

# הנדסת תוכנה – תירגול

## פיתוח צד לקוח



# AngularJS - הקדמה

\* **AngularJS** הינה JS Framework (פלטפורמה) לפיתוח אפליקציות Web בצד הלקוח מבית היוצר של חברת גוגל

\* הפלטפורמה בנויה בשיטת MVC שכוללת Model, View, Controller וכן Services

\* ובשונה מפלטפורמות פיתוח אחרות, היא מסוגלת להשתלט על כל חלקי האפליקציה ולהכתיב את אופי העבודה כולו

\* **Single Page Application (SPA)**

\* **מנגנון Two-Way Data-Binding**

\* **UI Bootstrap**

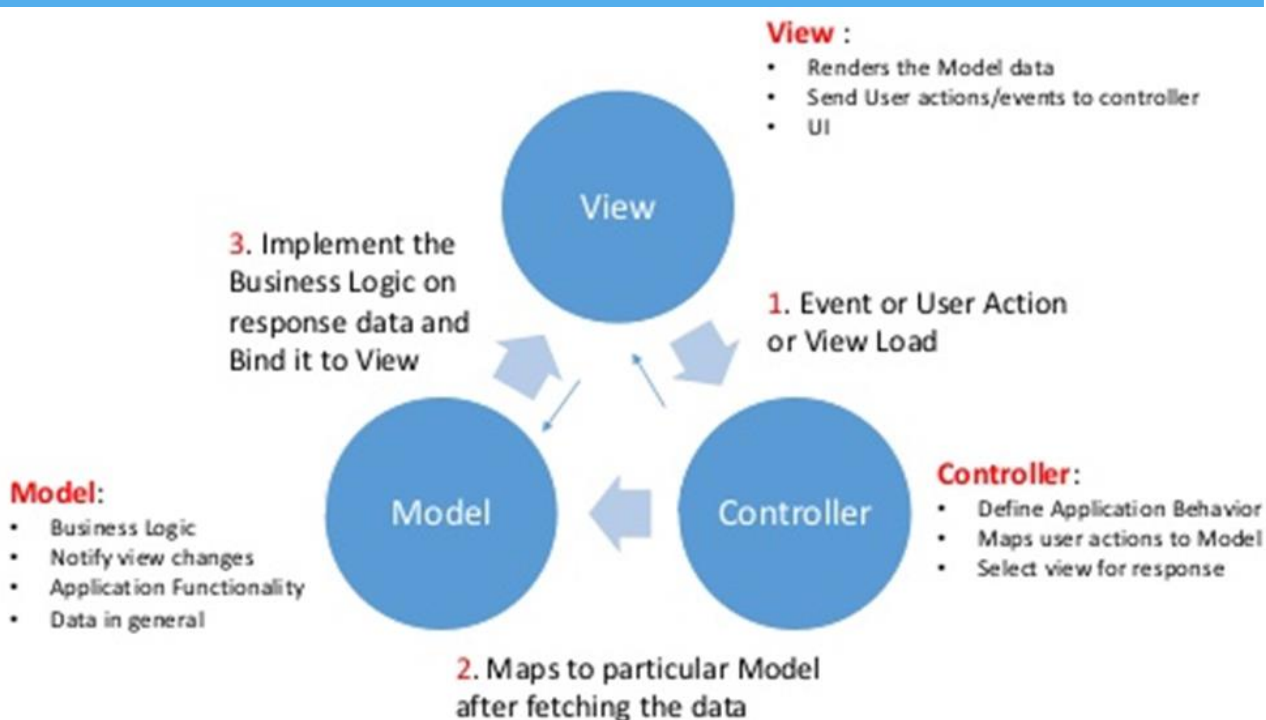
<https://angularjs.org/>

# MVC - Model View Controller

\* **Model (Data)** – הנתונים של האפליקציה

\* **View (UI)** – המסכים של האפליקציה, מה שהמשתמש רואה (Html UI)

\* **Controller (Logic)** – הלוגיקה של האפליקציה, בקר בין התצוגה לנתונים



# Single Page Application (SPA)

\* **single web app** - הינה אפליקציית web אשר מתאימה לעמוד אינטרנטי אחד במטרה לספק זרימה שוטפת יותר של חווית ממשק משתמש (UX)

\* ב-AngularJS כל הקוד מצוי כבר בתוך טעינת הדף הבודד ב-client, או שהמשאבים המתאימים מוטענים בצורה דינאמית ומתווספים לדף על פי הצורך, בד"כ בתגובה לפעולה של המשתמש

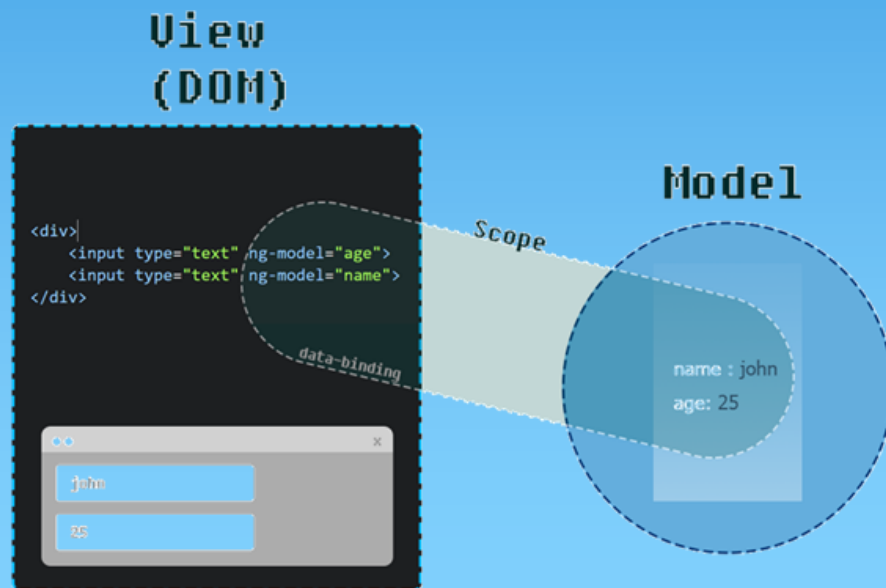
\* הדף אינו נטען מחדש בשום שלב בתהליך, וכן אינו מבצע העברה לדף אחר, כאשר "עוברים דף" זו רק אשליה, ובעצם אותו דף html יחיד "מציג פנים אחרות" באמצעות העלמה של חלקים, והצגת חלקים אחרים.

# Two-Way Data-Binding

\* **data binding** - הצמדה בין ה- Model אל ה- View תוך כדי מתן מענה לשינויים שקורים בהתאם להתנהגות המשתמש.

\* מי שבעצם מחבר בין שני החלקים ומקשר ביניהם הוא כלי עזר בשם scope

\* ב- AngularJS "כלי עזר" כמו ה- scope נקרא Service



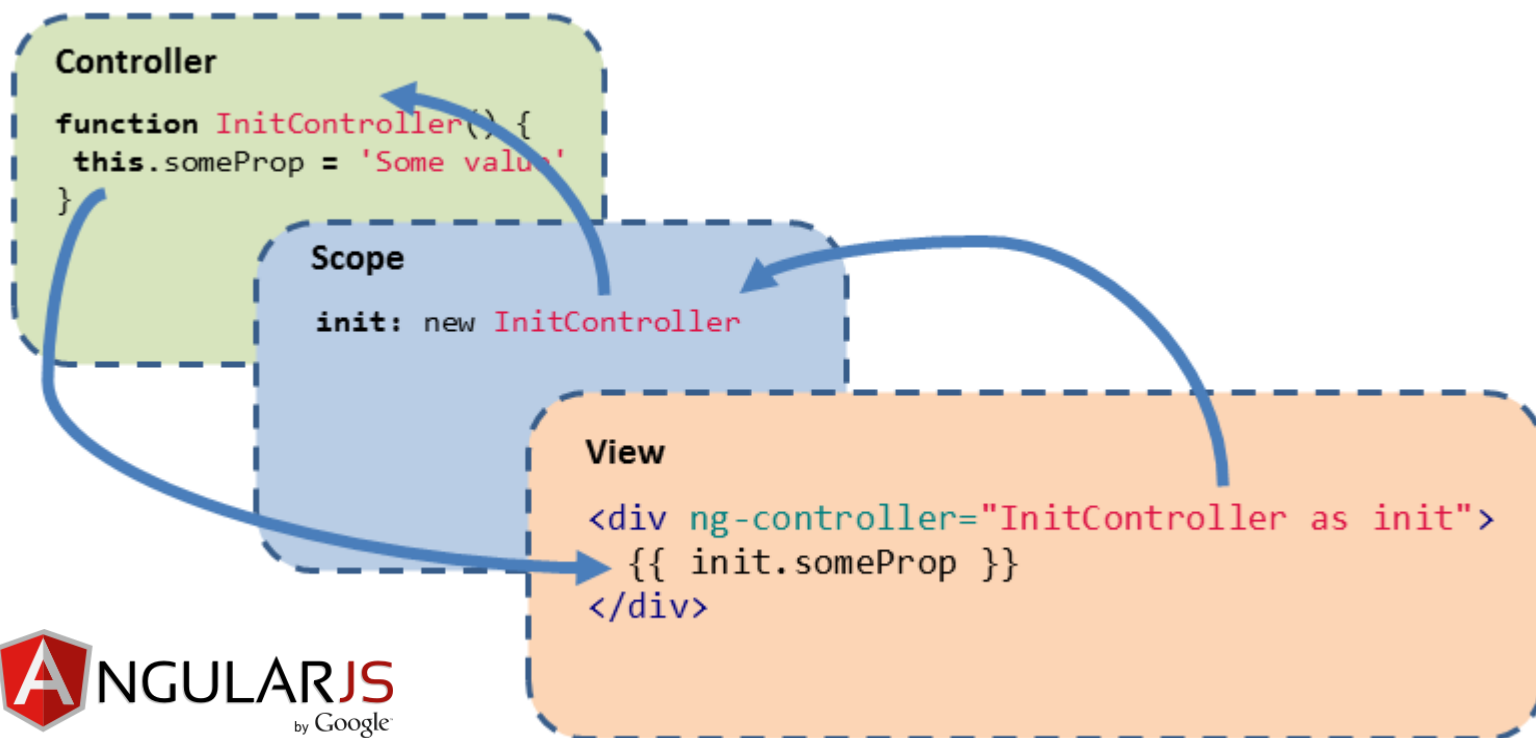
# איך זה עובד?

- \* הלוגיקה של האפליקציה נמצאת בתוך קובץ js ראשי (לרוב נקרא app.js), תחת Module מסוים
  - \* ה-Module הינו הייצוג המרכזי של האפליקציה שלנו, בו אנו מגדירים את תכונות האפליקציה ואת ההתנהגות
  - \* ה-Module הראשי יכול להיות מורכב מעוד כמה Modules שמכילים גם Controllers, Directives, Services ופונקציות (נרחיב בהמשך)
  - \* הגדרה של Module:
- ```
var mod = angular.module('myApp', []);
```
- \* ה-Controllers הוא חלק האחראי על טיפול בנתונים (Model) לצורך הצגתם (View). ה-Controllers מכיל פונקציות JavaScript

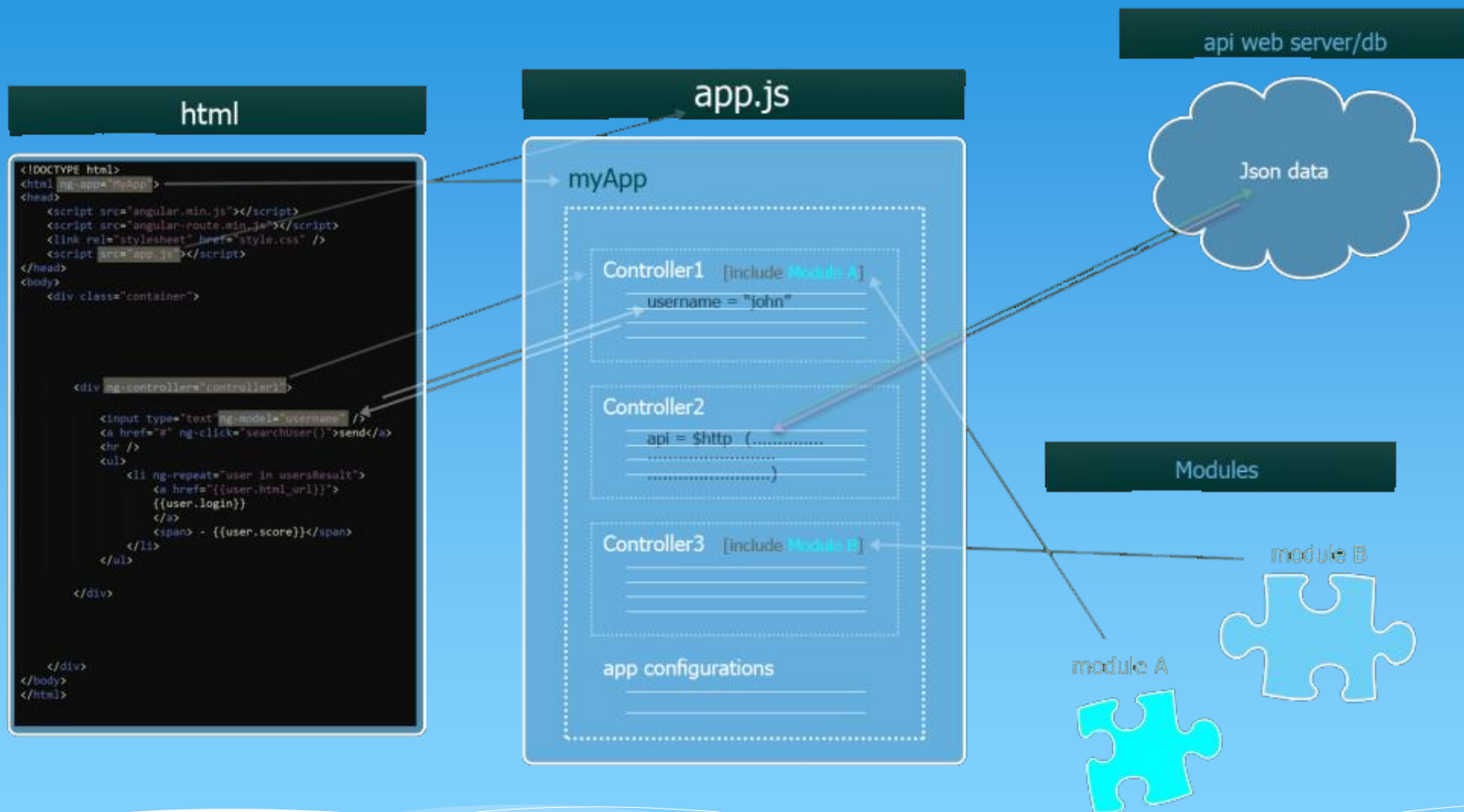
# המשך - איך זה עובד?

\* AngularJS מאכסנת את הנתונים במבנה הנקרא scope (מאחורי הקלעים scope הוא בעצם אובייקט JavaScript רגיל)

\* התצוגה (html \ view) מכילה תגיות חדשות (של AngularJS), שנקראות directives למשל (ng-controller) וכן חלקי טקסט העטופים בסוגריים מסולסלים כפולים



# איך זה עובד?





## Index.html

```
<body ng-app='myApp'>
  <div ng-controller="myController">
    <input type="text" placeholder="Enter your name" ng-model="name">
    <p> Hello {{name}} !!!</p>
    <button ng-click="greet()">click me!</ button >
  </div>
</body >
```

## app.js

```
var myApp = angular.module('myApp', []);

myApp.controller('myController', function($scope, $window){
  $scope.greet = function(){
    $window.alert("Hello" + $scope.name + " have a greet day!");
  };
});
```

# Twitter Bootstrap

\* פלטפורמה לעיצוב דפי אינטרנט ואפליקציות web בחינם

\* מבוסס על HTML5, CSS3 and JavaScript

\* תומך בכל הדפדפנים העיקריים (גם IE7)

\* Responsive Design

\* UI Bootstrap

<http://getbootstrap.com/>

<https://angular-ui.github.io/bootstrap/>

# What is Responsive Web Design?

גישת עיצוב אתרים המספקת חוויית צפייה אופטימלית על פני  
מגוון רחב של מכשירים



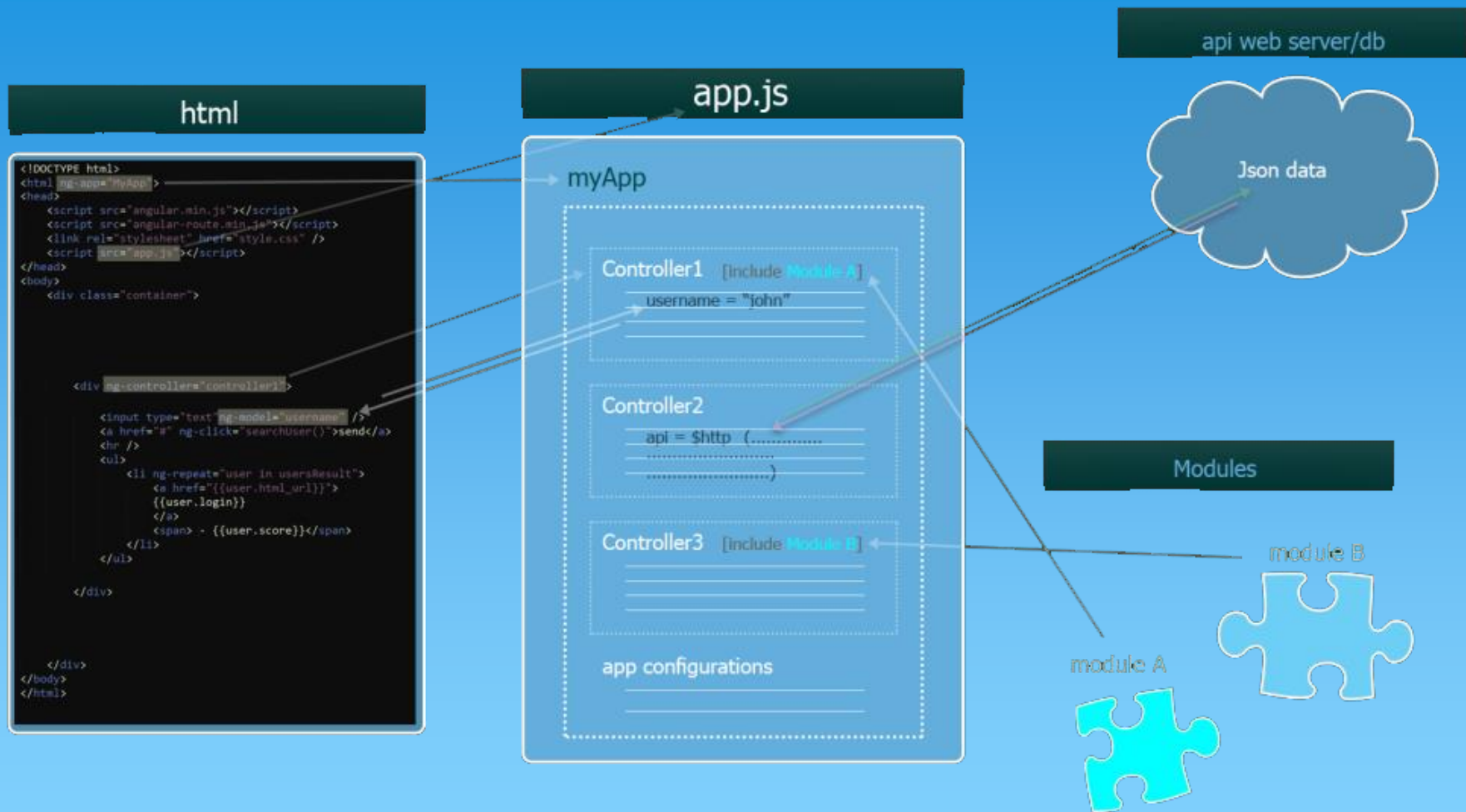
# Grid system

Bootstrap כוללת grid system בעלת 12 עמודות המתאימה  
עצמה לפי גודל תצוגת המכשיר

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

<http://getbootstrap.com/css/#grid>

# AngularJS - תזכורת



# איך זה עובד?

- \* הלוגיקה של האפליקציה נמצאת בתוך קובץ js ראשי (לרוב נקרא app.js), תחת Module מסוים
  - \* ה-Module הינו הייצוג המרכזי של האפליקציה שלנו, בו אנו מגדירים את תכונות האפליקציה ואת ההתנהגות
  - \* ה-Module הראשי יכול להיות מורכב מעוד כמה Modules שמכילים גם Controllers, Directives, Services ופונקציות (נרחיב בהמשך)
  - \* הגדרה של Module:
- ```
var mod = angular.module('myApp', []);
```
- \* ה-Controllers הוא חלק האחראי על טיפול בנתונים (Model) לצורך הצגתם (View). ה-Controllers מכיל פונקציות JavaScript

# הוראות (directives) ב-AngularJS

- \* Angular מרחיב את אוצר המילים של ה-HTML באמצעות הוראות (directives) שאותם מוסיפים בתור שדות לתגיות ה-HTML (ה- ng-\*)
- \* לדומגא ראינו את השימוש ב- directives :
- \* **ng-app** - מגדיר שקובץ ה-HTML (או חלקים ממנו) שייכים לאפליקציה האנגולרית.
- \* **ng-controller** - שמחברת את הקוד של הקונטרולר ל-view
- \* **ng-model** - הוראה שמגדירה את מקור המידע לשימוש האפליקציה.

# AngularJS Common ng Directives

\* ngApp

\* ngController

```
<html ng-app="myApp">  
  <head>  
    <title>Angular Sample Application</title>  
    <script src="angular.js"></script>  
  </head>  
  <body ng-controller="myAppCtrl">  
  </body>  
</html>
```

```
var myApp = angular.module("myApp", []);  
myApp.controller("myAppCtrl", function ($scope) {  
  
});
```



# AngularJS Common ng Directives

## \* ngRepeat

```
<body ng-controller="myAppCtrl">
  <table>
    <thead>
      <tr>
        <th>Name:</th>
        <th>Age:</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="arr in myArray">
        <td>{{ arr.name }}</td>
        <td><span ng-bind="arr.age"></span></td>
      </tr>
    </tbody>
  </table>
</body>
```

```
var myApp = angular.module("myApp", []);
myApp.controller("myAppCtrl", function ($scope) {
  $scope.myArray = [{name: 'A', age: 5},
                    {name: 'B', age: 10},
                    {name: 'C', age: 7}];
});
```

# האזנה ותגובה לאירועים ב- AngularJS

\* directives שמאזינים לאירועים. שימוש ב- directive של אירועים מאפשר לנו לכתוב קוד שמגיב לאירועים שונים. לדוגמה, קוד שעושה משהו בתגובה להקלקה על כפתור או בתגובה ליציאה משדה בטופס.

directive	למה הוא משמש?
ng-click	כשמקליקים על אלמנט
ng-dblclick	כשמקליקים הקלקה כפולה על אלמנט
ng-mouseenter	כשנכנסים עם העכבר לתוך אלמנט
ng-mouseleave	כשיוצאים עם העכבר מתוך אלמנט
ng-keypress	לחיצה על מקש במקלדת
ng-focus	כשנכנסים לשדה בטופס
ng-blur	כשיוצאים משדה בטופס

# AngularJS Common ng Directives

\* **ngShow**    \* **ngHide**    \* **ngClick**    \* **ngModel**

```
<body ng-controller="myAppCtrl">  
  <input ng-model="id">  
  <button ng-hide="ifExist(id)" ng-click="add(id)"> Add </button>  
  <button ng-show="ifExist(id)" ng-click="remove(id)"> Remove</button>  
</body>
```

```
<body ng-controller="myAppCtrl">  
  <input ng-model="id">  
  <button ng-show="!ifExist(id)" ng-click="add(id)"> Add </button>  
  <button ng-hide="ifExist(id)" ng-click="remove(id)"> Remove </button>  
</body>
```

# more ng Directives

<https://docs.angularjs.org/api/ng/directive>

\* היתרון העיקרי של directives הינו מרכיב הניתן לשימוש חוזר, ניתן להשתמש בהם בקלות ואף להוסיף תכונות שונות.

\* איך נוכל להשתמש בזה?

# Custom Directives Example

```
app.controller( "MessageCtrl", function() {  
    $scope.message = "Product created!";  
});  
  
app.directive("notification", function() {  
    return {  
        restrict: 'E',  
        scope: { message: '@' },  
        template: '<div class="alert">{{message}}</div>'  
    }  
});
```

```
<notification message="{{message}}"> </notification>
```

# AngularJS filters

\* מסנן ליבה במודול ng, מאפשר לנו לבחור תת קבוצה של פריטים ממערך כמערך חדש (תבנית אחרת)

```
<body ng-controller="myAppCtrl">
  <table>
    <thead>
      <tr>
        <th>Name:</th>
        <th>Age:</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="arr in myArray | orderBy: 'age' ">
        <td>{{ arr.name }}</td>
        <td><span ng-bind="arr.age"></span></td>
      </tr>
    </tbody>
  </table>
</body>
```

**Name: Age:**

A 5

C 7

B 10

# AngularJS filters

```
<body ng-controller="myAppCtrl">
  <table>
    <thead>
      <tr>
        <th>Name:</th>
        <th>Age:</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="arr in myArray | orderBy: '- age' ">
        <td>{{ arr.name }}</td>
        <td><span ng-bind="arr.age"></span></td>
      </tr>
    </tbody>
  </table>
</body>
```

Name:	Age:
B	10
C	7
A	5

# AngularJS filters components

<a href="#">currency</a>	Formats a number as a currency (ie \$1,234.56). When no currency symbol is provided, default symbol for current locale is used.
<a href="#">number</a>	Formats a number as text.
<a href="#">date</a>	Formats date to a string based on the requested format.
<a href="#">json</a>	Allows you to convert a JavaScript object into JSON string.
<a href="#">lowercase</a>	Converts string to lowercase.
<a href="#">uppercase</a>	Converts string to uppercase.
<a href="#">limitTo</a>	Creates a new array or string containing only a specified number of elements. The elements are taken from either the beginning or the end of the source array, string or number, as specified by the value and sign (positive or negative) of limit. If a number is used as input, it is converted to a string.
<a href="#">orderBy</a>	Orders a specified array by the expression predicate. It is ordered alphabetically for strings and numerically for numbers. Note: if you notice numbers are not being sorted as expected, make sure they are actually being saved as numbers and not strings. Array-like values (e.g. NodeLists, jQuery objects, TypedArrays, Strings, etc) are also supported.



# Angular Modules - ngRoute

\* ngRoute הינו מודול של אנגולר המאפשר ניתוב URL לאפליקציה

\* על מנת להשתמש בו נדרש להוסיף אותו לאפליקציה

\* נוסיף את הקובץ לאפליקציה

```
angular-route.js
```

\* נגדיר אותו כתלות לעבודה במודול שלנו באפליקציה

```
var myApp = angular.module("myApp", ["ngRoute"]);
```

# ngRoute - Module Components

\* Directive :

\* ngView - הוא directive המשלים את השירות \$route ע"י הוספת התבנית לדף HTML הראשי, כל פעם שיש שינוי בנתיב הנוכחי, התצוגה תשתנה בהתאם לתצורה של השירות \$route

```
<ng-view></ng-view> OR <div ng-view=""></div>
```

\* Provider

\* \$routeProvider - משמש להגדרת הנתיבים

\* Service

\* \$route – משמש כמקשר בין הבקרים לתצוגות עליהם הם אחראים

\* \$routeParams – מאפשר לאחזר את הסט הפרמטרים הנוכחי של הניתוב

# איך זה עובד?

```
var app = angular.module( "githubViewer", ["ngRoute"] );
app.config( function($routeProvider) {
  $routeProvider
    .when( "/main", {
      templateUrl: "main.html",
      controller: "MainController"
    })
    .when( "/user/:username", {
      templateUrl: "user.html",
      controller: "UserController"
    })
    .when("/repo/:username/:reponame", {
      templateUrl: "repo.html",
      controller: "RepoController"
    })
    .otherwise( { redirectTo: "/main" } );
});
```

<https://github.com/jce-il/github-viewer>

<http://hello-todo.azurewebsites.net/#/main>

# AngularJS Services

- \* מארגן את הלוגיקה
- \* מעביר את הנתונים הלאה מה- controllers
- \* משתף את הנתונים בין ה- controllers
- \* built-in services תמיד יתחילו בסימן \$
- \* על מנת להשתמש בו, נוסיף כאותו כתלות לרכיב (controller, service) (filter, directive)

```
myApp.controller('myController', [ '$http', function( $http ) {  
    ....  
}]);
```

# AngularJS Services

\* כיצד נפנה לשרת?

\* \$http – שירות להעברת בקשות לשרת, מחזירה callback עם הצלחה/שגיאה

\* סינטקס:

```
$http ( { method: 'GET', url: '<link to server>' } )  
  .success( function (data) {  
    ....  
  })  
  .error( function(data) {  
    ....  
  });
```

```
$http.get( '/path/to/resource.json', {pram: 'val' } );
```

# AngularJS Creating Services

\* ניתן ליצור service משלנו, באמצעות פונקציית factory

\* סינטקס:

```
myApp.factory('myService', function() {  
    ....  
});
```

\* ועוד...

<https://docs.angularjs.org/guide/services>