



AKADEMIN FÖR TEKNIK OCH MILJÖ  
Avdelningen för industriell utveckling, IT och samhällsbyggnad

---

# Grafer för Momentum

Martin Thörnqvist

Maj 2013

Examensarbete, Grundnivå (högskoleexamen), 15 hp  
Datavetenskap  
Examensarbete för Internetteknologi  
Internetteknologi

Handledare: Carina Pettersson  
Examinator: Atique Ullah

---

# Grafer för Momentum

av

Martin Thörnqvist

Akademien för teknik och miljö  
Högskolan i Gävle

S-801 76 Gävle, Sweden

Email:

*nit11mtt@student.hig.se*

## **Förord**

Jag vill tacka min handledare Elias på Momentum för all hjälp jag har fått för att utföra detta arbete.

## **Abstrakt**

Jag har gjort grafer som visar förbrukning åt kunder i Momentums marknadssystem. I rapporten beskriver jag hur det har gått och vad jag lyckats åstadkomma. Målet var att förenkla för kunderna när de ser över sin förbrukning. Arbetet är gjort i största del i JavaScript och använder Highcharts JS till graferna. Resultatet blev lyckat.

# Innehåll

<b>1 Inledning</b> .....	<b>1</b>
1.1 Bakgrund .....	1
1.2 Syfte .....	1
1.3 Metod .....	1
1.4 Fakta om företaget .....	1
1.5 Min handledare.....	2
<b>2 Genomförande</b> .....	<b>2</b>
2.1 Teknik.....	2
2.2 Utveckling .....	2
<b>3 Slutprodukt</b> .....	<b>4</b>
<b>4 Diskussion</b> .....	<b>4</b>
<b>5 Resultat</b> .....	<b>4</b>
5.1 Slutsats.....	5
<b>6 Referenser</b> .....	<b>6</b>
<b>7 Bilagor</b> .....	<b>6</b>
7.1 Bilder .....	6
7.2 Kod.....	8

# 1 Inledning

## 1.1 Bakgrund

Jag gjorde mitt arbete hos Momentum och det var där jag tidigare gjorde min praktik. Momentum är ständig ute efter utveckling av sina produkter. Anledningen till att jag gjorde detta arbete började med att jag var med på deras årsredovisning och där snappade jag upp vilka funktioner som Momentums VD skulle vilja se i marknadssystemet, ett av det han ville se var någon förenkling av att kunna analysera förbrukning på "Min sida" som kunder använder sig av. Det vill säga förbrukning av vatten och el. Jag tyckte detta arbete verkade vara något för mig och när det sedan började närma sig examensarbete frågade jag om jag kunde få göra detta arbete och det fick jag.

## 1.2 Syfte

Mitt arbete på Momentum har gått ut på att effektivisera och förenkla för kunder gällande deras förbrukning på vatten och el. Att med hjälp av grafer lättare kunna jämföra tidigare förbrukning och få en översikt av hur mycket man förbrukar. Även att göra det snyggt för att förhöja användarupplevelsen.

Syftet var att ta med mig det jag har lärt mig och sätta det i bruk och göra ett arbete i det riktiga arbetslivet.

## 1.3 Metod

Jag har gjort arbetet på min bärbara dator och jag har både jobbat hemifrån och på Momentums kontor i Gävle. Med dagens hjälpmedel såsom internettelefoni, skärmdelning och chatt har det inte varit några problem att jobba hemifrån. I arbetet har det förutom programutveckling även ingått avstämningstelefonmöten och planering mot min handlare för att säkerställa leveransen. I mötena har vi använt oss av TeamViewer som är en programvara för skärmdelning och Skype för internettelefoni.

Angående problem som har uppstått så har jag frågat min handledare och utvecklarna på kontoret men även så klart använt internet för att leta fram exempel, läst dokumentationer och tutorials.

## 1.4 Fakta om företaget

Momentum har kontor i Falun, Malmö, Stockholm, Uppsala och Gävle med totalt ca 50 anställda. De är en ledande leverantör av programvaror som effektiviserar och förenklar för fastighetsförvaltare. Det erbjuder ledande produkter skräddarsydda efter kundens behov.

På kontoret i Gävle där jag gjorde mitt examensarbete jobbar det 4 stycken systemutvecklare. Detta kontor är nytt och dom jobbade tidigare på kontoret i Falun. På kontoret i Gävle jobbar alla som systemutvecklare.

Momentum betyder rörelseenergi.

[www.momentum.se](http://www.momentum.se)

## 1.5 Min handledare

Min handledare heter Elias Chaaya och han har jobbat som systemutvecklare på Momentum i lite mer än 3 år. Han har varit väldigt hjälpsam under mitt arbete och han har lärt mig många nya saker. Han har varit en riktig bra handledare helt enkelt.

## 2 Genomförande

### 2.1 Teknik

Arbetet skulle skrivas på klientsidan i JavaScript. Så i största del bestod arbetet i form av kodning i JavaScript och lite HTML, i dessa har jag sedan tidigare kunskaper inom och jag tycker det är kul att jobba med JavaScript. Till själva graferna finns färdiga bibliotek, så ett bibliotek för graferna fick jag leta upp. Datat som skulle fylla graferna fanns det skrivna SQL-frågor för att hämta ut, SQL användas för hantering av data i databaser. Jag använde mig sedan av ett JSON-objekt till att jobba med datat. ”JSON JavaScript Object Notation, är ett kompakt , textbaserat format för datorer som används för att utbyta data”[6]. Jag har använt mig mycket av jQuery som är ett JavaScriptbibliotek och det är ganska enkelt och roligt att jobba med. Jag använde mig av utvecklingsverktyget Microsoft Visual Studio 2012 som jag tidigare är bekant med och jag anser det vara en mycket trevlig miljö att jobba i.

### 2.2 Utveckling

Eftersom att skriva ett eget bibliotek med grafer skulle ta alldeles för lång tid gick dom första veckorna åt att leta fram ett färdigt bibliotek och det visade sig finnas många. Eftersom jag hade fria händer över att välja vilket bibliotek som skulle användas så kunde jag välja det som jag tyckte verkade bäst. Jag fastnade för två stycken och det var Highcharts JS och Google Charts. Jag gick igenom guider om dessa och kollade på många olika exempel. Till sist valde jag att börja arbeta med Google Charts för att jag tyckte det verkade bäst gällande dokumentation. Jag kodade lite testsidor för att lära mig hur Google Charts fungerar, jag gjorde en testsida om hur det skulle tänkas se ut med förbrukning på vatten och el. I detta skede väntade jag på att få tillgång till marknadssystemet och även datat på kundernas förbrukning.

När jag sedan fick tillgång till marknadssystemet kan man säga att jag kunde börja jobba på riktigt. Jag hade nu även ett JSON-objekt att jobba med som där i fanns all nödvändig data för graferna, alltså beteckningarna som avser vilken förbrukning det gäller, förbrukningsmängd, belopp på förbrukningar, vilken period förbrukningen gäller för med mera.

Per Lindberg som är utvecklingschef sitter på kontoret i Falun och med honom satt jag i telefonmöten för att komma fram till vad som ska göras. Jag antecknade vad han sa och vad som skulle finnas på förbrukningssidan och jag hade då en klar bild hur det skulle se ut och vara. På sidan skulle den finnas flikar för att välja mellan olika förbrukningar, som varmvatten, kallvatten och el. Graferna skulle sedan vara i form av ett kolumndiagram, där varje månad (Januari-December) skrivs ut på X-axeln och varje årtal och dess förbrukningsmånad blir en ny stapel.. Allt måste ske dynamiskt och varje flik ska bara läggas till beroende på vad kunden har för data, t.ex. om den bara har el så ska ingen flik för vatten skapas. Ett önskemål från Per var också att med hjälp av radioknappar kunna filtrera om värdena i graferna skulle visa på kronor eller den enhet gällande förbrukning, t.ex. kWh när det gäller el.

Jobbet var nu att genom JSON-objektet sätta ut värdena på rätt plats där dom ska vara i Google Charts. Google Charts använder sig av kolumnindex och radindex för placering av data i deras så kallade DataTable-klass som sedan visar värdena i grafen, det var en del jobb med att få till en loop som går igenom JSON-objektet och placera ut rätt månad och värde på rätt kolumnindex och radindex. När jag hade gjort det började jag jobba med flikar och radioknapparna. Varje gång man ändrar flik eller radioknapp så ska så klart värden i grafen ändras. Till flikarna använde jag mig av jQuery Tabs som är ett plugin och det gör det väldigt enkelt att skapa flikar. Jag använde mig av Underscore.js som är ett bibliotek för JavaScript som kan användas till att få ut unika rader från JSON-objektet. De unika värden jag ville få ut var av typen String och heter beteckning, det var alltså beteckningen på förbrukningen, t.ex. Varmvatten. Värdena la jag in i en variabel som jag sedan körde en loop på som skapade en ny flik för varje värde.

Nu var det dags för radioknapparna. Efter att flikarna genereras upp behövde jag ta ut vilken flik som är aktiv, det vill säga den fliken som är markerad. Detta för att hämta ut den enhet som gäller den aktiva fliken, när fliken för vatten är aktiv ska bara enheten för vattenförbrukning finnas som en radioknapp och när fliken för el är aktiv ska bara enheten för elförbrukning finnas som en radioknapp. Då använde jag mig av jQuery.grep, grep är en jQuerymetod för att filtrera ut objekt ur en array till en ny array. Genom att skapa en ny array och köra jQuery.grep på JSON-objektet kunde jag lägga in rätt förbrukningar i den nya arrayen. jQuery.grep retunerar då i mitt fall bara dom förbrukningar som har samma beteckning som den aktiva fliken då villkoret är skrivet att grep ska retunera objekt med beteckningar i JSON-objektet som är lika med aktiva fliken. Alltså i villkoret är det två Strängar som jämförs. Nu hade jag en array med bara förbrukning samma som den aktiva fliken och kan då använda underscore.js för att få ut enheten unikt på förbrukningen och på samma sätt som jag gjorde med att skapa flikarna skapa radioknapparna. Värt att säga är att också att radioknappen för kronor är hårdkodad då alla förbrukningar använder sig av kronor.

Funktionen som körs när graferna ritas upp har två stycken parametrar, den ena är för vilken typ av förbrukning och den andra vilken värde som ska visas i staplarna, kronor eller förbrukningens enhet exempelvis kWh. Funktionen heter drawChart. När sedan drawChart funktionen kallas skickas den aktiva flikens och den aktiva radioknappens värden med som parametrar så att rätt värden kan visas i grafen. Nu var det dags att kunna ändra värdena i graferna på sidan genom att byta flik eller välja en annan radioknapp. Då skrev jag en "change" funktion för radioknapparna och en funktion för flikarna som gör att varje gång man ändrar på dom så kallar man på drawChart-funktionen och skickar med dom nya värdena och graferna ritas upp på nytt.

När arbetet började nå sitt mål och all generering av data skedde dynamiskt började jag titta på animationer på graferna, något för synens skull. Som jag berättade tidigare använde jag mig av Google charts och det visade sig vara svårare än jag trodde att få till några animationer. När jag pratar om animationer menar jag när graferna ritas, alltså att man kan se staplarna röra sig. Det finns även en funktion som jag ville ha med och det var att kunna filtrera på årtal genom bocka av dom årtalen man inte ville titta på, t.ex. om man bara vill jämföra dom två senaste åren. Jag började leta efter en lösning på detta men det visade sig att det skulle behövas mycket arbete för att göra det. Det var då jag började titta på Highcharts JS igen, och i Highcharts JS sker det snygga animationer som standard och även en filtreringsfunktion finns som standard. Eftersom det var en hel del jobb med att få till dessa animationer och filtrering på årtal i Google charts bestämde jag mig för att byta till Highcharts JS. Jag började läsa Highcharts JS dokumentation och det visade sig att inte vara något mer komplicerad än Google charts. Jag hade ju mina variabler, funktioner, loopar och if-satser så det jobbet som återstod var bara att sätta in data i Highcharts JS och det gick rätt enkelt.

Highcharts JS var till skillnad från Google Charts uppbyggt på ett sådant sätt att data lagras i så kallade Series som sedan placeras i Categories. Categories representerar då i mitt fall månaderna på x-axeln. Varje Serie har ett namn och en array som innehåller datat och där i placeras förbrukningsvärdena.

Valet att byta bibliotek var väldigt bra för att jobbet med att försöka få till dom funktioner som fanns som standard i Highcharts JS i Google Charts skulle ta upp mycket mera tid. Dessutom tyckte både jag och min handledare att Highcharts JS såg mycket bättre ut än Google Charts.

Jag kände mig nu klar med jobbet och bestämde ett telefonmöte med Per för att visa upp mitt arbete och jag fick positiv respons.

### **3 Slutprodukt**

Förbrukningssidan sker nu helt dynamiskt. Man kan navigera runt bland flikarna. I graferna kan man välja om man vill se värdena i kronor eller i enhet med radioknapparna. Man kan filtrera på årtal genom att bocka av dom åren man inte vill titta på. När man för musen över en staplarna visar ett tooltip dom exakta värdena för den gällande månaden. Det finns en funktion för exportering om man vill skriva ut grafen eller ladda ner den som en PDF eller en bild. Detta gör det nu enklare för en kund att analysera sin förbrukning.

Det arbete jag har gjort finns nu med i den senaste versionen av Momentum PM Marknad och kommer med tiden att levereras ut till kunder i och med uppdateringar av systemet.

### **4 Diskussion**

Under arbetets gång har jag stött på många problem och då lärt mig väldigt mycket. Tack vare min tidigare praktik på Momentum och erfarenhet av JavaScript underlättades arbetet. Många gånger har jag testat mig fram hur saker inom programmering fungerar vilket har bidragit till att jag har lärt mig nya saker.

Utifrån dom mål som sattes upp tycker jag att arbetet blev lyckat. Men det finns saker som skulle kunna förbättras och en sak jag tycker man skulle kunna ha är att i graferna bara visa t.ex. dom fem senaste åren som standard och sedan få valet att kunna välja fler år. Detta för att om en kund har data från långt bak i tiden så blir det många staplar, så att en standard på fem år tillbaka skulle se bättre ut när man först tittar på grafen.

Jag, utvecklingschefen och alla andra som tittat på mitt arbete är nöjda över vad jag har lyckats åstadkommit.

### **5 Resultat**

Till Momentum har jag bidragit med att deras ”Min sida” för kunder har blivit bättre när det gäller förbrukningsdelen. Jag har uppnått deras önskemål. Det som fanns tidigare var endast en tabell över förbrukningar men som nu i min mening bara kommer användas som ett komplement till graferna då man mycket enklare får ett grepp över sin förbrukning när man tittar på dessa.

För mig själv har jag knutit mina kontakter ännu mer sedan min tidigare praktik på Momentum. Jag har visat att jag kan jobba på egen hand och jag har hunnit leverera en färdig produkt. Jag har blivit bättre och jag har fördjupat mig inom programmering och att leta rätt på lösningar till problem som uppstår under ett arbete.

## **5.1 Slutsats**

### **Google**

Min slutsats är att inom programmering finns det många redan färdiga lösningar på problem och man behöver inte återuppfinna hjulet igen. Mycket av informationen kring mina problem i mitt jobb kunde man enkelt söka sig fram till via Google.

Det finns massor av information att hämta och det växer för varje dag. Enligt mig kan man bli bra på att googla genom att lära sig att välja träffar som relevanta till den lösning du letar efter.

[www.stackoverflow.com](http://www.stackoverflow.com) är en mycket bra sida för den som programmerar.

### **Testa**

Testa sin kod och gå igenom den steg för steg är mycket viktigt så man får en inblick av vad som händer i koden, så att man ser det resultat man får är det man vill ha. I webbläsaren Google Chrome är det väldigt enkelt att köra debugger på det JavaScript du vill genom att skriva `:debugger` i koden där du vill börja debugga och sedan trycka F12 i webbläsaren innan koden körs. F12 gör att du får upp JavaScript-konsolen i Google Chrome.

### **Våga göra fel**

Att göra fel gör bara att du istället lär dig att göra rätt. Att testa sig fram och göra fel gör att man får en förståelse hur det fungerar när man går igenom felet och rättar till det.

### **Låt andra titta på arbetet**

Genom att visa upp arbetet till någon annan kan göra att de hittar något som du själv har missat och du kan även få förslag på vad som kan göras bättre. Det bidrar till ett bättre arbete när man får synpunkter på vad som är bra och dåligt.

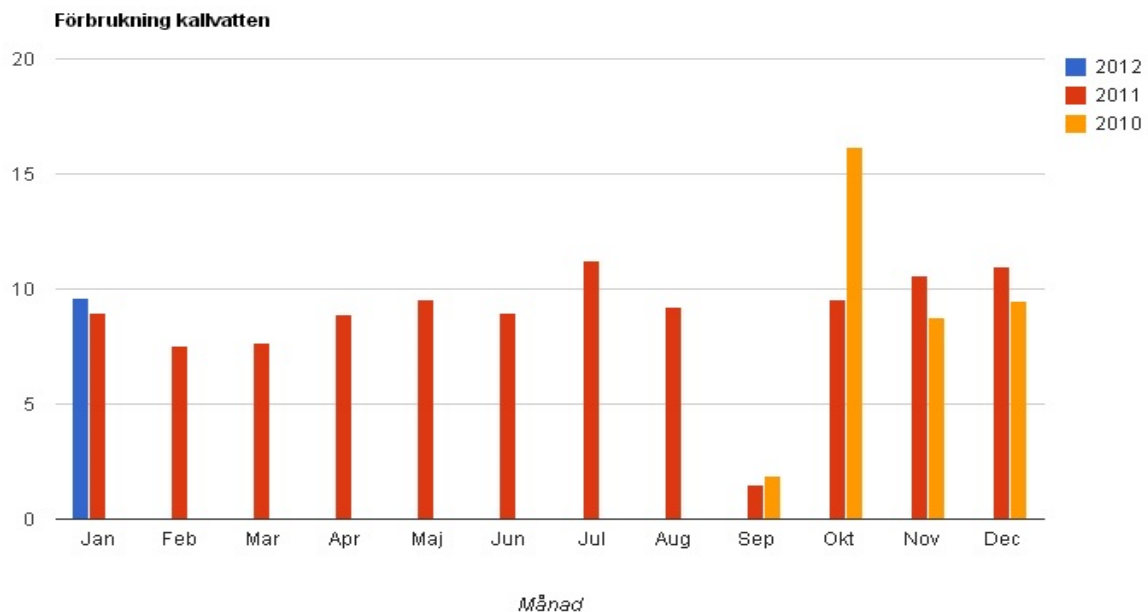


## 6 Referenser

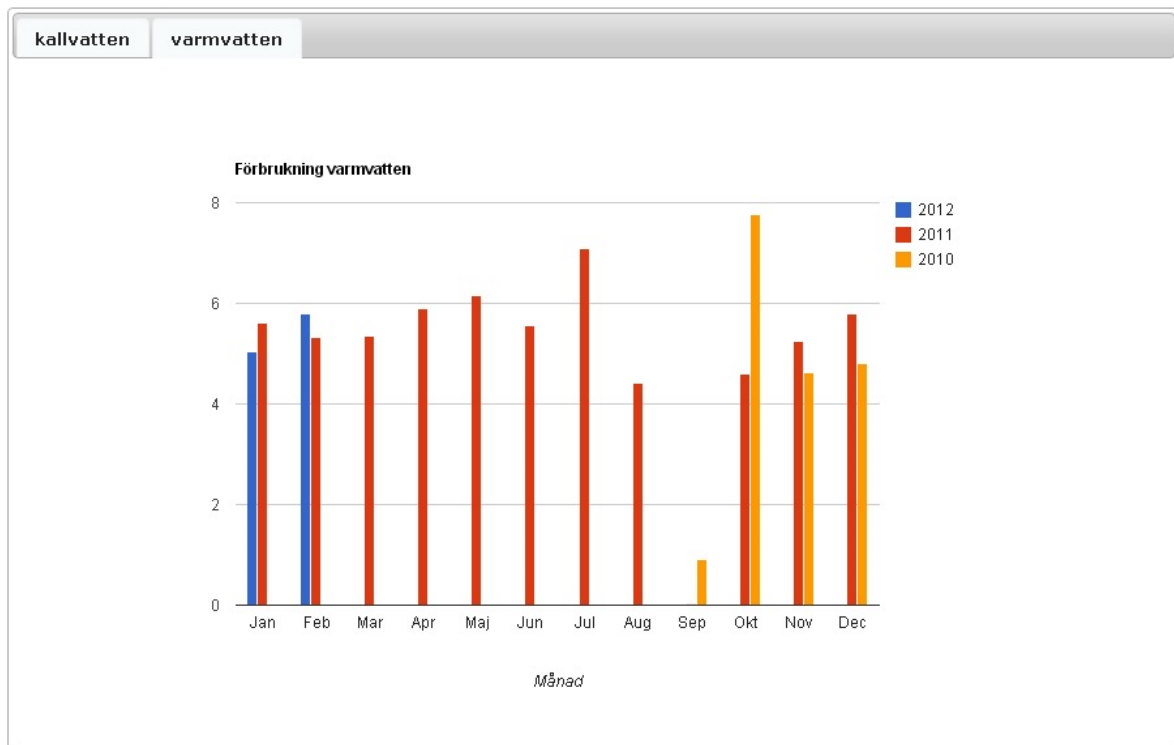
- [1] Highcharts JS dokumentation, [www.highcharts.com](http://www.highcharts.com) (Maj 2013)
- [2] Google Charts dokumentation, <https://google-developers.appspot.com/chart/> (Maj 2013)
- [3] Frågor och svar om programmering, [www.stackoverflow.com/](http://www.stackoverflow.com/) (Maj 2013)
- [4] Underscore.js, <http://underscorejs.org/> (Maj 2013)
- [5] JSON, <http://www.json.org/> (Maj 2013)
- [6] Ordförklaring till JSON, <http://sv.wikipedia.org/wiki/JSON> (Maj 2013)
- [7] Officiell hemsida jQuery, <http://jquery.com/> (Maj 2013)
- [8] SQL, <http://sv.wikipedia.org/wiki/SQL> (Maj 2013)
- [9] HTML, <http://sv.wikipedia.org/wiki/HTML> (Maj 2013)

## 7 Bilagor

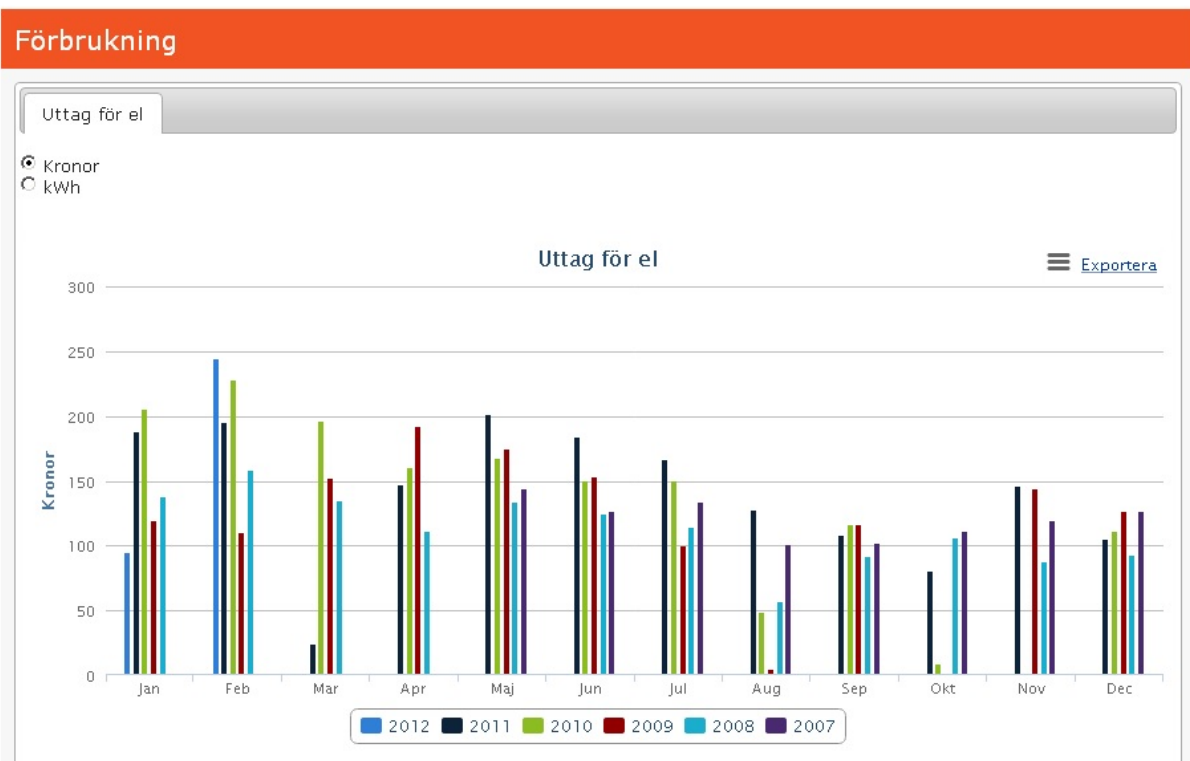
### 7.1 Bilder



En graf i tidigt skede skapad i Google Charts som visar förbrukning på kallvatten.



En graf skapad i Google Charts och flicker.



En graf skapad i Highcharts JS som visar elförbrukningen i kronor hos en kund.

```

> momentum.mat
[ ] Object , ▶ Object , ▶ Object , ▶ Object , ▶ Object , ▶ Object , ▶ Object , ▶ Object , ▶ Object , ▶ Object , ▶ Object
▶ Object , ▶ Object
  Beteckning: "Förbrukning kallvatten"      Beteckning: "Förbrukning varmvatten"
  KundNr: 144482                          KundNr: 144482
  Mätarställning: null                    Mätarställning: null
  MätarställningDatum: null              MätarställningDatum: null
  NettoBelopp: 26                         NettoBelopp: 107
  Period: "2011-03"                       Period: "2011-03"
  RadNr: 4                                 RadNr: 5
  Reskontrallr: 3042496                   Reskontrallr: 3042496
  TAAPris: 10                              TAAPris: 41
  TAAvserAvtalNr: null                     TAAvserAvtalNr: null
  TAAvserFom: "2011-01-01T00:00:00"       TAAvserFom: "2011-01-01T00:00:00"
  TAAvserProduktNr: null                  TAAvserProduktNr: null
  TAAvserTom: "2011-01-31T00:00:00"      TAAvserTom: "2011-01-31T00:00:00"
  TADebiterasMangd: 2.5780000000000003   TADebiterasMangd: 2.6029999999999998
  TAEinheit: "Kbm"                       TAEinheit: "Kbm"
  TAMangd: 7.66                           TAMangd: 5.34
  TAnormalMangd: 5.082                    TAnormalMangd: 2.737
  TASpecas: true                          TASpecas: true
  ▶ __proto__: Object                    ▶ __proto__: Object
▶ Object ]
> |

```

JSON-objektet som jag har arbetat med.

## 7.2 Kod

Kod där jag använder underscore.js för att få ut beteckningar och sedan lägga till dom som flikar.

```

var uniqueBeteckning = _(momentum.mat).pluck('Beteckning');
uniqueBeteckning = _(uniqueBeteckning).uniq();
$.each(uniqueBeteckning, function (i, item) {
  $("#tabs ul").append('<li><a href="#chart_div">' + item + '</a></li>');
});

```

Kod för change-funktionen på radioknapparna.

```

$('#input[name="rbList"]').live("change",function (e) {
  e.preventDefault();
  var selectedRb = $(this).val();
  selectedTabTitle = $("#tabs li.ui-tabs-active").text();
  drawChart(selectedRb, selectedTabTitle);
});

```

Kod för funktionen som körs när växling mellan flikarna görs och där jag bland annat använder jQuery.grep

```

$('#tabs').tabs({
  activate: function (e, ui) {
    selectedTabArray = jQuery.grep(momentum.mat, function (n, i) {
      return (n.Beteckning == $(ui.newTab).text());
    });
    var uniqueEnhet = _(selectedTabArray).pluck('TAEinheit');
    uniqueEnhet = _(uniqueEnhet).uniq();

    $("#radiobuttonlist").html("");
    $("#radiobuttonlist").append('<input type="radio" name="rbList"
value="Kronor" checked="checked" /> Kronor <br>');

    $.each(uniqueEnhet, function (i, item) {
      $("#radiobuttonlist").append('<input type="radio" name="rbList"
value=' + item + ' /> ' + item);
    });
  }
});

```

```
    drawChart($('input[name="rbList"]:checked').val(), $(ui.newTab).text());
  }
});
```

Kod från drawChart-funktionen där värdena till grafen läggs till.

```
if (addedYears.indexOf(year) > -1) {
  for (var i in dataSeries) {
    var currentSerie = dataSeries[i];
    if (currentSerie.name == year) {
      currentSerie.data[rowIndex] = unitType;
    }
  }
}
else {
  addedYears.push(year);
  var jsonObj = { name: year, data: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  jsonObj.data[rowIndex] = unitType;
  dataSeries.push(jsonObj)
}
}
```