

Angular 2 Core



Igor Minar
Tobias Bosch

ng-europe 2014



To Do

1



2



3



4



5





Carrier 5:47 PM

Tobias

New todo +

☐

New Macbook Pro

×

☐

Tesla Model X

×

☐

Nerf N-Strike Bow Blaster

×

☐

Oculus Rift

×

☐

Quadcopter

×

☐

Watch earth from space

×



Tobias



Good kids



Bad kids



AngularJS 1.3

```
<div ng-controller="SantaTodoController">
  <input type="text" ng-model="newTodoTitle">
  <button ng-click="addTodo()">+</button>

  <tab-container>
    <tab-pane title="Tobias">
      <div ng-repeat="todo in todosOf('tobias')">
        <input type="checkbox"
              ng-model="todo.done">
          {{todo.title}}
        <button ng-click="deleteTodo(todo)">
          X
        </button>
      </div>
    </tab-pane>
    ...
  </div>
```



AngularJS 2.0

```
<div>
```

```
  <input type="text" [value]="newTodoTitle">
```

```
  <button (click)="addTodo()">+</button>
```

```
<tab-container>
```

```
  <tab-pane title="Good kids">
```

```
    <div [ng-repeat|todo]="todosOf('good')">
```

```
      <input type="checkbox"
```

```
        [checked]="todo.done">
```

```
        {{todo.title}}
```

```
      <button (click)="deleteTodo(todo)">
```

```
        X
```

```
      </button>
```

```
    </div>
```

```
  </tab-pane>
```

```
...
```



AngularJS 2.0

```
<div>
  <input type="text" [value]="newTodoTitle">
  <button (click)="addTodo()">+</button>

  <tab-container>
    <tab-pane title="Good kids">
      <div [ng-repeat|todo]="todosOf('good')">
        <input type="checkbox"
          [checked]="todo.done">
          {{todo.title}}
        <button (click)="deleteTodo(todo)">
          X
        </button>
      </div>
    </tab-pane>
  </div>
```

...



AngularJS 2.0

```
<div>
```

```
  <input type="text" [value]="newTodoTitle">
```

```
  <button (click)="addTodo()">+</button>
```

```
<tab-container>
```

```
  <tab-pane title="Good kids">
```

```
    <div [ng-repeat|todo]="todosOf('good')">
```

```
      <input type="checkbox"
```

```
        [checked]="todo.done">
```

```
        {{todo.title}}
```

```
      <button (click)="deleteTodo(todo)">
```

```
        X
```

```
      </button>
```

```
    </div>
```

```
  </tab-pane>
```

```
...
```



AngularJS 2.0

```
<div>
  <input type="text" [value]="newTodoTitle">
  <button (click)="addTodo()">+</button>

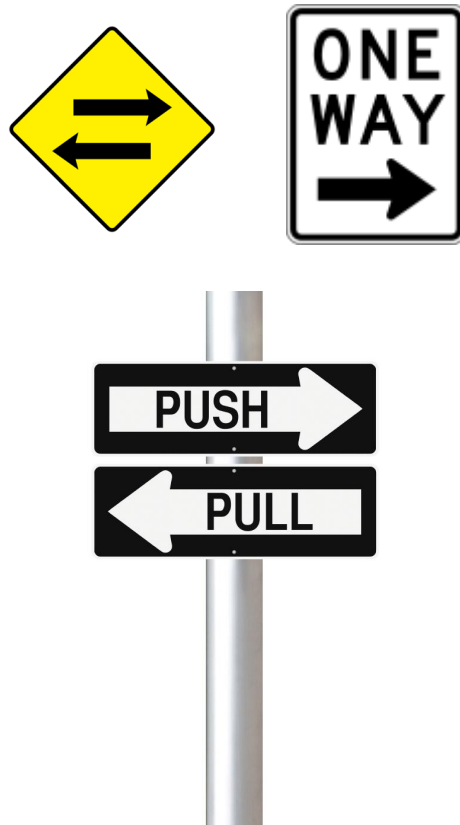
  <tab-container>
    <tab-pane title="Good kids">
      <div [ng-repeat|todo]="todosOf('good')">
        <input type="checkbox"
          [checked]="todo.done">
          {{todo.title}}
        <button (click)="deleteTodo(todo)">
          X
        </button>
      </div>
    </tab-pane>
```

...



Databinding

```
<elm attr="..."> elm.property=...
```







Directive Definition Object ("DDO")

```
myModule.directive('directiveName', function factory(injectables) {
  return {
    priority: 0,
    template: '<div></div>', // or // function(tElement, tAttrs) { ... },
    // or
    // templateUrl: 'directive.html', // or // function(tElement, tAttrs) { ... },
    transclude: false,
    restrict: 'A',
    templateNamespace: 'html',
    scope: false,
    controller: function($scope, $element, $attrs, $transclude, otherInjectables) { ...
  },
  controllerAs: 'stringAlias',
  require: 'siblingDirectiveName', // or // ['^parentDirectiveName', '?
optionalDirectiveName', '?^optionalParent'],
  compile: function compile(tElement, tAttrs, transclude) {
    return {
      pre: function preLink(scope, iElement, iAttrs, controller) { ... },
      post: function postLink(scope, iElement, iAttrs, controller) { ... }
    }
    // or
    // return function postLink( ... ) { ... }
  },
  // or
  // link: {
  //   pre: function preLink(scope, iElement, iAttrs, controller) { ... },
  //   post: function postLink(scope, iElement, iAttrs, controller) { ... }
  // }
  // or
```





SantaTodoApp component

```
@ComponentDirective
class SantaTodoApp {
  constructor() {
    this.newTodoTitle = '';
  }
  addTodo: function() { ... }
  removeTodo: function(todo) { ... }
  todosOf: function(filter) { ... }
}
```



SantaTodoApp component

```
@ComponentDirective({ ... })  
class SantaTodoApp { ... }
```

```
@TemplateDirective({ ... })  
class NgRepeat { ... }
```

```
@DecoratorDirective({ ... })  
class NgClass { ... }
```



SantaTodoApp component

```
@ComponentDirective
class SantaTodoApp {
  constructor() {
    this.newTodoTitle = '';
  }
  addTodo: function() { ... }
  removeTodo: function(todo) { ... }
  todosOf: function(filter) { ... }
}
```





ng-repeat and \$parent

```
Selected user: {{selectedUser.name}}  
<div ng-repeat="user in users"  
      ng-click="selectedUser = user">  
</div>
```



ng-repeat and \$parent

```
Selected user: {{selectedUser.name}}  
<div ng-repeat="user in users"  
      ng-click="$parent.selectedUser = user">  
</div>
```



Registering directives

```
@ComponentDirective({  
  directives: [TabContainer, TabPane, NgRepeat]  
})  
class SantaTodoApp {  
  constructor() {  
    this.newTodoTitle = '';  
  }  
  addTodo: function() { ... }  
  removeTodo: function(todo) { ... }  
  todosOf: function(filter) { ... }  
}
```



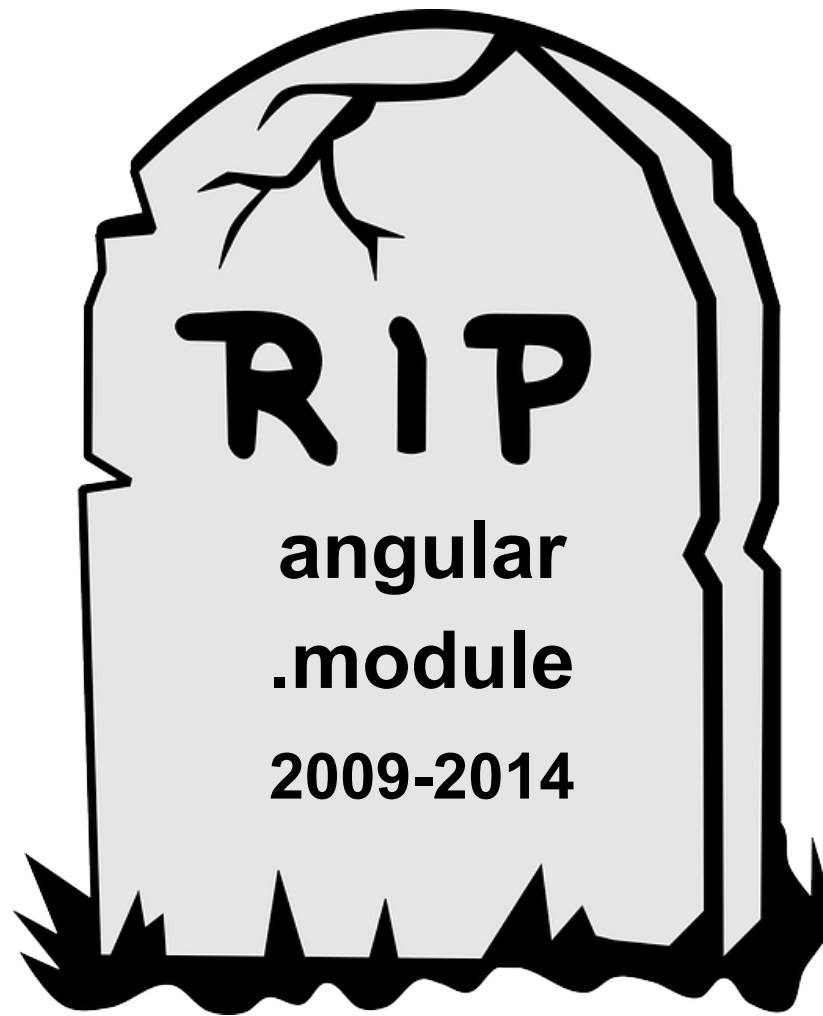
Dependency Injection



Defining services

```
class TodoStore {  
  constructor(win:Window) {  
    this.win = win;  
  }  
  add(todo) {  
    // access this.win.localStorage ...  
  }  
  remove(todo) { ... }  
  todosOf(filter) { ... }  
}
```





Using services

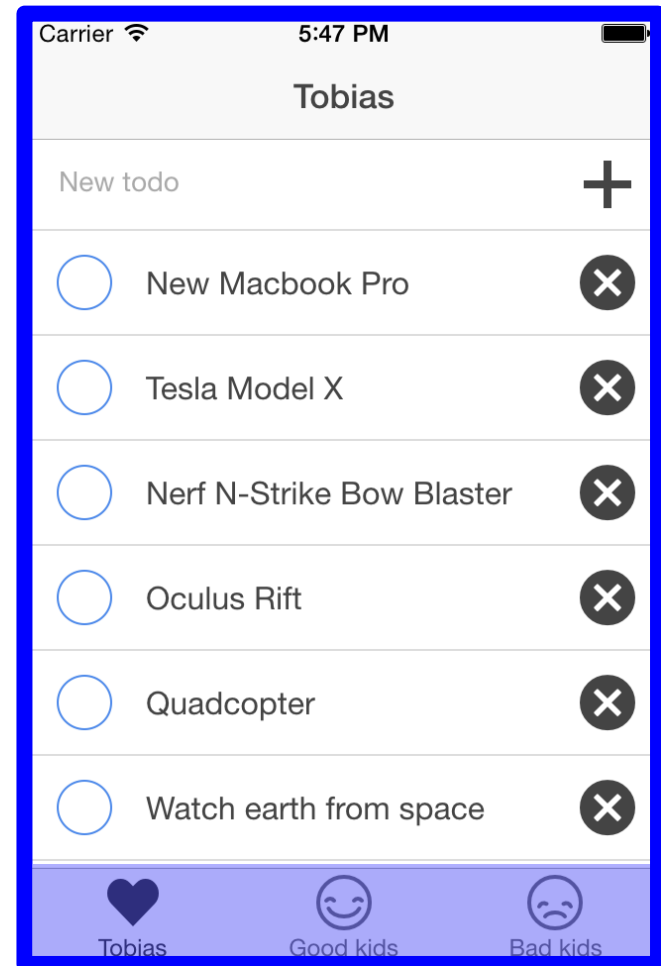
```
import {TodoStore} from './store';

@ComponentDirective({
  directives: [TabContainer, TabPane, NgRepeat]
})
class SantaTodoApp {
  constructor(store:TodoStore) {
    ...
  }
  ...
}
```



Directives and DI

```
<tab-container>  
  <tab-pane title="Tobias">  
    New Macbook Pro  
    Tesla Model X  
    ...  
  </tab-pane>  
  <tab-pane title="Good kids">  
    ...  
  </tab-pane>  
  <tab-pane title="Bad kids">  
    ...  
  </tab-pane>  
</tab-container>
```



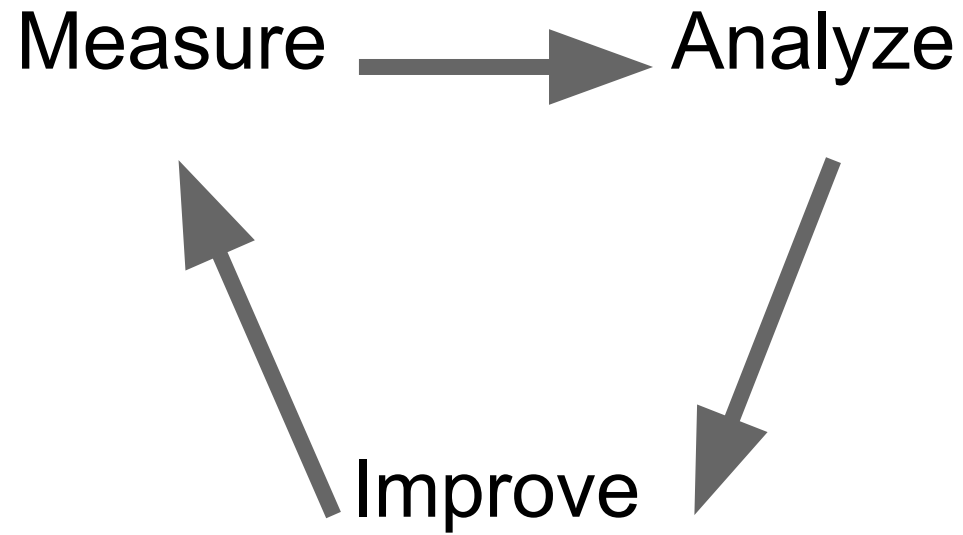
Directives and DI

```
class TabPane {  
    constructor(  
        tabContainer: TabContainer,  
        element: HTMLElement  
    ) { ... }  
    ...  
}
```

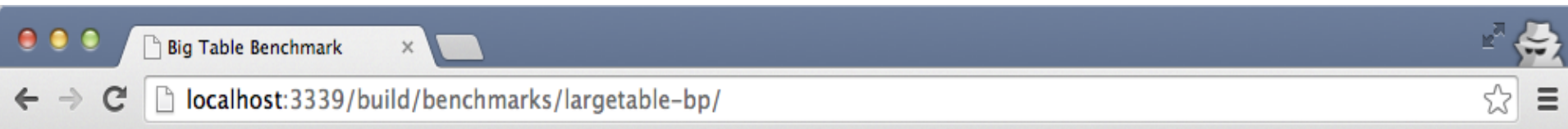
```
class TabContainer {  
    constructor(tabs: Query<TabPane>) { ... }  
    ...  
}
```



Performance



Measure



Benchmarkpress

Loop

Once

Loop 25x

Profile

Results:

	test time (ms)	gc time (ms)	garbage (KB)	retained memory (KB)
destroy	mean: 205.79ms ± 1% (stddev 2.12) (min 204.29 / max 207.29)	mean: 58.85ms (combined: 264.64ms)	mean: 170.84KB	mean: -17.44KB
	207.29	62.84	256.27	-78.02
	204.29	54.86	85.41	43.15
	mean: 156.67ms ± 1% (stddev 2.12) (min 154.55 / max 158.79)	mean: 35.21ms (combined: 264.64ms)	mean: 138.38KB	mean: 4881.70KB

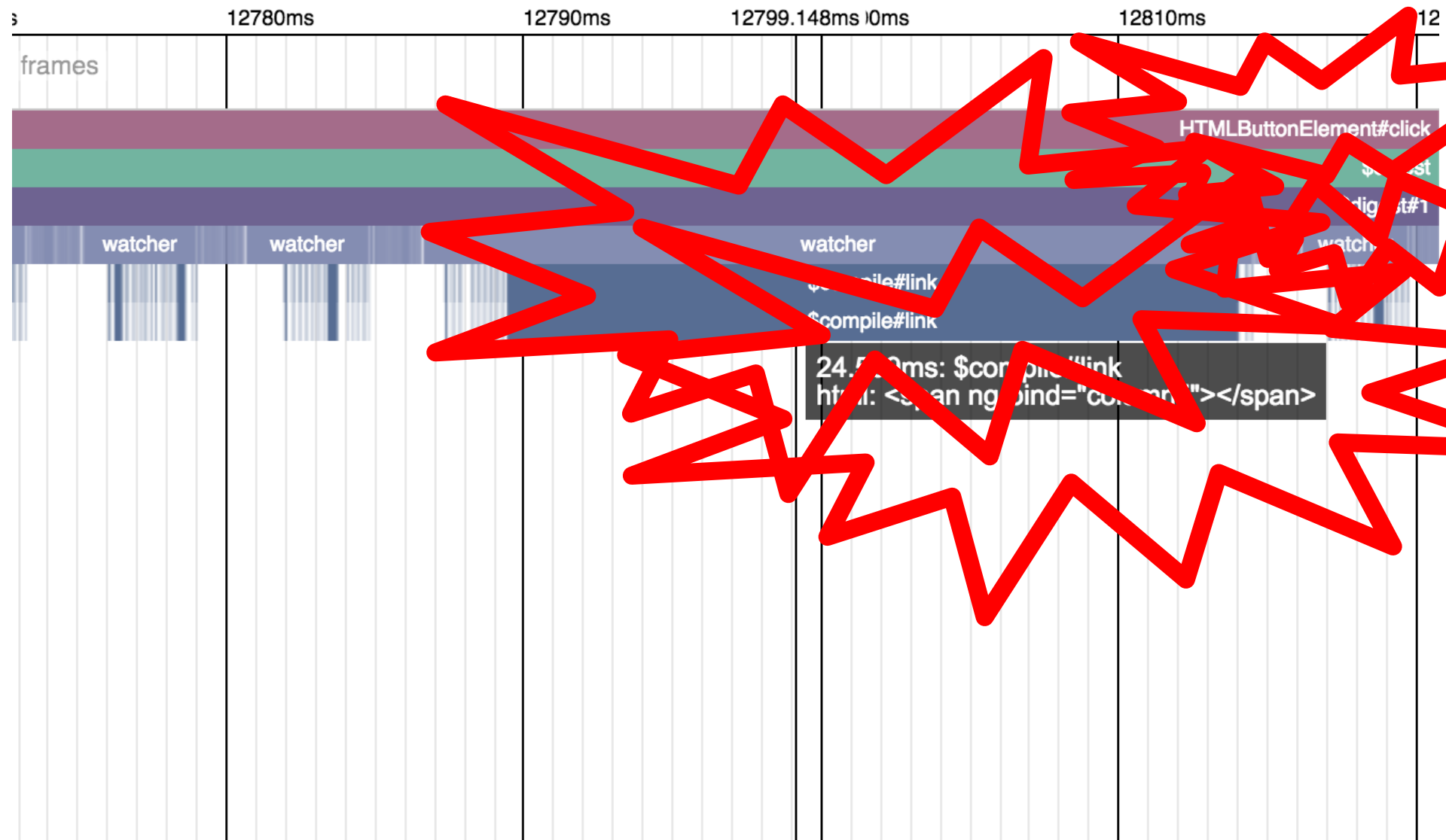
Macro optimizations



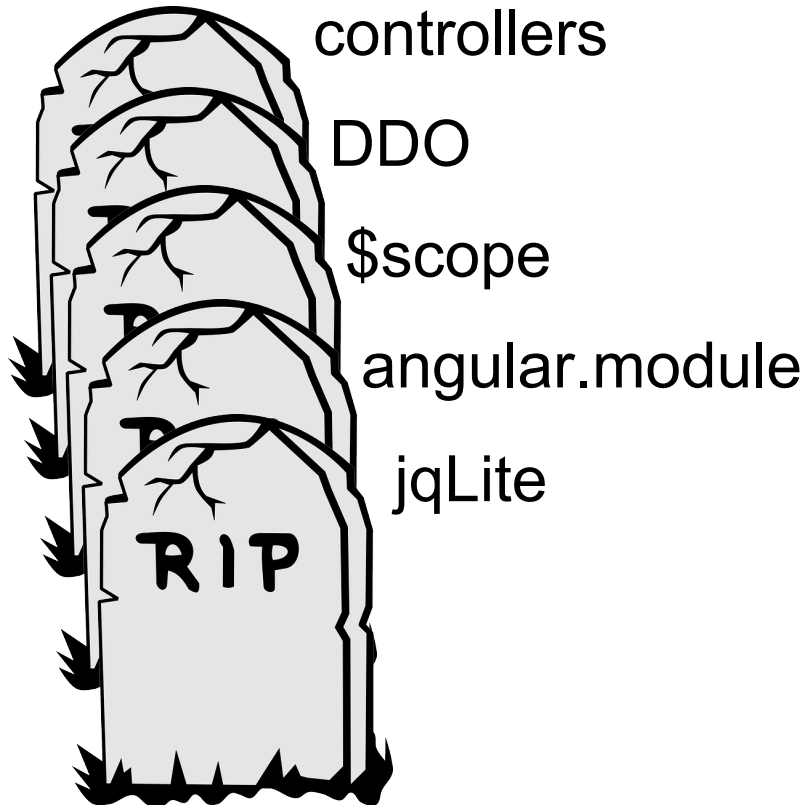
Analyze



Analyze - Web Tracing Framework



Summary



+ generic binding syntax

+ DI query mechanism

+ benchpress

+ WTF instrumentation

