# 02 AngularJS

## Framework Analysis

Public Code Repository

by
Sergey N. Bolshchikov
http://bolshchikov.net
sergey.bolshchikov@new-proimage.com
New ProImage, 2012

# Outline

I. Introduction
II. Philosophy

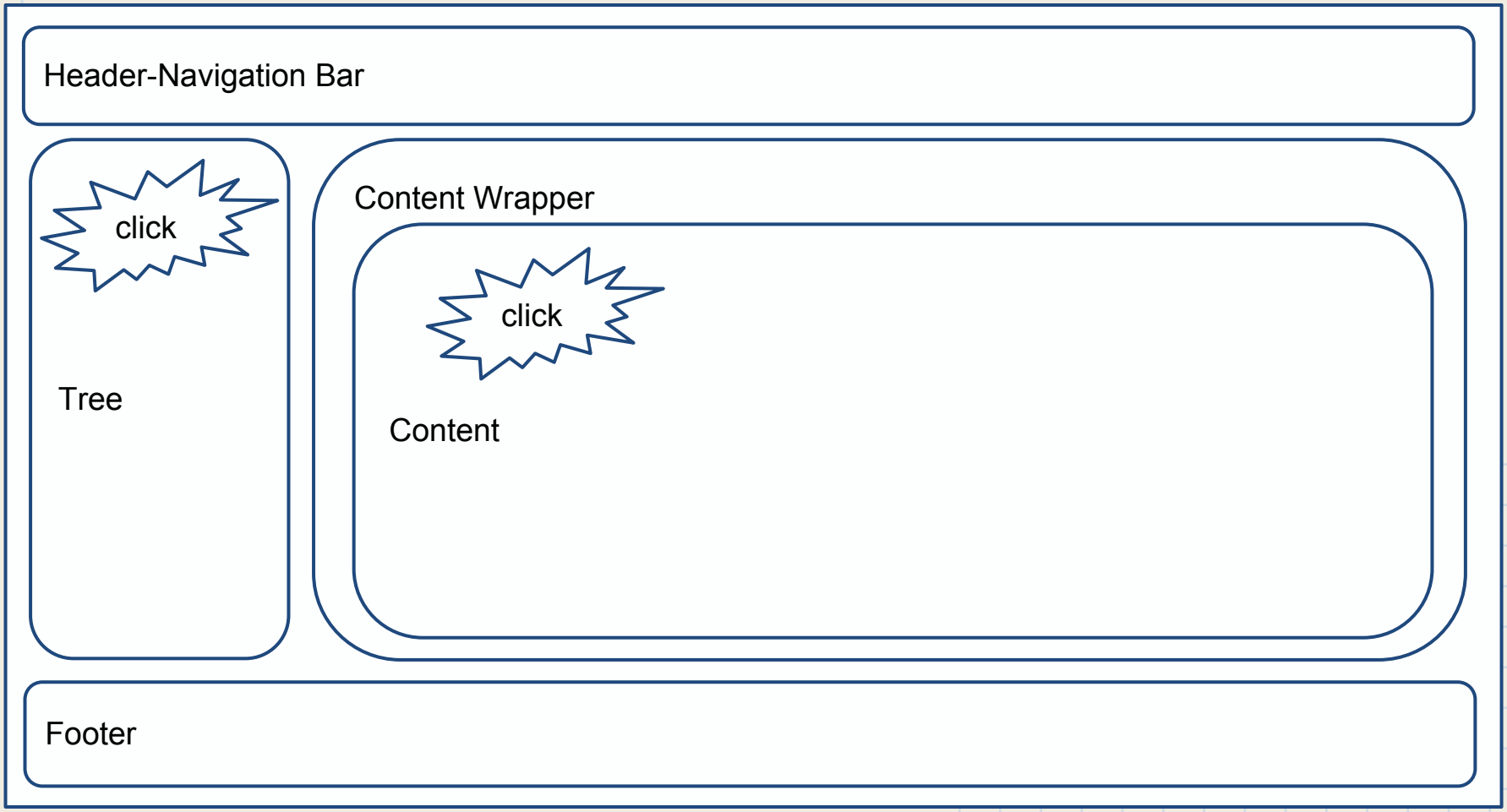
# Outline

I. Introduction
II. Philosophy

**No outline today**
**Just dive in**

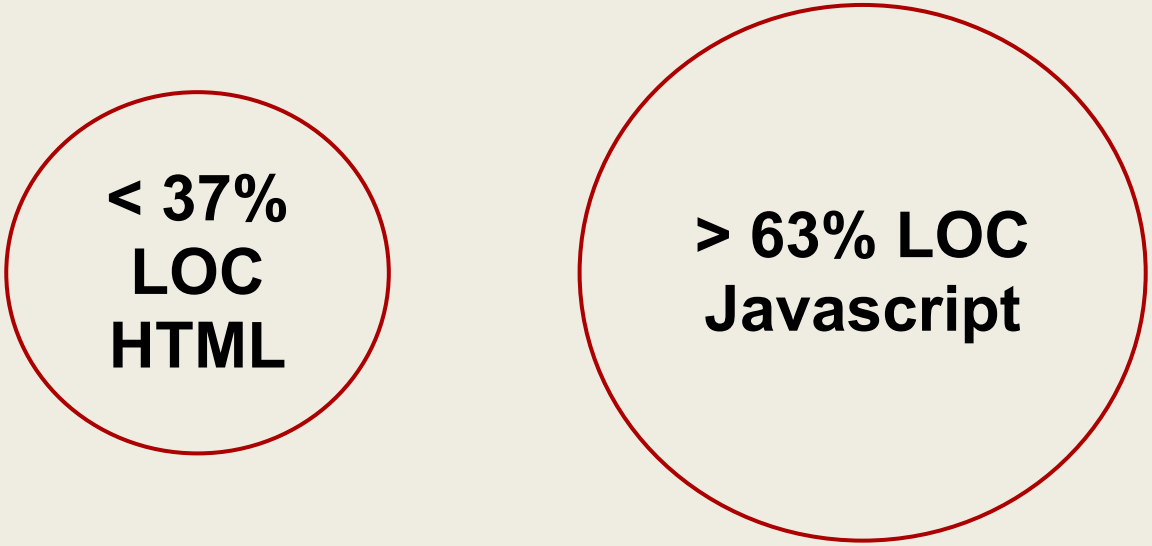
# Introduction

I want to build well structured and dynamic web application.



# Introduction

Traditional solution:

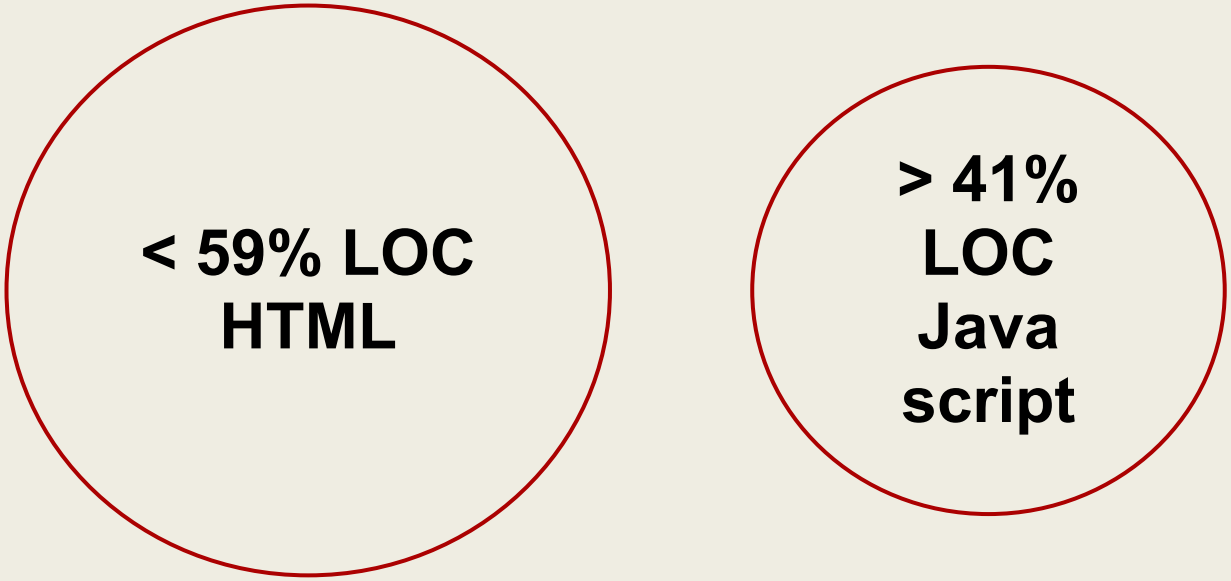


# Philosophy

- Angular is what HTML could have been if it had been designed for applications.
- HTML is a great declarative language for static documents. It does not contain much in the way of creating application.
- Building web-applications is an exercise in what do I have to do, so that I trick the browser in to do what I want.
- That's why we have frameworks - set of utility functions and libraries for DOM manipulation.
- Angular takes another approach.
- Angular teaches the browser new syntax.

# Introduction

AngularJS solution:



# Static HTML

```
<!doctype html>
<html lang="en" ng-app>
<head>
  <meta charset="utf-8">
  <title>My AngularJS App</title>
</head>
<body>
  <table class="table table-stripped">
    <thead>
      <tr>
          <th>ID</th><th>Complete</th><th>Name</th><th>Deadline</th>
      </tr>
    </thead>
    <tbody>
        <tr>
            <td>1001</td><td>false</td><td>Do Groceries</td><td>01/08/12</td>
        </tr>
        <tr>
            <td>1002</td><td>false</td><td>Barber the cat</td><td>01/08/12</td>
        </tr>
    </tbody>
  </table>
</body>
</html>
```

See example live

# Declarative HTML

```
<!doctype html>
<html lang="en" ng-app>
<head>
  <meta charset="utf-8">
  <title>My AngularJS App</title>
</head>
<body>
  <table class="table table-stripped" ng-controller="TodoCtrl">
    <thead>
      <tr>
          <th>ID</th><th>Complete</th><th>Name</th><th>Deadline</th>
      </tr>
    </thead>
    <tbody>
        <tr ng-repeat="todo in todos">
          <td ng-click="setTrue(todo.id)">{{todo.id}}</td>
          <td>{{todo.done}}</td>
          <td>{{todo.name}}</td>
          <td>{{todo.deadline}}</td>
        </tr>
    </tbody>
  </table>

</body>
</html>
```

See example live

# Declarative HTML

```
<!doctype html>
<html lang="en" ng-app>
<head>
  <meta charset="utf-8">
  <title>My AngularJS App</title>
</head>
<body>
  <table class="table table-stripped" ng-controller="TodoCtrl">
    <thead>
      <tr>
          <th>ID</th><th>Complete</th><th>Name</th><th>Deadline</th>
      </tr>
    </thead>
    <tbody>
        <tr ng-repeat="todo in todos">
          <td ng-click="setTrue(todo.id)">{{todo.id}}</td>
          <td>{{todo.done}}</td>
          <td>{{todo.name}}</td>
          <td>{{todo.deadline}}</td>
        </tr>
    </tbody>
  </table>

</body>
</html>
```

# Declarative HTML

```
<!doctype html>
<html lang="en" ng-app>
<head>
  <meta charset="utf-8">
  <title>My AngularJS App</title>
</head>
<body>
  <table class="table table-stripped" ng-controller="TodoCtrl">
    <thead>
      <tr>
          <th>ID</th><th>Complete</th><th>Name</th><th>Deadline</th>
      </tr>
    </thead>
    <tbody>
        <tr ng-repeat="todo in todos">
          <td ng-click="setTrue(todo.id)">{{todo.id}}</td>
          <td>{{todo.done}}</td>
          <td>{{todo.name}}</td>
          <td>{{todo.deadline}}</td>
        </tr>
    </tbody>
  </table>

</body>
</html>
```

# Declarative HTML

```
<!doctype html>
<html lang="en" ng-app>
<head>
  <meta charset="utf-8">
  <title>My AngularJS App</title>
</head>
<body>
  <table class="table table-stripped" ng-controller="TodoCtrl">
    <thead>
      <tr>
          <th>ID</th><th>Complete</th><th>Name</th><th>Deadline</th>
      </tr>
    </thead>
    <tbody>
        <tr ng-repeat="todo in todos">
          <td ng-click="setTrue(todo.id)">{{todo.id}}</td>
          <td>{{todo.done}}</td>
          <td>{{todo.name}}</td>
          <td>{{todo.deadline}}</td>
        </tr>
    </tbody>
  </table>

</body>
</html>
```

# Declarative HTML

```
<!doctype html>
<html lang="en" ng-app>
<head>
  <meta charset="utf-8">
  <title>My AngularJS App</title>
</head>
<body>
  <table class="table table-stripped" ng-controller="TodoCtrl">
    <thead>
      <tr>
          <th>ID</th><th>Complete</th><th>Name</th><th>Deadline</th>
      </tr>
    </thead>
    <tbody>
        <tr ng-repeat="todo in todos">
          <td ng-click="setTrue(todo.id)">{{todo.id}}</td>
          <td>{{todo.done}}</td>
          <td>{{todo.name}}</td>
          <td>{{todo.deadline}}</td>
        </tr>
    </tbody>
  </table>

</body>
</html>
```

# MVC

Angular says:
"There are many ways to structure the code for an application.
For Angular apps, we encourage the use of the Model-View-Controller (MVC) design pattern to decouple the code and to separate concerns.
With that in mind, let's use a little Angular and JavaScript to add model, view, and controller components to our app."

# Controller

- a controller is a JavaScript function
- It contains data
- It specifies the behavior
- It should contain only the business logic needed for a single view.

# Controller

```
<table class="table table-stripped" ng-controller="TodoCtrl">
function TodoCtrl(scope) {
    scope.todos = [
    {
        'id': 1001,
        'done': false,
        'name': 'Do Groceries',
        'deadline': new Date()
    },
    {
        'id': 1002,
        'done': false,
        'name': 'Barber the cat',
        'deadline': new Date()
    }];
    scope.setTrue = function(id) {
        var el = (function(id){
            for (var i=0; i<scope.todos.length; i++) {
                if (scope.todos[i].id === id) {
                    return scope.todos[i]
                }
            }
        })(id);
        el.done = true;
    }
}
TodoCtrl.$inject = ['$scope'];
```

# Controller

```
<table class="table table-stripped" ng-controller="TodoCtrl">
function TodoCtrl(scope) {
    scope.todos = [
    {
        'id': 1001,
        'done': false,
        'name': 'Do Groceries',
        'deadline': new Date()
    },
    {
        'id': 1002,
        'done': false,
        'name': 'Barber the cat',
        'deadline': new Date()
    }];
    scope.setTrue = function(id) {
        var el = (function(id){
            for (var i=0; i<scope.todos.length; i++) {
                if (scope.todos[i].id === id) {
                    return scope.todos[i]
                }
            }
        })(id);
        el.done = true;
    }
}
TodoCtrl.$inject = ['$scope'];
```

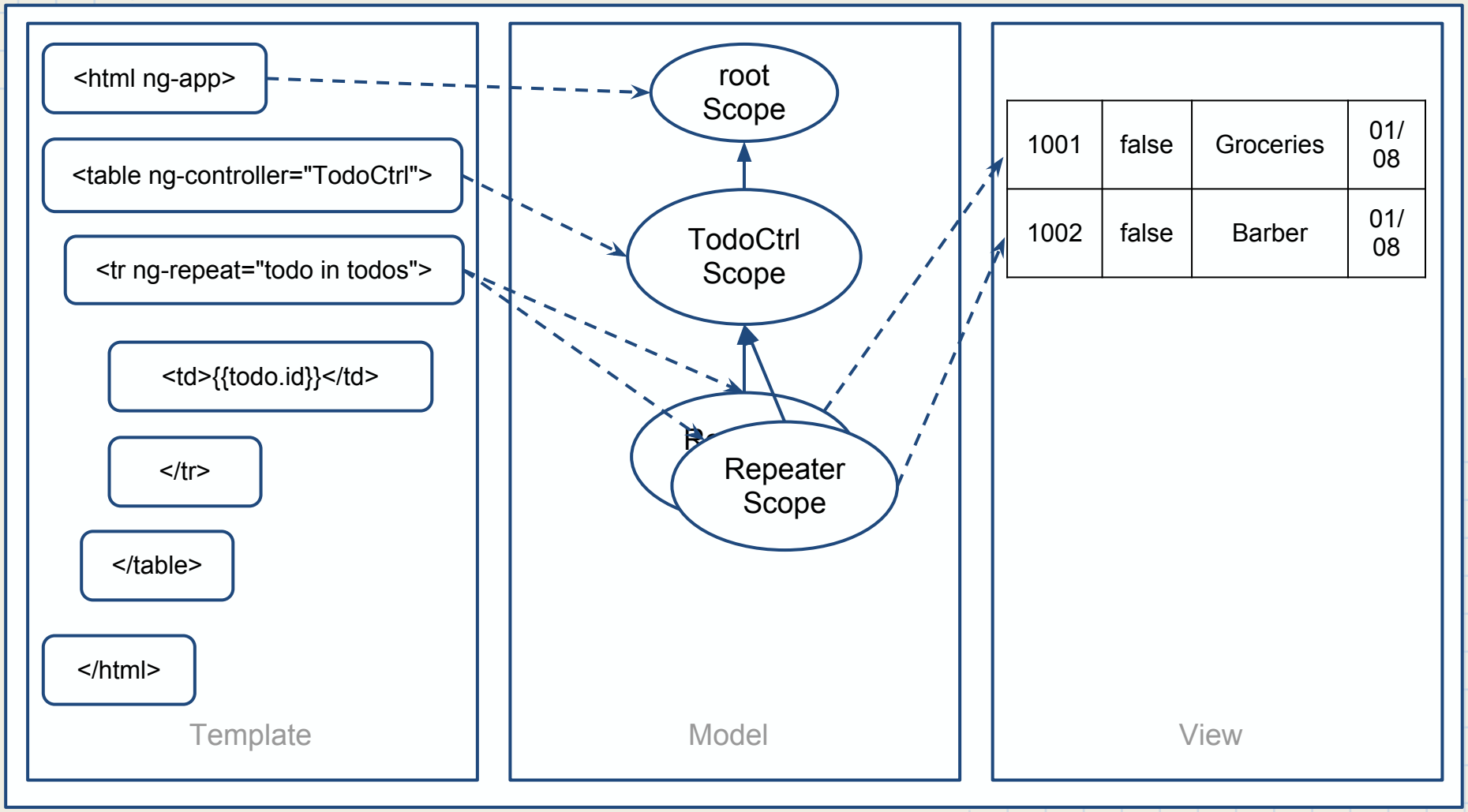
# Controller

```
<table class="table table-stripped" ng-controller="TodoCtrl">
function TodoCtrl(scope) {
    scope.todos = [
    {
        'id': 1001,
        'done': false,
        'name': 'Do Groceries',
        'deadline': new Date()
    },
    {
        'id': 1002,
        'done': false,
        'name': 'Barber the cat',
        'deadline': new Date()
    }];
    scope.setTrue = function(id) {
        var el = (function(id){
            for (var i=0; i<scope.todos.length; i++) {
                if (scope.todos[i].id === id) {
                    return scope.todos[i]
                }
            }
        })(id);
        el.done = true;
    }
}
TodoCtrl.$inject = ['$scope'];
```

# Scope

- an object that refers to the application model (application itself)
- an execution context for expressions like {{ todo.name }}
- Scopes are arranged in hierarchical structure which mimic the DOM structure of the application
- Scopes can watch expressions and propagate events

# Scope

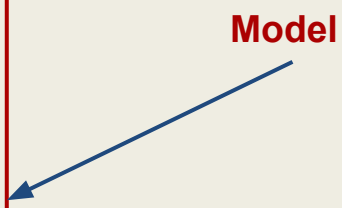

<html ng-app>
<table ng-controller="TodoCtrl">
<tr ng-repeat="todo in todos">
<td>{{todo.id}}</td>
</tr>
</table>
</html>
Template
root Scope
TodoCtrl Scope
Repeater Scope
Model
1001 false Groceries 01/08
1002 false Barber 01/08
View

# Controller

```
<table class="table table-stripped" ng-controller="TodoCtrl">
function TodoCtrl(scope) {
    scope.todos = [
    {
        'id': 1001,
        'done': false,
        'name': 'Do Groceries',
        'deadline': new Date()
    },
    {
        'id': 1002,
        'done': false,
        'name': 'Barber the cat',
        'deadline': new Date()
    }];
    scope.setTrue = function(id) {
        var el = (function(id){
            for (var i=0; i<scope.todos.length; i++) {
                if (scope.todos[i].id === id) {
                    return scope.todos[i]
                }
            }
        })(id);
        el.done = true;
    }
}
TodoCtrl.$inject = ['$scope'];
```

# Model: attrs of Scope

```
<table class="table table-stripped" ng-controller="TodoCtrl">
function TodoCtrl(scope) {
    scope.todos = [
    {
        'id': 1001,
        'done': false,
        'name': 'Do Groceries',
        'deadline': new Date()
    },
    {
        'id': 1002,
        'done': false,
        'name': 'Barber the cat',
        'deadline': new Date()
    }];
    scope.setTrue = function(id) {
        var el = (function(id){
            for (var i=0; i<scope.todos.length; i++) {
                if (scope.todos[i].id === id) {
                    return scope.todos[i]
                }
            }
        })(id);
        el.done = true;
    }
}
TodoCtrl.$inject = ['$scope'];
```

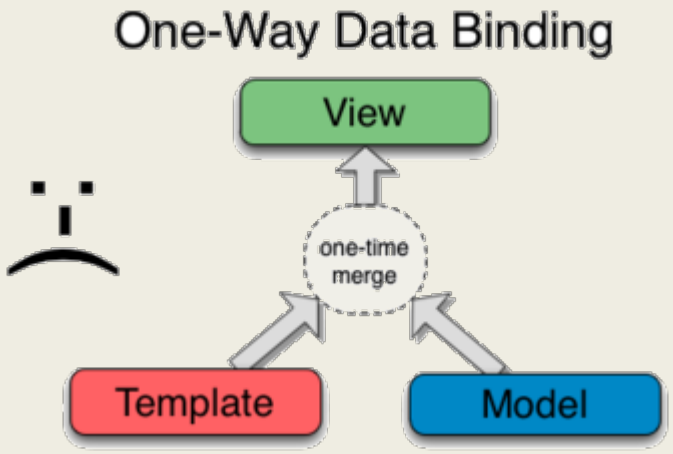


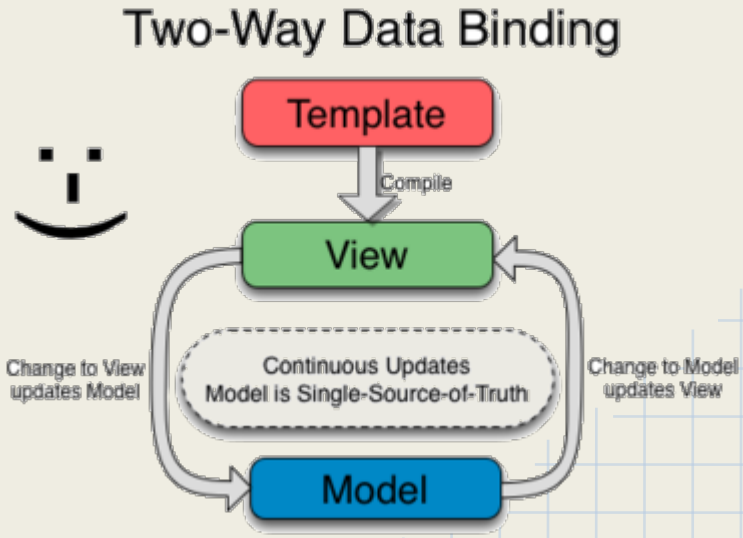
# Template

```
<!doctype html>
<html lang="en" ng-app>
<head>
  <meta charset="utf-8">
  <title>My AngularJS App</title>
</head>
<body>
  <table class="table table-stripped" ng-controller="TodoCtrl">
    <thead>
      <tr>
          <th>ID</th><th>Complete</th><th>Name</th><th>Deadline</th>
      </tr>
    </thead>
    <tbody>
        <tr ng-repeat="todo in todos">
          <td ng-click="setTrue(todo.id)">{{todo.id}}</td>
          <td>{{todo.done}}</td>
          <td>{{todo.name}}</td>
          <td>{{todo.deadline}}</td>
        </tr>
    </tbody>
  </table>

</body>
</html>
```

# Data-binding and interaction




See live example

# Routing

```
angular.module('myApp').
    config(['$routeProvider',
        function($routeProvider) {
            $routeProvider.when(
                '/folder/:name',
                {templateUrl: 'partials/folder.html',
                controller: FolderCtrl
        });
    }])
```

# Performance

- Browser support: IE 8+, Chrome, FF, Safari, Opera
- Framework size: 503KB
- Application size: 756KB