Introduction to AngularJS JavaScript Framework

Tuna Tore

What is AngularJS

- **AngularJS** is **%100** JavaScript Client Side Framework
- **Google** employee Miško Hevery started developing it and now it is **officially** supported by Google
- You can run **AngularJS** applications on mobile and desktop browsers
- It is **FREE** and open source
- It has brackets with **ng** prefix hence it sounds like Angular

https://angularjs.org/

The benefits of using AngularJS

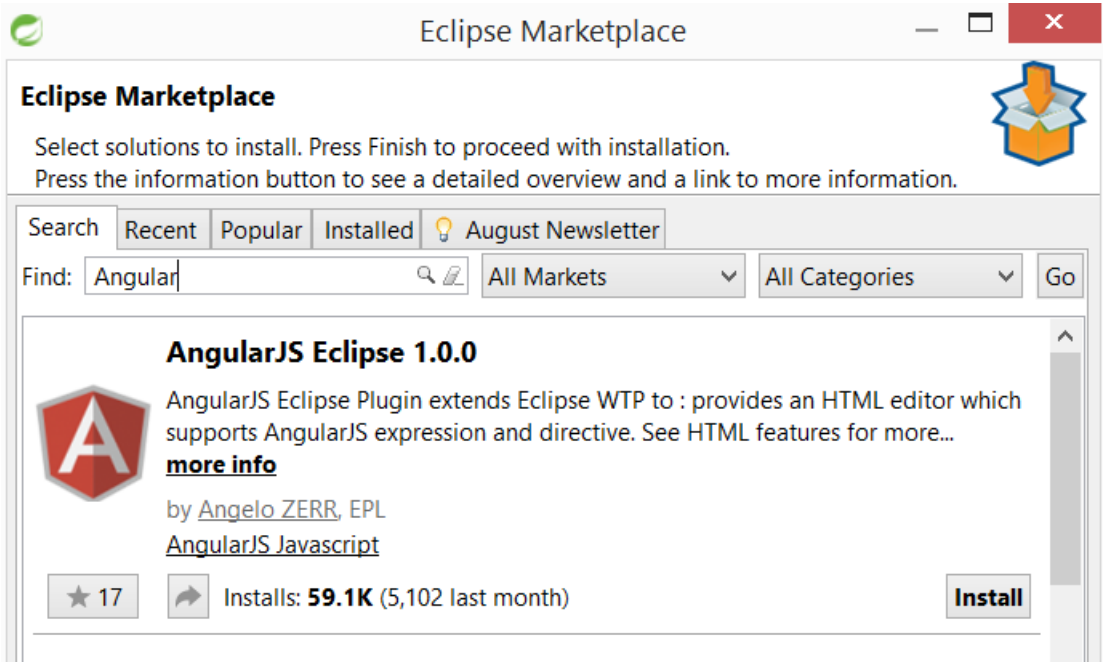
- **AngularJS** provides separation of concerns while developing applications
- It separates application layers to models, views and controllers (**MVC**)
- Provides **easily maintainable** and **unit testable** code
- Uses **Dependency Injection** Design principle in order to inject dependencies of components
- It extends **HTML DOM** model by adding attributes to HTML tags
- It allows developers to design rich client side user interfaces

More information about AngularJS

- **AngularJS** uses the minified version of **jQuery** called **jQlite** in order to modify HTML DOM tree (**JQuery** is a JavaScript library provides easy use of JavaScript language benefits)
- You should use **angular.min.js** JavaScript on production environment hence the scripts in this file is minimized using compression and it is not human readable
- It supports most of the current browsers such as Safari, Chrome, Firefox, Opera 15, IE9 and mobile browsers (Android, Chrome Mobile, iOS Safari).

Installing AngularJS Plugin for STS(Spring Tool Suite) or Eclipse

- **Help>Eclipse Marketplace**



More information about AngularJS

- **AngularJS** uses the minified version of **jQuery** called **jQlite** in order to modify HTML DOM tree (**JQuery** is a JavaScript library provides easy use of JavaScript language benefits)
- You should use **angular.min.js** JavaScript on production environment hence the scripts in this file is minimized using compression and it is not human readable
- It supports most of the current browsers such as Safari, Chrome, Firefox, Opera 15, IE9 and mobile browsers (Android, Chrome Mobile, iOS Safari).

(SPA)Single Page Application

- **Single Page Application** is a web application or website designed on only one page
- Using **AngularJS**, developers can display and re-render the parts of the web application without refreshing whole page
- The final implementation of the application will be similar to desktop application
- Appropriate sources are loaded based on user actions
- Requests to servers are done through Ajax calls and the response can be in XML, JSON or HTML output

Two way data binding in AngularJS

- **Data binding** is the synchronization of model and view data automatically through the framework
- **AngularJS** handles the synchronization of data between layers
- When the **model changes then the view reflects** changes or changing the view will change the model as well

Expressions in AngularJS

- **AngularJS** expressions are written in double braces
- Expressions bind model data to **HTML** pages
- It is evaluated by browsers based on scope object (which is used to store objects)
- **Expressions** can work with arrays, numbers, objects, functions and strings
- **Valid examples**

  {{ 1 + 1 }}

  {{ udemyCourse.name}}

  {{ courses[0]}}

Modules in AngularJS

- **AngularJS** supports modular development and this provides separation of business layers
- **AngularJS** application starts by defining main module
- **Modules** can be defined in different js files
- **Modules** can be injected to other modules
- Developers can define controllers, services, filters, factories, directives, etc in different modules
- Developers can import modules using
  <script src="test.js"></script>

Controllers in AngularJS

- **Controllers** are defined and attached to **HTML** DOM using **ng-controller** directive
- When a controller is created by **AngularJS** , a **$scope** object is attached into controllers
- Developers should use controllers in order to set the initial state of variables in **$scope**
- **$scope** functions and behaviors have to be done inside controllers
- Controllers should not be used to define complex programming

$scope in AngularJS

- **Scopes** provide the separation between the **model** and **view**
- It represents the data model of applications
- **Scope** is used as a data model between controllers and views
- Both controllers and directives can access to scopes
- Scopes are injected into controllers as a parameter by **AngularJS**
- Scopes are created in hierarchical order

Directives in AngularJS

- **Directives** are markers on **HTML DOM** which add functionality to HTML pages
- **AngularJS** directives add additional behaviors to HTML tags
- **Directives** can be HTML element attribute, element tag/name or css class
- **AngularJS** has built-in directives such as **ng-bind, ng-model, ng-repeat** etc
- Developers can also add custom directives to **AngularJS** applications
- Custom **directives** are useful while changing the content of **HTML DOM**