

# Angular – Part Two

The Fun Part Begin!

# סיכום השיעור הקודם

אז מה למדנו עד כה?

- Module
- Directive
- Two-Way Data Binding
- קצת משחקים HTML נחמדים

# Ng-model

מחברים בין רכיבים שונים על המסך  
החיבור עובד לרכיבים כמו בחירה נגללת, וקלטים אחרים

מקבל את הערך אליו הוא מתחבר למשל `ng-bind="review.stars"`  
מה שזה עושה בתכלס, זה מעדכן את הערך הזה בהתאם לשינוי,  
ואז אם יש רכיב במסך שמראה את הערך, הוא יעודכן מיידית  
(Two way data binding זוכרים?)

כמובן, שכמו מקודם, הערך המתקבל צריך להגיע מתוך Controller.

# ng-submit

אי פעם שאלתם את עצמכם "איך אני גורם לנתונים להתעדכן שסיימו למלא את הטופס?"  
אז הנה התשובה!

פשוט שולחים לng-submit פונקציה בקונטרולר, שעושה מה שאנחנו רוצים שיקרה, זהו!

# Form Validation

אם רוצים, ניתן לבצע וידוא לטפסים (וידוא שאימייל תקין, שכל הפרטים מולאו וכו..)

כל מה שצריך:

- נוסף לקלטים novalidate כדי לבטל את האימות של HTML עצמו
- נסמן ב required כל קלט שהוא חובה
- נוסף ל ng-submit בדיקה ל \$valid
- \$valid הוא משתנה אוטומטי של אנגולר שמסמן שהכל תקין

# Custom Directive

ניתן ליצור Elements, Attributes משלנו, שיטענו HTML מקובץ נפרד.  
דבר זה נועד למטרות הבאות:

1. אם יש קוד HTML שחוזר על עצמו הרבה (הרשמה, ...NavBar)
2. אם נרצה להפוך את הHTML לדינאמי יותר וכמובן להתנהגותי יותר.

איך עושים את זה?

1. נעביר את חלק הHTML שלנו לקובץ HTML חדש
2. ניצור Directive משלנו שיחזיר לנו את הHTML

```
app.directive('DirectiveName', function(){  
  Return {  
    ...  
  };  
});
```

# המשך

כאמור, אנו יכולים ליצור element או attribute מתי נחליט מה צריך?

- Attribute – אם אנחנו הולכים לשלב את זה עם עוד דברים, זה לא תמיד יופיע בדיוק אותו דבר
- Element – אם זה משהו חד משמעי, שתמיד יופיע באותו פורמט (גלריה, לוגין...)

בואו נזכר מה זה כל אחד בפועל...

```
<element attribute></element>
```

```
<div class="..." style="..." id="..."></div>
```

# המשך

בואו נסתכל על הקוד שוב...

```
app.directive('DirectiveName', function(){  
  return {  
    restrict: '', → 'E' for Element, 'A' for Attribute  
    templateUrl: '', → The HTML name  
    controller: '', → Controller to Attach (next slide)  
    controllerAs: '' → Controller Alias  
  };  
});
```



# המשך (שוב)

```
app.directive('DirectiveName', function(){  
  return {  
    ...  
    controller: '', → Controller to Attach (next slide)  
    controllerAs: '' → Controller Alias  
  };  
});
```

אז למה שנוסיף קונטרולר?  
פשוט, כדי שנוכל לכתוב בו את הפונקציות והנתונים של אותו רכיב במסך שהרגע יצרנו

# תלויות

הדרך לצמצם את הקוד ולסדרו לפי תפקידים.

זוכרים את הסוגריים המרובעים האלו? → `var app = angular.module('ModuleName', []);`

תפקידם להודיע למודול באילו מודולים של אנגולר נוספים הוא תלוי

את השמות הוא מקבל **כסטרינג**

**השמות שהוא מקבל, הם שמות המודולים**

# Services – Go Net Go!

אם נזכר בקורס תכנות בסביבת אינטרנט, כאשר רצינו לעשות קריאה לשרת בשביל נתונים השתמשנו בAJAX כמובן שגם כאן אפשר, אבל אנחנו נראה איך אנגולר עושה זאת.

- \$http – המקביל של AJAX עושה כל מה שAJAX עושה

סינטקס –

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