Day - 9

14.02.2025

Why Angular JS?

HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.

```html
1.  <!doctype html>
2.  <html ng-app>
3.    <head>
4.      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
5.    </head>
6.    <body>
7.      <div>
8.        <label>Name:</label>
9.        <input type="text" ng-model="yourName" placeholder="Enter a name here">
10.       <hr>
11.       <h1>Hello {{yourName}}!</h1>
12.     </div>
13.   </body>
14.  </html>
```

Data Binding

Data-binding is an automatic way of updating the view whenever the model changes, as well as updating the model whenever the view changes. This is awesome because it eliminates DOM manipulation from the list of things you have to worry about.

Controller

Controllers are the behavior behind the DOM elements. AngularJS lets you express the behavior in a clean readable form without the usual boilerplate of updating the DOM, registering callbacks or watching model changes.

Plain JavaScript

Unlike other frameworks, there is no need to inherit from proprietary types in order to wrap the model in accessors methods. AngularJS models are plain old JavaScript objects. This makes your code easy to test, maintain, reuse, and again free from boilerplate.

Create Components

Directives

Directives are a unique and powerful feature available in AngularJS. Directives let you invent new HTML syntax, specific to your application.

Reusable Components

We use directives to create reusable components. A component allows you to hide complex DOM structure, CSS, and behavior. This lets you focus either on what the application does or how the application looks separately.

Localization

An important part of serious apps is localization. AngularJS's locale aware filters and stemming directives give you building blocks to make your application available in all locales.

```
1.   angular.module('components', [])
2.
3.    .directive('tabs', function() {
4.      return {
5.        restrict: 'E',
6.        transclude: true,
7.        scope: {},
8.        controller: function($scope, $element) {
9.          var panes = $scope.panes = [];
10.
11.         $scope.select = function(pane) {
12.           angular.forEach(panes, function(pane) {
13.             pane.selected = false;
14.           });
15.           pane.selected = true;
16.         }
17.
18.         this.addPane = function(pane) {
19.           if (panes.length == 0) $scope.select(pane);
20.           panes.push(pane);
21.         }
22.        },
23.        template:
24.          '<div class="tabbable">' +
25.            '<ul class="nav nav-tabs">' +
```

```
26.        '<li ng-repeat="pane in panes" ng-class="{active:pane.selected}">'+
27.          '<a href="" ng-click="select(pane)">{{pane.title}}</a>' +
28.         '</li>' +
29.        '</ul>' +
30.        '<div class="tab-content" ng-transclude></div>' +
31.      '</div>',
32.      replace: true
33.    };
34.  })
35.
36.  .directive('pane', function() {
37.    return {
38.      require: '^tabs',
39.      restrict: 'E',
40.      transclude: true,
41.      scope: { title: '@' },
42.      link: function(scope, element, attrs, tabsController) {
43.        tabsController.addPane(scope);
44.      },
45.      template:
46.        '<div class="tab-pane" ng-class="{active: selected}" ng-transclude>' +
47.        '</div>',
48.      replace: true
49.    };
50.  })
```

Deep Linking

A deep link reflects where the user is in the app. This is useful so users can bookmark and email links to locations within the app. Round trip apps get this automatically, but AJAX apps by their nature do not. AngularJS combines the benefits of deep linking with desktop app-like behavior.

Form Validation

Client-side form validation is an important part of a great user experience. AngularJS lets you declare the validation rules of the form without having to write JavaScript code. Write less code, go have beer sooner.

Server Communication

AngularJS provides built-in services on top of XHR as well as various other backends using third party libraries. Promises further simplify your code by handling asynchronous return of data.