att ge dig kunskap om nya och relaterade HTML5-API:er för JavaScript-klienter.

Vår förhoppning är att du i detta moment utvecklar en egen applikationsidé och som du utan problem skulle kunna visa upp för tilltänkta arbetsgivare när du söker jobb. För er som inte klarar att tänka ut en egen applikationsidé kommer det finnas ett mer eller mindre fördefinierat problem, men det är klart önskvärt att du låter din egen kreativitet flöda här.

Uppgiften

1.1 En mashup-applikation

Det är dags att ge sig i kast med en större projektuppgift i denna kurs och tanken är att Du ska skapa en mashupapplikation av något slag. Vilken typ av applikation ser vi helst att ni kommer fram till själv och vi ser positivt på egen kreativitet, egna initiativ och utmaningar. Kanske kan du jobba vidare på idéen du hade på seminarium 2? Kanske har du en annan idé du vill utveckla? **Du börjar med att beskriva din idé genom att**

skicka in max en A4 som beskriver din idé. Gör detta så fort som möjligt så du kan sätta igång med arbetet. Senaste datumet för inlämning av idé framgår av punkt 1.4.

1.2 Krav på applikationen för godkänt (betyg 3)

Vi kommer alltså ha en fri hållning till vad för information och data du jobbar med i din applikation men det finns ett antal krav vi kommer ha på din applikation för att uppnå godkänt (betyg 3) på momentet:

- **Krav på externa API:er**

En mashup handlar ju om att kombinera olika typer av data, oftast från externa webb-API:er. Detta krav kan du uppfylla på två olika sätt:

- Bygg en applikation som främst hämtar data från externa webb-API:er. Det ska i detta fall vara **minst två stycken** externa webb-API:er som sammankopplas på ett dynamiskt sätt.

- Bygg en applikation som stödjer sig på ett **eget skapat webb-API** och kombinerar detta med **minst ett** externt webb-API .

Tänk igenom så att Du felhanterar eventuella fel som kan uppstå t.ex. vad som ska hända ifall det externa webb-API:et inte svarar.

- **Användarhantering och personalisering**

Din applikation ska innehålla någon form av inloggning hanterad av tredjepartslösningar eg. OAuth, OpenID eller liknande. Du är fri att välja vilken “provider” av detta du vill, förslagsvis twitter eller facebook. Detta kan du använda för att t.ex. personalisera din applikation för användaren genom att knyta information till en specifik användare..

- **Local storage**

Applikationen ska använda sig av “local storage” på klienten. Exempelvis kan du spara ner din klients aktuella “state” i local storage så att när användaren avslutar sin session och senare påbörjar en ny så kan man fortsätta där man slutade. Det kan också finnas data som behöver cachas på klienten och då kan local storage vara ett utmärkt sätt att göra detta på. (Med cachning menas här att spara data som är intressant för applikationen. Exempelvis tidigare sökningar etc. Inte cachning i form av bilder, script etc. Detta kan du göra med Offline Web Applications istället)

- **Stabilitet och design**

Din applikation ska självklart fungera utan några buggar eller ohanterade felmeddelanden.

Det ska också synas att du arbetat igenom den grafiska designen för att få en lättförstådd och tilltalade applikation. Och nej, ogenomarbetad och fantasilös design är inte lika med enkelt och stilrent.

- **Rapport alt. blogginlägg**

Din applikation ska beskrivas antingen i form av en inlämnad rapport eller via ett blogginlägg. Rapporten ska vara i pdf-format. Rapporten eller inlägget presenteras senast någon dag innan leverans och slutlig redovisning. För mer information om datum se punkt 1.4. Mer information om rapportens utformande beskrivs i punkt 1.6.

1.3 Eventuellt betygshöjande funktioner (betyg 4 och 5)

För högre betyg än godkänd (G) ska applikationen redovisas innan det leveransdatum som finns (se punkt 1.4)

Vi ser gärna att ni utmanar er själva och testar nya grepp och tekniker vilket kan ses som betygshöjande. Här är några förslag som kan höja din applikation och betyg:

- Använd OAuth tillsammans med möjligheten att även hämta information om dina användare och använd detta i din applikation. Låt applikationen twittra eller uppdatera användarens facebook-status vid speciella händelser i applikationen.
- Inkludera flera HTML5-api:er så som geolocation, offline web browsning, svg e.t.c.
- Inkludera flera externa API:er och höj applikationens funktionalitet ytterligare.
- Fundera över cachning. Kan man optimera din applikation via cachning och mellanlagring av data?
- Skriver du ett eget webb-API bör du dokumentera detta på ett bra och tydligt vis så att andra kan använda det.
- Arbeta under projektets gång med Github och låt oss ta del av dina incheckningar av kod. Se punkt 1.7 för mer information om github.

1.4 Bedömning

Betyg kommer sättas på din applikation med avseende på följande:

- Funktionalitet - Hur mycket funktioner har din applikation, vilket mervärde ger de inkluderade externa API:er som används. Hur väl fungerar applikationen?
- Design och helhetskänsla. Känns din applikation genomarbetad och ser den proffsig ut?
- Kvalité på rapport. Ger din rapport en bra bild av din applikation och hur väl har du reflekterat över eventuella problem och lösningar.
- Presentationen. Är presentationen av applikationen proffsig och väl förberedd.
- Uppfyllande av leveransdatum (se punkt 1.4).
- Lärdomar utanför laborationskurs. Påvisar din applikation och rapport att du inkluderat och lärt dig tekniker som inte tagits upp i laborationskursen.
- Egen kreativitet. Är din eventuella applikationsidé nytänkande och intressant.

Glöm inte nämna de punkter du tycker är betygshöjande i din rapport. Lista gärna upp dem i punktform.

1.4 Viktiga datum

- **Din projekttidé presenteras senast fredagen den 21/12 klockan 13.00.** Du bör dock vara ute tidigare än så för att ligga bra till i din planering och utveckling.
- **Anmälan för muntlig redovisning sker via mejl till kursansvarig senast måndagen den 14/1 klockan 23.59.** Utifrån detta mejl ges ni en redovisningstid antingen på torsdagen eller på fredagen.
- **Din rapport skickas in till kursansvarige senast onsdagen den 16/1 klockan 12.00.**
- **Leverans- och muntlig redovisning av din applikation sker senast torsdagen den 17/1 alt. fredagen den 18/1.** Vi återkommer med tider efter att anmälningarna gjorts.

1.5 För er utan egna idéer

Här följer ett exempel på applikation för er som inte kan komma på en egen idé och som inte satsar på högre betyg än godkänt (3):

- Skriv en applikation som utgår från en position (latitude och longitude).
- Koppla ihop positionsdata med två externa API:er som kan ge positionsspecifik data så som t.ex. väderdata, fototjänster, twitter, sevärdheter, wikilocation e.t.c. Det vill säga; Vad kan vi få för information om din position.
- Gör en inloggning enligt kraven och ge användaren möjlighet att spara undan sin position över sessioner på servern.
- Använd local storage till att t.ex. spara undan senaste position så att denna markeras direkt när användaren startar applikationen

1.6 - Rapporten/Blogginlägg

Rapporten/inlägget du ska redovisa ska innehålla följande:

- **Inledning** där du beskriver vad du gjort och bakgrunden till din applikation
- **Serversida:** Beskriv hur din applikation fungerar på serversidan. Har du arbetat med cachning- i så fall hur? Hur hanterar du eventuella fel som kan uppstå i applikationen?
- **Klientsida:** Hur fungerar din applikation på klientsidan. Beskriv funktionaliteten och de tekniker du använt.
- **Egen reflektion kring projektet:** Här tar du upp det som hänt under arbetes gång. Vad har tagit mest tid, vilka eventuella problem har du stött på? Finns det funktioner du hade velat implementera men som inte hanns med? Hur skulle du vilja jobba vidare med denna applikation?

- **Saker du anser vara betygshöjande (om du anser dig ha sådana)** : Beskriv de funktioner som du implementerat (bortsett från de krav som fanns) som du tycker är betygshöjande. Motivera varför du tycker du ska högre betyg än godkänt (3).

Rapporten ska skickas in till kursansvarige i pdf-format.

1.6 Tips

Det finns många CSS-ramverk som kan underlätta ert designarbete.

Undersök om det finns färdiga bibliotek för autentisering via tredjepartslösningar.

Dyk upp på handledningspassen så vi kan hjälpa dig om du kör fast!

Campus-studenter - Kom till skolan och arbeta här för att enklare kunna utbyta idéer och tankar!

Använd diskussionsforumen för att ställa frågor samt att hjälpa varandra.

Inloggning via facebook: <https://developers.facebook.com/docs/howtos/login/server-side-login/>

Börja med rapporten tidigt och håll den uppdaterad under hela projektet.

1.7 Om github

Github (<http://www.github.com>) är en tjänst som baserar sig på versionshanteringssystemet Git. Github är en tjänst där man kan versionhantera sin kod och samtidigt dela den med andra. Github är en allt vanligare tjänst när man jobbar som utvecklare och vi har våra planer på att inkludera den på mer regelmässig basis i undervisningen och i slutändan kombinera ihop tjänsten med CoursePress.

Vi vill redan nu i denna kurs köra en test på detta och ser gärna att Du använder tjänsten för att versionhantera din kod samtidigt som ni kan dela den med oss lärare på ett smidigt sätt. Observera att vi med all säkerhet kommer använda github i kommande kurser så det är bara fördelar att göra det redan i detta projektet, det är dock inget krav.

Har du redan ett github-konto kan du såklart använda det ifall du vill. Tänk dock på att döpa ditt delade repository har ett sådant namn att vi direkt kan se vem som äger det.

Som student får man möjlighet att skapa ett så kallat mikrokonto där man har möjlighet att ha upp till 5 stycken privata repositories. Som vanlig användare måste man betala för privata repositories.

Detta skaffar du dig via: <http://www.github.com/edu>

Välj "I'm a student" och använd din student-epost du fått via LNU.

Ditt användarnamn ska helst vara enligt formen xx222xx_LNU, där xx222xx är ditt användarnamn.

Du bör ganska fort få svar från github och kan logga in via <http://www.github.com>

Följ guiden för att sätta upp git på: <https://help.github.com/articles/set-up-git>.

Se till att installera github-klienten för ditt operativsystem.

Skapa ett privat repository där du ska hantera din kod.

Bjud in användarna *thajo* och *leitet* till ditt skapade repository.

Börja att köra en commit på din kod. Se till att commit:a kod hela tiden under projektet så att vi kan följa er utveckling!

Lycka till!