

Quick Getting Started

1. npm install -g gulp
2. npm install -g jspm
3. jspm registry config github
4. Download skeleton-navigation
<https://github.com/aurelia/skeleton-navigation/releases>
5. Change directories to skeleton directory
6. npm install
7. jspm install -y
8. gulp watch

Make sure to have node installed first

Templating Examples

.one-way	<code><markdown-editor value.one-way="markdown"></markdown-editor></code>
.two-way	<code><markdown-editor value.two-way="markdown"></markdown-editor></code>
.bind	<code><input type="text" value.bind="firstName"> <a href.bind="url">Aurelia</code>
trigger	<code><button click.trigger="sayHello()">Say Hello</button></code>
delegate	<code><button click.delegate="sayHello()">Say Hello</button></code>
call	<code><div touch.call="sayHello()">Say Hello</div></code>
string interpolation	<code><div class="dot \${color}" \${isHappy ? 'green' : 'red'}"></div></code>
ref	<code><input type="text" ref="name"> \${name.value}</code>

Templating Examples (cont)

select	<code><option repeat.for="color of colors" value.bind="color">\${color}</option></code>
innerhtml	<code><div innerhtml.bind="htmlProperty"></div> <div innerhtml="\${htmlProperty}"></div></code>
textContent	<code><div textcontent.bind="stringProperty"></div> <div textcontent="\${stringProperty}"></div></code>
style	<code><div style.bind="styleString"></div> <div style.bind="styleObject"></div></code>

Template Example Code

```
<template>
  <section>
    <h2>${heading}</h2>
    <form role="form" submit.delegate="welcome()">
      <div class="form-group">
        <label for="fn">First Name</label>
        <input type="text" value.bind="firstName" class="form-control" id="fn" placeholder="first name">
      </div>
      <div class="form-group">
        <label for="ln">Last Name</label>
        <input type="text" value.bind="lastName" class="form-control" id="ln" placeholder="last name">
      </div>
    </form>
  </section>
</template>
```

Template Example Code (cont)

```
<div class="form-group">
  <label>Full Name</label>
  <p class="help-block">${fullName | upper}</p>
</div>
<button type="submit" class="btn btn-default">Submit</button>
</form>
</section>
</template>
```

All templates must start with `<template>` and end with `</template>`

HTTP Client Methods

Install	jspm install aurelia-http-client
createRequest(uri)	Custom configure individual requests
delete(uri)	HTTP Delete request
get(uri)	HTTP Get Request
head(uri)	HTTP Head Request



By **Erik Hanchett** (ErikCH)
cheatography.com/erikch/
www.ProgramWithErik.com

Published 29th March, 2015.
Last updated 29th March, 2015.
Page 1 of 3.

Sponsored by **Readability-Score.com**
Measure your website readability!
<https://readability-score.com>

HTTP Client Methods (cont)

jsonp(uri, callbackParameterName='JSONPRequestMessage') Sends JSONPRequestMessage

put(uri, content) HTTP Put Request

patch(uri, content) HTTP Patch Request

post(uri, content) HTTP Post Request

The result of sending a message is a promise, it has the following properties. response, responseType, content, headers, statusCode, statusText, isSuccess, reviver, requestMessage

HTTP Client Example (Flickr)

```
import {HttpClient} from 'aurelia-http-client';
var url = 'http://api.flickr.com/services/feeds/photos_public.gne?tags=rainier&tagmode=any&format=json';
export class Flickr{
  static inject() { return [HttpClient]; }
  constructor(http) {
    this.heading = 'Flickr';
```

HTTP Client Example (Flickr) (cont)

```
    this.images = [];
    this.http = http;
  }
  activate() {
    return
    this.http.jsonp(url).then(response => {
      this.images = response.content.items;
    });
  }
  canDeactivate() {
    return confirm('Are you sure you want to leave?');
  }
}
```

jsonp request returns a promise

Extending HTML Behaviours

Attached Behaviours Adds new behavior or functionality to existing HTML element

Custom Elements New tags to HTML Markup

Extending HTML Behaviours (cont)

Template Convert Dom into inert HTML
Controllers template

Behavior Examples

```
//Attached Behavior (jquery)
<div show.bind="isSaving" class="spinner"></div>
import {Behavior} from 'aurelia-templating';
export class Show {
  static metadata() {
    return Behavior
      .attachedBehavior('show')
      .withProperty('value', 'valueChanged', 'show');
  }
}
//Custom Element
<template>
  <require from="./say-hello">
</require>
  <input type="text" ref="name">
  <say-hello to.bind="name.value"></say-hello>
</template>
import {Behavior} from 'aurelia-templating';
```



By **Erik Hanchett** (ErikCH)
cheatography.com/erikch/
www.ProgramWithErik.com

Published 29th March, 2015.
Last updated 29th March, 2015.
Page 2 of 3.

Sponsored by **Readability-Score.com**
Measure your website readability!
<https://readability-score.com>

Behavior Examples (cont)

```
export class SayHello {
  static metadata() {
    return Behavior
      .customElement('say-hello')
      .withProperty('to');
  }
  speak() {
    alert('Hello ${this.to!}');
  }
}
//Template Controllers
<li repeat.for="customer of
customers">${customer.fullName}</li
>
```

Routing Activation Lifecycle

canActivate(params, queryString, routeConfig)	Hook to be navigated too
--------------------------------------------------	-----------------------------

Routing Activation Lifecycle (cont)

activate(params, queryString, routeConfig)	Custom logic before view-model is displayed
canDeactivate()	Router can navigate from view?
deactivate()	Logic as you are being navigated from

Router Example

```
import {Router} from 'aurelia-
router';
import bootstrap from 'bootstrap';
export class App {
  static inject() { return
[Router]; }
  constructor(router) {
    this.router = router;
    this.router.configure(config =>
{
    config.title = 'Aurelia';
    config.map([
      { route: ['', 'welcome'],
moduleId: './welcome', nav: true,
title: 'Welcome' },
```

Router Example (cont)

```
      { route: 'flickr',
moduleId: './flickr', nav: true },
      { route: 'child-router',
moduleId: './child-router', nav:
true, title: 'Child Router' }
    ]);
  });
}
~
```

DI Example

```
import {HttpClient} from 'aurelia-
http-client';
export class CustomerDetail{
  static inject() { return
[HttpClient]; }
  constructor(http) {
    this.http = http;
  }
}
```



By **Erik Hanchett** (ErikCH)
cheatography.com/erikch/
www.ProgramWithErik.com

Published 29th March, 2015.
 Last updated 29th March, 2015.
 Page 3 of 3.

Sponsored by **Readability-Score.com**
 Measure your website readability!
<https://readability-score.com>