weiter mit 55 01 08.Deeper Into AngularJS (intro)

## Zen coding Examples:

@\*h1.heading\*@

<h1 class="heading"></h1>

@\*h1#heading\*@

<h1 id="heading"></h1>

@\*div>h1\*@

<div>

<h1></h1>

</div>

creating sibling structure with +

@\*div.message-header>h1+p.lead\*@

<div class="message-header">

<h1>Message Board</h1>

<p class="lead">Latest messages</p>

</div>

creating nested structures with >

@\*div>ul>li\*5\*@

<div>

<ul>

<li></li>

<li></li>

<li></li>

<li></li>

<li></li>

</ul>

</div>

zen-coding grouping with () and >{}

@\*.messages>.message>(.title>{The title})+.date+.contents+.toolbar+.replies\*@

<div class="messages">

<div class="message">

<div class="title">The title</div>

<div class="date"></div>

<div class="contents"></div>

<div class="toolbar"></div>

<div class="replies"></div>

</div>

</div>

@\*.messages>.message>(.title>lorem)+.date+.contents+.toolbar+.replies\*@

<div class="messages">

<div class="message">

<div class="title">Tincidunt integer eu augue augue nunc elit dolor, luctus placerat scelerisque euismod, iaculis eu lacus nunc mi elit, vehicula ut laoreet ac, aliquam sit amet justo nunc tempor, metus vel.</div>

<div class="date"></div>

<div class="contents"></div>

<div class="toolbar"></div>

<div class="replies"></div>

</div>

</div>

## Database Migrations:

Watch out for declaring explicitly the project name if DAL and ASP.NET MVC projects lives in separated projects:

DAL-ProjectName = “Persistance”

.NET MVC-ProjectName = “MessageBoard”

Commands:

* *Enable-Migrations -ProjectName "*Persistance” *-StartUpProjectName "*MessageBoard*"*
* *Add-Migration Initial -ProjectName "Persistance (evtl. mit -IgnoreChanges -Force)"*
* *Update-Database -ProjectName "Persistance" -verbose*

<http://professorweb.ru/my/entity-framework/6/level2/2_11.php>

<http://www.asp.net/web-api/overview/data/using-web-api-with-entity-framework/part-3>

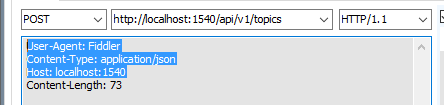
<http://www.mortenanderson.net/code-first-migrations-for-entity-framework>

after every change in the model push “add-migration” to database

update to reflect changes in SQL

**API**

POST: Make sure having the request-header right when posting to the web-api via Fiddler. Content-Type must be spelled correctly.



**AngularJS**

### Filters

The idea behind filters is being able to intercept the call and do smth with the data before data-binding happens.

**Routes**

* As of Angular 1.5, added angular-route.min.js to \_Layout.cshtml and injected ngRoute into angular.module() in order to make interpagerouting happen.
* Edited $Routeprovider for ControllerAs systax