



Örebro Universitet, Handelshögskolan  
Klientprogrammering för webbsystem, HT16  
Ann-Sofie Hellberg, Johan Aderud  
2016-08-29

## Projekt: Black Eagle Games

### Introduktion

Syftet med detta projekt är att testa era kunskaper om klientprogrammering för webbapplikationer med HTML, CSS, JavaScript och jQuery. Följande regler gäller för projektet:

- Projektet skall genomföras i **grupper om två personer**.
- **Redovisning** av projektet är **fredagen** den **23:e september**.
- Som slutbetyg på projektet ges betyg godkänt (G) eller underkänt (U). Rest kan ges en gång på projektet, om resten inte är godkänd får studenten göra om projektet nästa gång kursen ges. Tid för presentation av resten bestäms efter första redovisningen.
- Innan redovisningen ska projektet ha packas ihop (zip eller rar) och skickas till **Ann-Sofie Hellberg** och **Johan Aderud** via **Blackboard-meddelande**. **Deadline för inlämning är torsdagen den 22:a september kl. 12:00**. Sent inskickade projekt kommer inte bedömas under ordinarie examination utan ni får då vänta till om-examinationen av projektet.
- Ni kommer att muntligt få redovisa er applikation genom att dels visa den för examinerande lärare samt vara beredda på att svara på frågor om er applikation under redovisningen. Det ligger på projektgruppens ansvar att se till att projektet är redovisningsbart vid redovisningstillfället.
- Då denna kurs endast fokuserar på klientsidan så tillhandahålls serversidans funktionalitet under domänen <http://ofcourse.oru.se/~IK2009/>.

### Projektbeskrivning

Ni har fått i uppgift att skapa en webbplats åt företaget Black Eagle Games, som är en framgångsrik, oberoende spelutvecklare. Företaget utvecklar dator- och TV-spel inom olika genrer finansierade av ledande spelförlag.

### JSON-data

Vissa anrop till servern ska göras med jQuerys `getJSON`-metod. Servern svarar med ett JSON-objekt som innehåller minst två fält: `status` och `message`. `status` innehåller en siffra, som är 0 om frågan misslyckades och 1 om den lyckades. `message` innehåller ett felmeddelande vid fel. Om förfrågan lyckades (`status == 1`), innehåller JSON-objektet dessutom fältet `result`, som är ett JavaScript-objekt med resultatdata (detta fält kallas nedan "resultobjekt").

Om vi hade en JSON-tjänst på [http://ofcourse.oru.se/json/get\\_user.php](http://ofcourse.oru.se/json/get_user.php) som tog parametern `userid`, och som returnerade ett resultatobjekt med fältet `username`, skulle vi alltså anropa den på följande sätt:

```
$.getJSON(  
    'http://ofcourse.oru.se/json/get_user.php',  
    {userid: 10},  
    function (data) {  
        if (data.status == 1) { // Lyckades  
            // Resultat:  
            $('#output').text(data.result.username);  
        } else { // Misslyckades  
            // Felmeddelande:  
            $('#output').text(data.message);  
        }  
    }  
);
```

Eller, om tjänsten returnerade en array med alla användare om man inte angav userid:

```
$.getJSON(  
    'http://ofcourse.oru.se/json/get_user.php?callback=?',  
    function (data) {  
        if (data.status == 1) { // Lyckades  
            for (var i = 0; i < data.result.length; i++) {  
                $('#output').append(  
                    data.result[i].username + '<br />'  
                );  
            }  
        } else { // Misslyckades  
            alert(data.message); // Felmeddelande  
        }  
    }  
);
```

Obs! Detta är endast exempel för att tydliggöra hur ni ska lägga upp koden, tänk på att byta ut url:en samt eventuella parametrar!

## Krav

- All JavaScript och CSS ska skapas i separata filer.
- Sidan ska använda responsiv-design; den ska anpassa sig till två vyer (för en smartphone och en webbläsare på PC/Mac).
- All kod måste vara kommenterad till den grad att det icke självklara är beskrivet.
- Formatering av er webbsida (teckensnitt, positionering, osv.) ska ske med CSS.

## Webbplatsen ska bestå av följande sidor:

### 1. Förstasidan (Hem)

På första sidan ska visas: **en välkomsttext**, samt **ett bildspel som växlar mellan olika bilder**.

### 2. Om företaget

Denna sida ska innehålla **en informationstext om företaget** (ta någon text från Internet eller skriv själv ihop en text). Dessutom ska finnas **en världskarta som visar alla Black Eagle Games kontor. När**



användaren klickar på ett kontor på kartan, ska kontorets besöksadress och telefonnummer hämtas med jQuery, och visas i ett element bredvid kartan (ni skall även hantera anrop som misslyckas, dvs., som returnerar ett meddelande med status 0). Det skall även finnas en möjlighet att visa en lista över samtliga kontor.

Information om kontor hämtas med jQuerys `getJSON`-metod från URL:en

[http://ofcourse.oru.se/~IK2009/json/get\\_office.php](http://ofcourse.oru.se/~IK2009/json/get_office.php), och när den anropas utan parameter returnerar den en samling av "stads-objekt". Objektet som returneras innehåller fälten `name` (som innehåller kontorets beteckning), `address` (kontorets gatuadress), och `phone` (telefonnummer till kontoret).

Om URL:en anropas med en parameter, `city`, så kommer det istället för en samling bara ett objekt, den valda staden. Anropet sker genom att parametern anges som en querystring i formen `?city=orebro`. Tänk på att alla städer skrivs med små bokstäver och aao istället för åö!

*Black Eagle Games har kontor i Örebro och Göteborg i Sverige, Sheffield i England, Toronto i Kanada, Snoqualmie i USA, och Kyoto i Japan.*

### 3. Spel

En sida som listar alla **Black Eagle Games** spel. (bilder finns i arkivet som finns att ladda ner från Blackboard. Texter skriver ni själva, alternativt tar en text från Internet)

### 4. Kontakt

En sida där användare kan skicka in frågor till Black Eagle Games. Sidan skall innehålla ett formulär där användaren ska skriva i sin e-post adress, ett ämne och en fråga. E-posten skall ha rätt format för en e-postadress (detta fält skall valideras med hjälp av Regular Expressions), ämne och fråga får ej vara tomma. Valideringen skall göras med hjälp av JavaScript.

Eftersom det är valideringen som är det viktiga på den här sidan finns koden för AJAX-anropet nedan:

```
$('#submit').click(function() {
    $.ajax({
        url: 'http://ofcourse.oru.se/~IK2009/contactform/contact_form.php',
        type: 'GET',
        success: function(msg) {
            alert(msg);
        },
        error: function() {
            alert('Nu var det något som gick fel :(');
        }
    });
});
```

För att detta ska fungera måste det finnas en knapp (knappen måste ha `input type="button"`) i formuläret med id-t `submit` (alternativt ändra id-t i koden ovan). Om anropet lyckas kommer det dyka upp en popup-ruta som säger "Tack för ditt meddelande". Er uppgift är alltså att se till att innan detta anrop görs, måste användaren ha fyllt i uppgifterna på rätt sätt. Om inte, så ska användaren få veta vad som är fel. När meddelandet har skickats ska formuläret tömmas. Tips: detta sker i den anonyma funktionen som kallas vid success, i samband med `alert(msg)`.

## 5. Nyheter

På denna sida skall det finnas ett nyhetsflöde där Black Eagle Games postar nyheter om spel, företaget osv. Företagets nyhetsflöde ligger externt som en XML-fil och er uppgift är att presentera (med hjälp av jQuery) innehållet i XML-filen för användaren.

XML-filen kommer ni åt på: [http://ofcourse.oru.se/~IK2009/xml/get\\_news.php](http://ofcourse.oru.se/~IK2009/xml/get_news.php)

## Därutöver ska alla sidor innehålla:

En navigationsmeny för att ta sig mellan sidor.

## 6. För grupper med tre personer

Om er grupp består av tre personer tillkommer dessa krav på hemsidan:

- Det ska finnas en sida där alla lediga jobb listas. Dessa hämtas i JSON-format från [http://ofcourse.oru.se/~IK2009/json/get\\_jobs.php](http://ofcourse.oru.se/~IK2009/json/get_jobs.php). Även här används parametern `city` för att plocka ut data för en viss stad. Skriv in adressen i webbläsaren för att se JSON-objektet och vilken information som finns om varje jobb.
  - o Listan på jobb ska designas så att användaren bara ser namnet på staden och den lediga positionen när sidan laddas. Det ska sedan vara möjligt att klicka på positionen för att visa ytterligare beskrivning av tjänsten. Genom att klicka på positionen en gång till ska informationen döljas. Tips: använd jQuery.
- All HTML ska valideras enligt W3Cs standard för HTML5. (Den ska inte ge några error, men varningar är ok.) För att validera koden kan exempelvis följande verktyg användas: <http://validator.w3.org/>

## Övrigt

- Tänk på design. Det är inget krav att er webbplats är grafiskt väldesignad, men ansträng er för att göra det "så bra som möjligt".
- Tänk på tillgänglighet! Även om det inte finns några formella krav på att sidan skall uppfylla några tillgänglighetskrav så skall ni beakta de riktlinjer som presenterats tidigare i kursen.
- Hantera er tid. Vänta inte tills sista dagarna med att göra projektet!
- Bilder. Det finns mängder med gratis bilder på Internet som ni kan använda för projektet.
- Använd Internet! Om det är någon information som finns i överflöd på Internet så är det guider/tips/tutorials osv. om hur man skapar webbsidor. Kopiera dock inte lösningar rakt av då du måste kunna förklara lösningarna under redovisningstillfället!
- Plagiera inte andra gruppers uppgifter! Misstänkta plagiat kommer att anmälas till disciplinnämnden.
- Handledning i datasal ges under de tider som finns i ert schema. Har ni frågor går det även bra att kontakta lärarna.
- Tänk på att "snygga" till projektet innan det skickas in! Vi vill inte få in filer/funktioner som inte hör till projektet.