Skapa din egen webbportfolio

Tina Durmén Blunt

tina.blunt@gaia.se

tinablunt.com

gaia effektfabriken

Vi kombinerar smart IT med kreativ digital kommunikation. Lite enklare. Lite skönare.


effekt.
by gaia

Visual studio gratis ☺

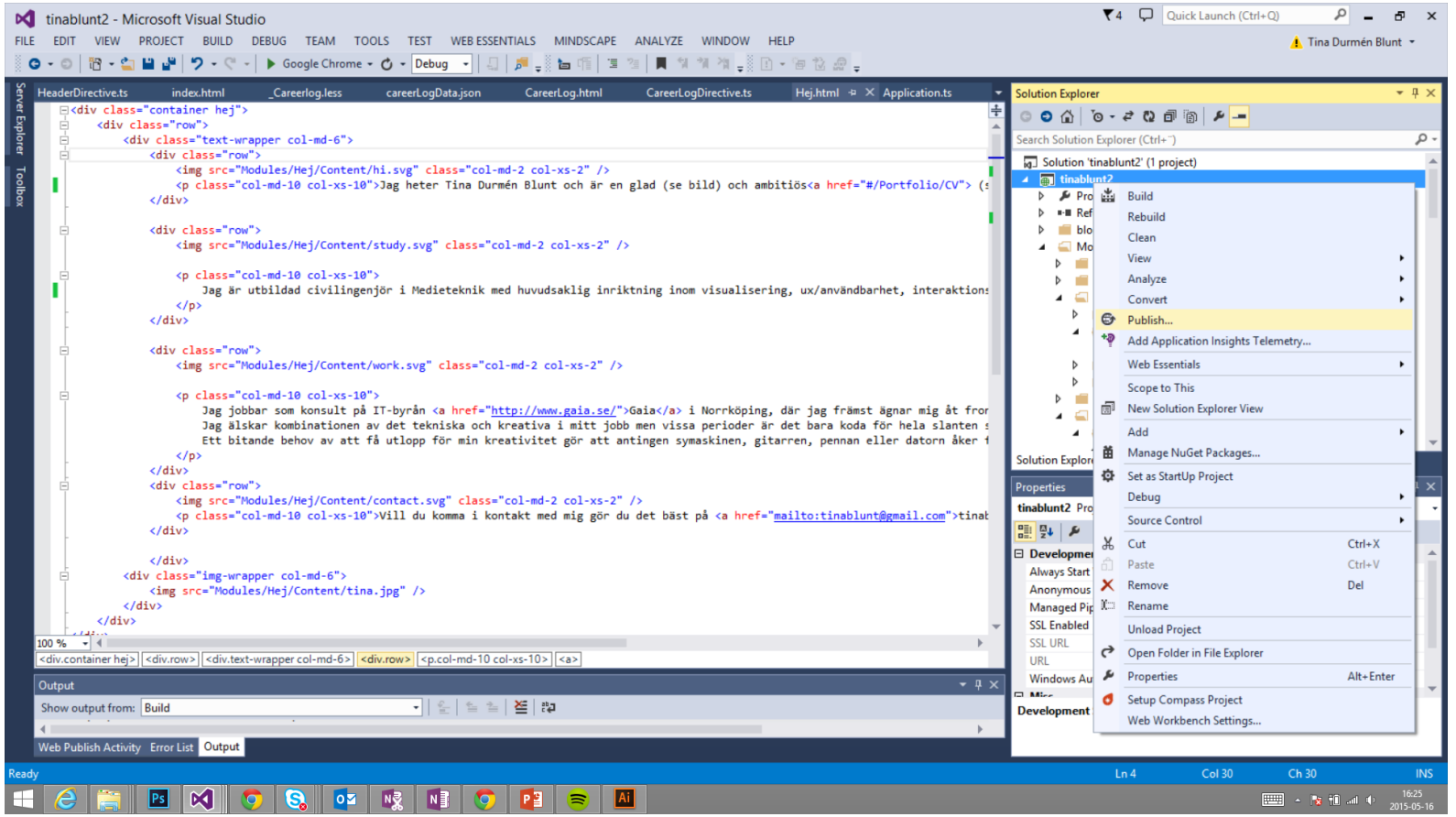
https://www.visualstudio.com/en-us/products/free-developer-offers-vs.aspx

Github

http://michaelcrump.net/setting-up-github-to-work-with-visual-studio-2013-step-by-step/

https://github.com/

Jag har min egen sida på webbhotellet one.com. Där kan man ställa in ftp-lösenord osv som man sen använder när man publicerar från Visual Studio. Man kan även köpa domäner på one.com. Alternativ till webbhotell är Azure eller Github som har ett visst antal gratis-sidor man kan sätta upp.



Kodskellettet

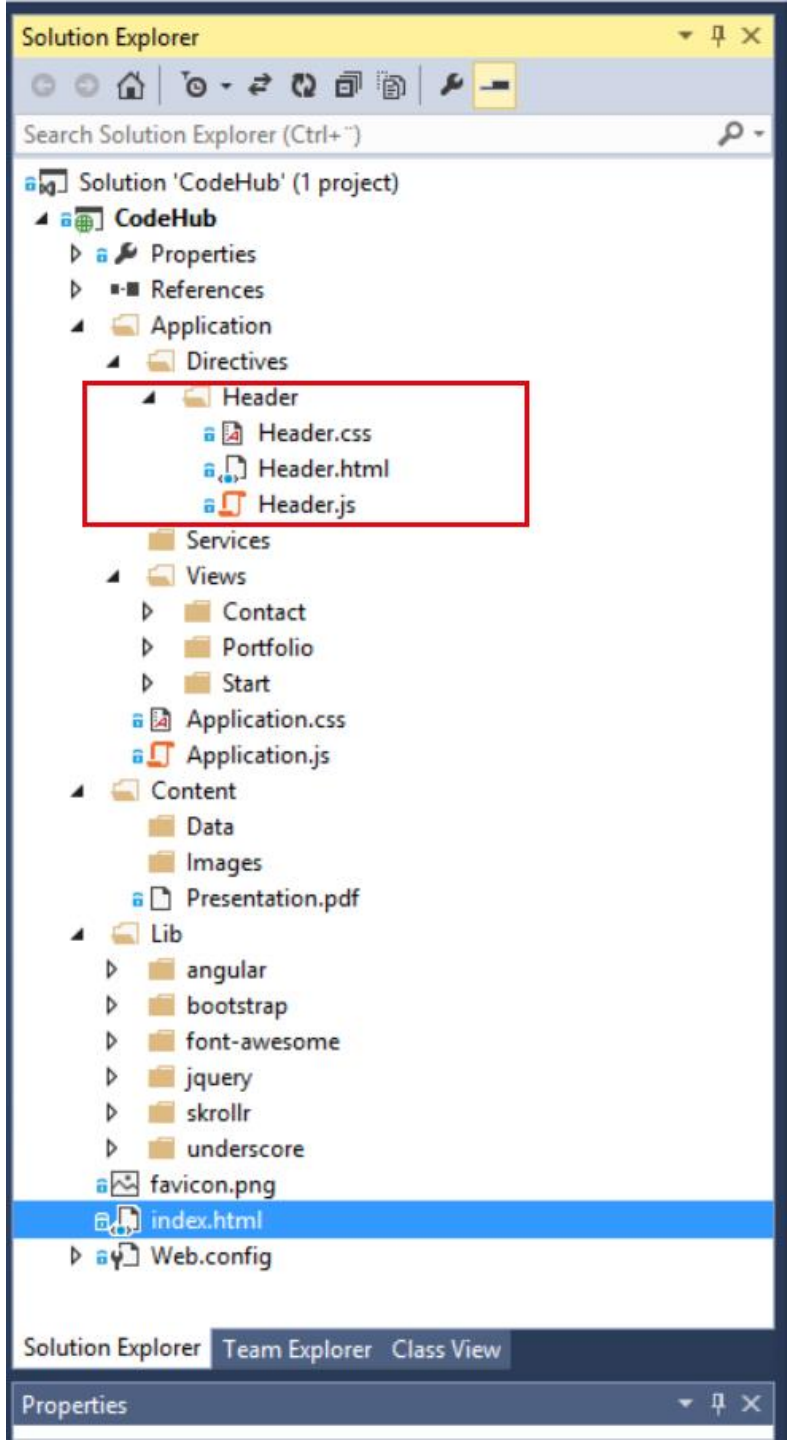
https://github.com/tblunt/CodeHub

Kodstruktur

Angular

Skrollr

Lite bootstrap (endast grid)



Struktur

Vad ska man tänka på?

Det ska vara lätt att hitta i och förstå koden.

Delar ska vara så utseparerade det går för att man lätt ska kunna flytta, ta bort och återanvända dem.

Dynamisk webb med Angular

https://angularjs.org/

Angular

Controllers

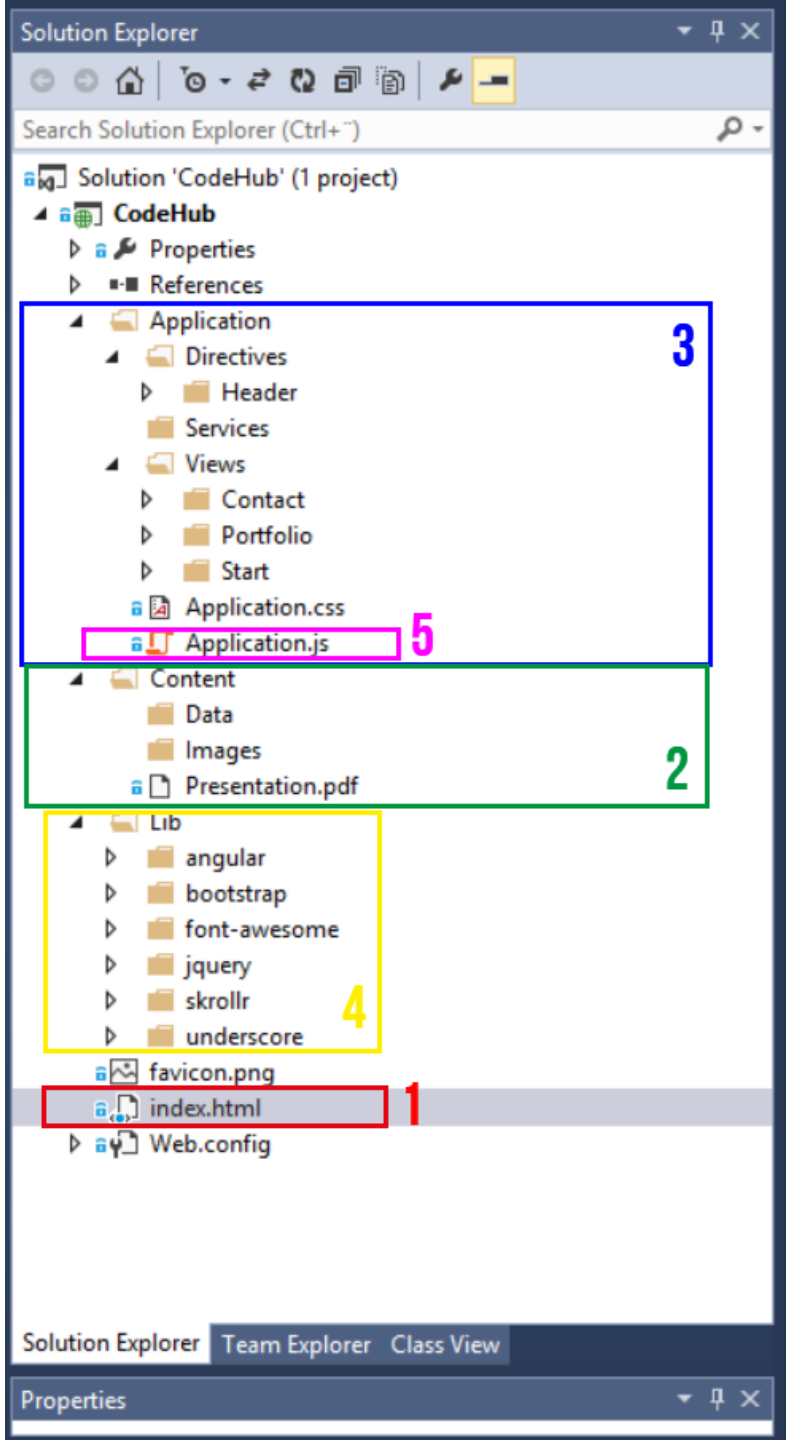
Kontrollerar allt som har med sin vy att göra.

Directives

Kontrollerar sin del av vyn. Egna html-element. Bra för att kunna återanvända och separera kod.

Services

Har hand om att hämta och bearbeta data.



1. index.html

Länkar till alla script och css. Innehåller topnavigeringen och div med ng-view som angular stoppar in innehåll i beroende på url:en.

2. Content

Bilder, json-data och annat.

3. Application

Angular-modulen som bygger upp hela sidan. I vissa fall har man fler moduler med direktiv, controllers och services.

4. Lib

Alla externa skript.

5. Application.js

Modulen initieras och vyerna knyts till url:en. Så fort man ska lägga till en ny vy så måste man lägga till den här.

```html
<h1>List på några kurser</h1>
<div ng-repeat="item in courseData">
    <h3>{{item.title}}</h3>
    <p>{{item.description}}</p>
</div>
```

ng-repeat (courseData.html)

Loopa över data-array som sitter på scopet (courseData.js)

```html
<div class="portfolio-menu">
    <div class="menu-item" ng-click="goToView(0)" id="menu-nummer1">
        <p>Nummer 1</p>
    </div>
    <div class="menu-item" ng-click="goToView(1)" id="menu-nummer2">
        <p>Nummer 2</p>
    </div>
    <div class="menu-item" ng-click="goToView(2)" id="menu-nummer3">
        <p>Nummer 3</p>
    </div>
    <div class="menu-item" ng-click="goToView(3)" id="menu-nummer4">
        <p>Nummer 4</p>
    </div>
</div>
```

ng-click (Portfolio.html)

När användaren klickar körs scope-funktionen. OBS! Måste vara scope-funktion, annars kommer man inte åt den!!

```html
<div class="nav">
    <div class="nav-item" ng-class="{'selected': selectedView == '/'}">
        <a href="#/">
            Start
            <span class="fa fa-angle-double-down"></span>
        </a>
    </div>
    <div class="nav-item" ng-class="{'selected': selectedView == '/Portfolio'}">
        <a href="#/Portfolio">
            Portfolio
            <span class="fa fa-angle-double-down"></span>
        </a>
    </div>
    <div class="nav-item" ng-class="{'selected': selectedView == '/Kontakt'}">
        <a href="#/Kontakt">
            Kontakt
            <span class="fa fa-angle-double-down"></span>
        </a>
    </div>
</div>
```

ng-class (header.html)

Sätter en klass beroende på villkoret.

```
(function () {
    'use strict';

    //hämtar Application-modulen som redan är skapad i Application.js
    var application = angular.module('Application');

    //Binder direktivet till vår modul
    application.directive('header', ['$location', '$route', function ($location, $route) {

        function link($scope, $element, $attrs) {

            //vyn man befinner sig på ('/', '/Portfiolio' eller '/Kontakt')
            $scope.selectedView = $location.path();

            //triggas då användaren klickar i headern
            $scope.$on('$routeChangeStart', function() {
                $scope.selectedView = $location.path();
            });

        }

        return {
            templateUrl: "Application/Directives/Header/Header.html",
            restrict: 'E',
            link: link
        };
    }])
})();
```

Direktiv

`Application.directive('header' ....`

'header' blir namnet på direktivet <header>

Kamelbokstäver översätts i html till bindestreck t.ex:

'courseList' bilr <course-list>

templateUrl: binder html till scriptet

Restrict: vilken typ av dirktiv det ska vara. I det här fallet gör vi ett helt eget element <header> men man kan även lägga till direktivet som ett attribut: <div *header*>

Link: Funktionene som körs när direktivet ritas ut.

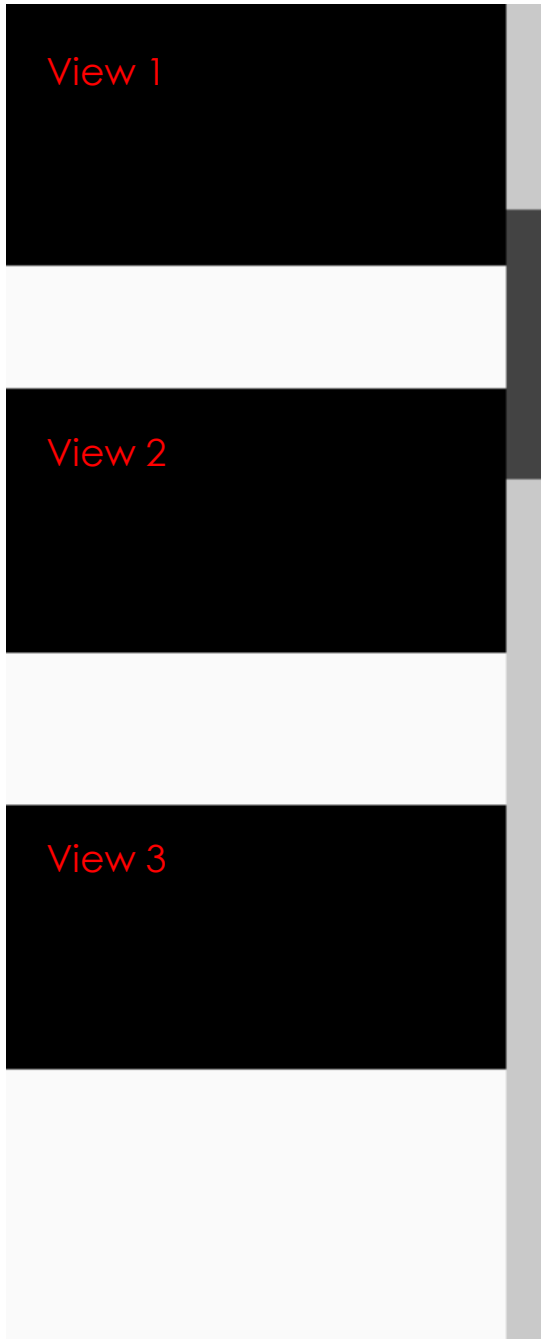
Parallax scrolling med skrollr.js

http://prinzhorn.github.io/skrollr/

Skrollr

Binder css-animationer till olika tidpunkter med hjälp av data-attribut på html-elementen. Ex:

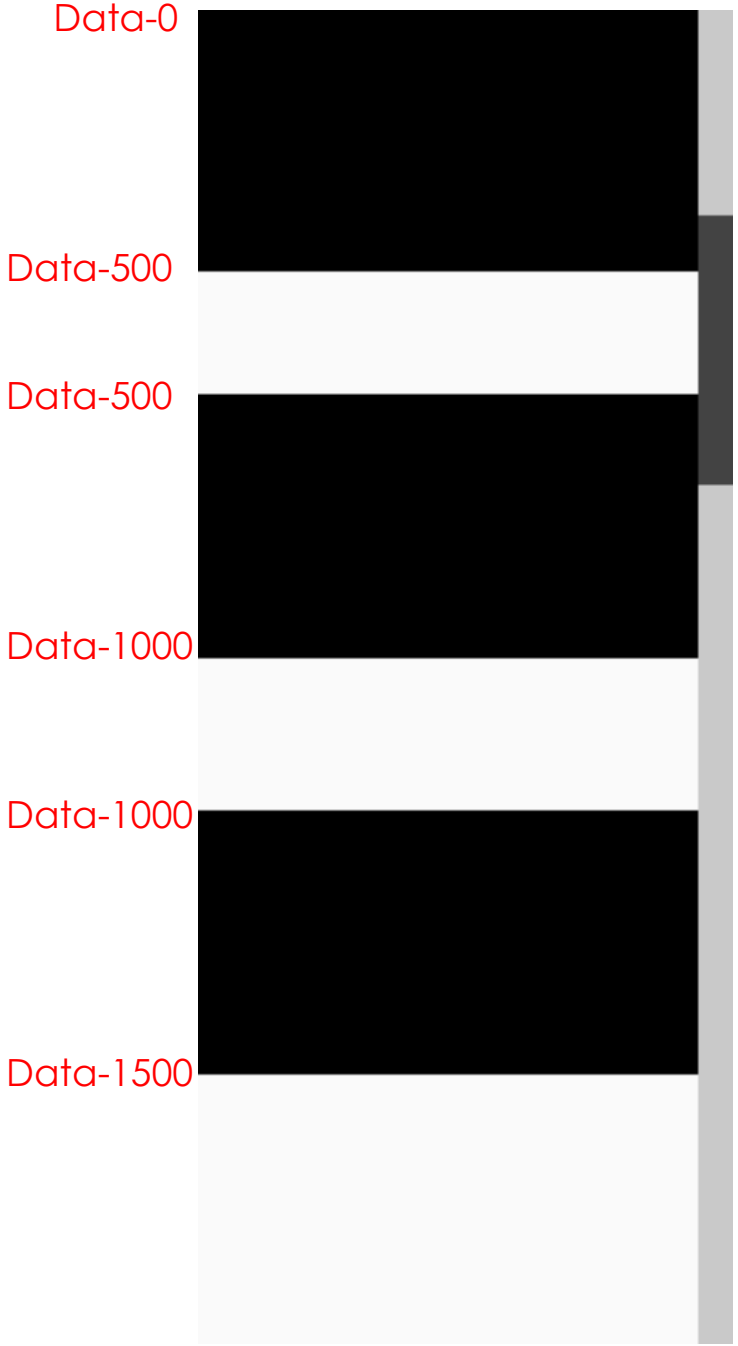
Detta betyder att när scrollen kommit ner 200px så börjar animationen. Så mellan 200 och 400 så kommer opaciteten gå från 0 till 1.



PortfolioController.js

Eftersom vi vill ha en responsiv sida vet vi ju inte exakt när våra animationer ska sätta igång. På mobil kommer kanske våra sektioner vara mycket längre, då kan vi inte ha satt statiska tider tex ”data-500”.

Därför använder vi script för att räkna ut när animationerna ska börja och sluta för varje vy. I scriptet sätter vi på attributen på divarna som ska animeras tex ”data-” *+ höjden för vyn*



I PortfolioController.js lägger skriptet på skrollrs data-attribut på alla divs som har classen 'view' (svarta rutorna).

getKeyframes()

Hämtar alla views. Varje vy få en keyframe-tid. Alltså när det är den vyns tur att börja animera. Keyframen räknas ut med alla vyernas höjder som ligger innan. Från bilden till vänster kommer keyframe se ut såhär:
[0,500,1000,1500]

animateViews()

Sätter på data-attributen på alla view-div med hjälp av keyframes-arrayen.
Animerar även menyknapparna till vänster med samma keyframe-värden.

```
_.each(views, function(view, index)  {

    if ($(view).height() < $(window).height()) {
        viewHeight = $(window).height();
    }
    else {
        viewHeight = $(view).height();
    }

    //egna höjden + föregående vy start-tid + hur länge vyn ska ligga stilla i början
    var nextViewKeyFrame = viewHeight + keyframes[index] + pauseAtBeginning;

    keyframes.push(nextViewKeyFrame);

});

$("body").height(keyframes[(keyframes.length - 1)]);
```

För att våra vyer ska ”snappa” till i början så lägger vi på lite tid (i det här fallet 500px) som vyn får på sig innan man börjar scrolla i den.

Det är därför det känns som att den fastnar i fönstrets överkant när den har flygit in.

I Portfolio.html kan du sätta på egna animationer eller göra animationen längre på view-divsen. Sätter du inte attibuten kommer vyn flyga vertikalt bara (se animateViews() där default-värdena sätts).


```
<div class="view" id="nummer2"
     before-enter="opacity:0;"
     after-enter="opacity:1;"
     before-leave="opacity:1;"
     after-leave="opacity:0;"
     enter-duration="500"
     leave-duration="500">
    <div class="container">
        <course-list></course-list>
    </div>
</div>
```