# BOSSY UI

**A suite of opinionated web development widgets made entirely with AngularJS**

*Sean German, Lukas Fink, Francisco Tadillo, Martin Mojica*

# Purpose

- 100%
  - AngularJS, No JQuery Dependencies
  - Themeable
  - Lightweight
  - Open Source
- Ease of Implementation and Maintainability.
- Designed from the ground up to be a full suite for end to end web development widgets.

# Importance of AngularJS

- Two-way Data binding
- MVC
- Custom-directive
- REST-friendly
- Advanced-Templates
- Dependency-Injection

# Product

The combobox, completely written in AngularJS, provides three features:

- Cascading Dropdown (dropdown choices are dependent on the selections of another dropdown).
- Checkbox Multiselect (allows user to select several options, select all or deselect all).
- Normal multi-select combobox.

# Demo – Bossy/UI combobox

**Cascading Dropdown**

Country: [ Select ▾ ]
State: [ Select ▾ ]
City: [ Select ▾ ]

**Checkbox Multiselect**

☐ Noodles
☐ Soup
☐ Chicken
☐ Salad

[ select all ]
[ deselect all ]

choices: []

**Multiselect**
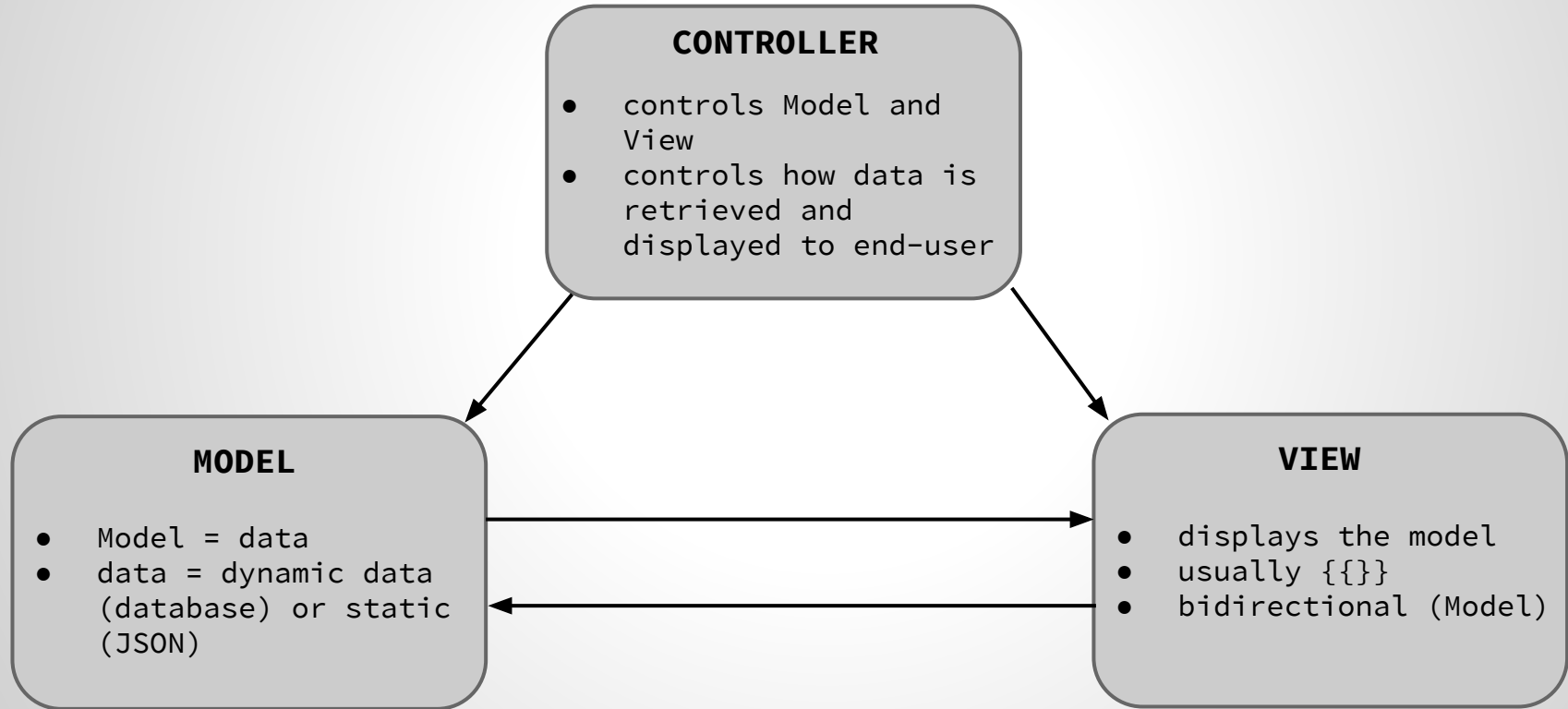
Select all
Deselect all
Chicken
Beef

choices: []

# Design

- follows the MVC architecture
  - Model to represent current state of the application
  - View to display the data
  - Controller to control the relation between Model and View


- program's logic is clearly separated from user interface

# MVC – AngularJS

**CONTROLLER**

- controls Model and View
- controls how data is retrieved and displayed to end-user

**MODEL**

- Model = data
- data = dynamic data (database) or static (JSON)

**VIEW**

- displays the model
- usually {{}}
- bidirectional (Model)

# Simple Angular MVC-example

controller

```
app.controller('CtrlName', function ($scope) {
    $scope.choices = {
        'california'    : 'sacramento',
        'texas'         : 'austin',
        'oregon'        : 'salem'
    }
});
```

model

view

```
<div> {{choices.california}} </div>
```

# Cascading Dropdown - Ctrl

Model

```
app.controller('AppCtrl', function($scope) {

    $scope.choices = {
        'Option A': {
            'Option A1': ['Option A1a', 'Option A1b', 'Option A1c'],
            'Option A2': ['Option A2a', 'Option A2b', 'Option A2c'],
            'Option A3': ['Option A3a', 'Option A3b', 'Option A3c']
        },
        'Option B': {
            'Option B1': ['Option B1a', 'Option B1b', 'Option B1c'],
            'Option B2': ['Option B2a', 'Option B2b', 'Option B2c'],
            'Option B3': ['Option B3a', 'Option B3b', 'Option B3c']
        },
        'Option C': {
            'Option C1': ['Option C1a', 'Option C1b', 'Option C1c'],
            'Option C2': ['Option C2a', 'Option C2b', 'Option C3b'],
            'Option C3': ['Option C3a', 'Option C3b', 'Option C3c']
        }
    };

})
```

connect to view with: ng-controller

# Cascading Dropdown - View

```
<select id="choiceOne" ng-model="choiceTwoOptions" ng-options="choiceOne for (choiceOne, choiceTwoOptions) in choices">
    <option value=''>Select</option>
</select>
```

- ng-model binds the view into the model
- ng-options to dynamically generate a list of option-tags using the array

# Checkbox Multiselect - Ctrl

Model →

```
app.controller('AppCtrl', function($scope) {

    $scope.choices = ['Option A', 'Option B', 'Option C', 'Option D'];

    $scope.name = {choices: []};

    $scope.selectAll = function() {
        $scope.name.choices = angular.copy($scope.choices);
    };

    $scope.deselectAll = function() {
        $scope.name.choices = [];
    };

});
```

stores Model and basic functions

```
app.directive('bossyCheckboxMultiselect'
```

custom directive to manipulate DOM-elements

# Checkbox Multiselect - View

```html
<label ng-repeat="choice in choices">
    <input type="checkbox" bossy-checkbox-multiselect="name.choices" bossy-list-value="choice"> {{choice}}
</label>
<br />
<button ng-click="selectAll()">select all</button>
<button ng-click="deselectAll()">deselect all</button>
```

- usage of directive within input-tag
- ng-click defines on-click action

# Verification

- Still in development.
- Working on implementing unit tests.
- Working on implementation of integration tests.

# Conclusion

Bossy UI's mission is to provide web developers of all levels, a smooth and headache-free development experience. Since their tools are built entirely with flexible AngularJS, web developers won't have to worry about any dependencies.

Building rich and interactive websites will become a much easier process for anyone who uses the widgets that we've built, and we are proud of that.