



# Intro to AngularJS

More than just data binding



---

WE ARE HIRING!



LogicalAdvantage.com | January 2015  
f /LogicalAdvantage t @LogicalAdvantage



## Agenda

- Overview of AngularJS
- Focus more on structure and use than features
- Highlight more complex areas
- Provide a starting point for you to begin analyzing for yourself
- Review some code



## What is it?

- Angular allows you to extend HTML vocabulary to provide dynamic behavior.
- Provides an MVC pattern for client side (browser) application development.
- Developed by a team at Google
- <http://angularjs.org>



## Why are we talking about it?



From: <http://larseidnes.com/2014/11/05/angularjs-the-bad-parts/>



# Browser becomes another device





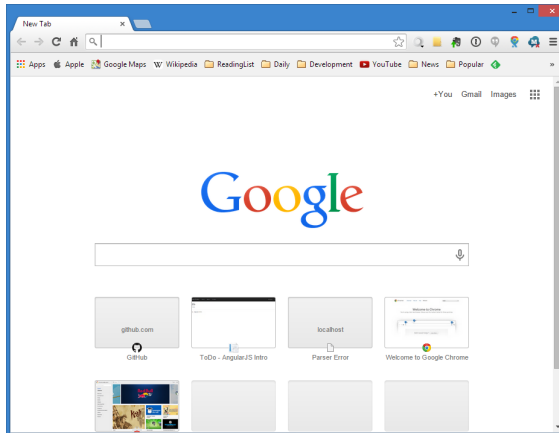
## Basic Angular Functionality

- <https://angularjs.org>

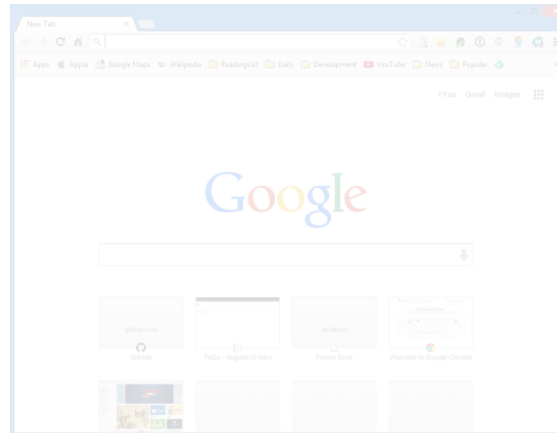


# How the Data Bind Magic works

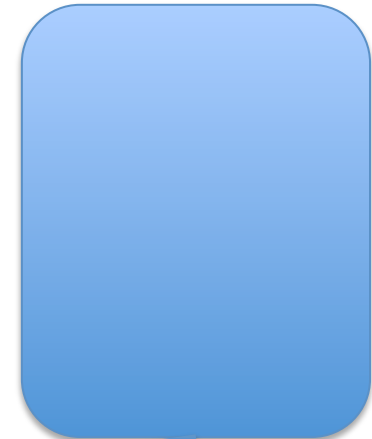
Browser DOM



Angular Shadow DOM



View Model



\$digest cycle



## Isn't MVC already on the server?

- Yes. Using Asp.Net MVC there is a server side MVC pattern.
- Using AngularJS there is a second MVC pattern in the browser.
- The two are independent of each other.







# Angular View

```
<div ng-app="todoApp">
  <div ng-controller="todoListController as tdl">
    <h2>ToDo</h2>
    <p>My ToDo's</p>
    <ul>
      <li ng-repeat="todo in tdl.todos" >
        <span>{{todo.title}}</span>-<span ng-class="{overdue: todo.overdue}">{{todo.formattedDueDate}}</span>
      </li>
    </ul>
  </div>
</div>
```

---

WE ARE HIRING!

9



LogicalAdvantage.com | January 2015  
f /LogicalAdvantage @LogicalAdvantage

# Angular Controller

```
angular
  .module('todoApp')
  .controller('todoListController', todoListController);

todoListController.$inject = ['todoDataService'];

function todoListController(todoDataService)
{
  var vm = this;
  vm.todos = [];

  activate();

  function activate() {
    return getTodos().then(function () {
      //log info
    });
  }
}
```



# Angular Model

Vm viewmodel

```
function todoListController(todoDataService)
{
    var vm = this;
    vm.todos = [];
```



# Angular App – Main container

```
angular
  .module('todoApp', [
  ]);
```



# Angular Service

```
angular
  .module('todoApp')
  .factory('configService', configService);

configService.$inject = ['$http'];

function configService($http) {
  return {
    todoApiUrl: todoApiUrl
  };

  function todoApiUrl() {
    return 'http://localhost:19042/api/todos';
  }
}
```



## Directives – UI Components

- Directives provide a mechanism for componentizing the UI.



## Angular Modules provide Testability

- Using tools such as JasmineJS and Qunit each module (Controller, Service, Directive) can be unit tested independently.



## Validation

- Always validate on the server as well as the UI.
- <http://scotch.io/tutorials/javascript/angularjs-form-validation>





# Angular Coding Standards

- From Angular  
<https://docs.google.com/document/d/1XXMvReO8-Awi1EZXAXS4PzDzdNvV6pGcuaF4Q9821Es/pub>
- Opinionated Style from John Papa  
<https://github.com/johnpapa/angularjs-styleguide>





## References

- <http://angularjs.org>
- <http://larseidnes.com/2014/11/05/angularjs-the-bad-parts/>
- <http://scotch.io/tutorials/javascript/angularjs-form-validation>

