

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><input type="text" value.bind="firstName" /></code> |
| two-way | <code><input type="text" value.two-way.bind="firstName" /></code> |
| .bind | <code><input type="text" value.bind="firstName" /></code> |
| trigger | <code><button click.trigger="sayHello()" /></code> |
| delegate | <code><button click.delegate="sayHello()" /></code> |
| call | <code><div touch.call="sayHello()" /></code> |
| string interpolation | <code><div class="dot \${color}" /></code> |
| ref | <code><input type="text" ref="name" /></code> |
| select | <code><option repeat.for="color of colors" value.bind="color" /></code> |
| inner html | <code><div innerhtml.bind="html" /></code> |
| textContent | <code><div textcontent.bind="stringProperty" /></code> |
| style | <code><div style.bind="styleString" /></code> |

Template Example Code

```
<template>
  <section>
    <h2>${heading}</h2>
    <form role="form"
      submit.delegate="welcome()">
```

Template Example Code (cont)

```
> <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>
<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 | <code>jspm install aurelia-http-client</code> |
| createRequest(uri) | Custom configure individual requests |
| delete(uri) | HTTP Delete request |
| get(uri) | HTTP Get Request |
| head(uri) | HTTP Head Request |

HTTP Client Methods (cont)

| | |
|---|-----------------------------|
| <code>jsonp(uri, callbackParameterName='jsoncallback')</code> | Sends JSONP Request Message |
| <code>put(uri, content)</code> | HTTP Put Request |
| <code>patch(uri, content)</code> | HTTP Patch Request |
| <code>post(uri, content)</code> | 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&format=json';
export class Flickr {
  static inject() { return [HttpClient]; }
  constructor(http) {
    this.heading = 'Flickr';
    this.htmlProperty = '';
    this.images = [];
    this.http = http;
  }
  activate() {
    return this.fetch();
  }
  fetch() {
    return this.http.get(url).then(response => {
      this.images = response.content.items;
    });
  }
  deactivate() {
    return confirm('Are you sure you want to leave?');
  }
}
```



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

Published 29th March, 2015.
 Last updated 11th May, 2016.
 Page 1 of 2.

Sponsored by CrosswordCheats.com
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

HTTP Client Example (Flickr) (cont)

```
> }
}
```

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

Template Controllers Convert Dom into inert HTML template

Behavior Examples

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

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 |
| 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-trap';
export class App {
  static inject() { return [Router]; }
  constructor(router) {
    this.router = router;
    this.router.config = {
      routes: (config => {
        config.title = 'Aurelia';
        config.map([
          { route: ['', 'welcome'], moduleId: './welcome', nav: true, title: 'Welcome' },
          { 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 CustomDetail {
  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 11th May, 2016.
 Page 2 of 2.

Sponsored by CrosswordCheats.com
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>