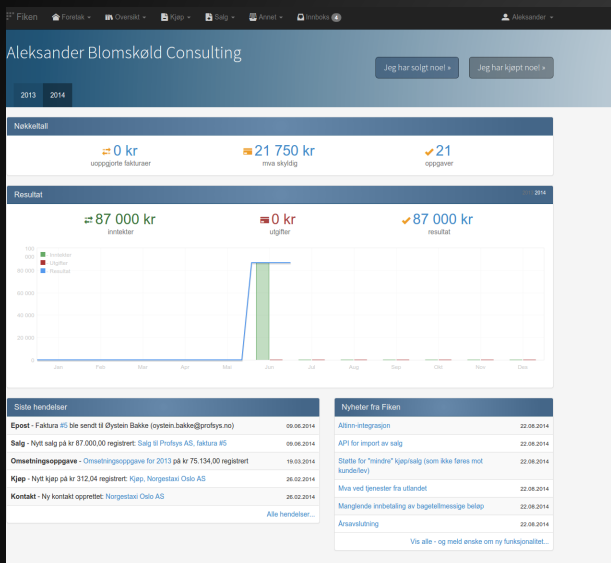


# AngularJS Crash Course

# Agenda

- Velkommen
- Om oss
- Hvorfor AngularJS
- Enkel bruk/databinding
- Filter
- Deklarative views
- Digest
- DI, services
- Directives
- \$http
- ng-route

# Om oss - Fiken.no



## Ny faktura

Konto, kunde og produkt: Velg eksisterende fra nedtrekkslistene eller opprett nye.

Opprett faktura bokfører fakturaen, og neste skjerm bilde gir mulighet til å sende den på epost eller skrive ut. Innbetaling registreres enten manuelt eller via import fra nettbank (CSV-fil eller OCR/kid-nummer).

Alle fakturaer finnes igjen under [salgsversikten](#).

Fakturadato: 04.09.2014

Dager til forfall: 21 Forfallsdato: 25.09.2014

Beskrivelse:

Kontonummer: Skandiabanken (ABC) (98011155812)

Fakturalinje

	Pris	Ant
<input type="text" value="Velg produkt..."/>		
<input type="text" value="Velg produkt..."/>		
<input checked="" type="radio"/> Ny produkt-linje eller <input checked="" type="radio"/> Ny fritræk-linje		

Lagre utkast Slett utkast Til fakturoversikten

Opprett faktura

Velg kontakt

javab

Fra bregg (20)

Javabin Økonomi AS Postboks 5163 Majorstuen, OSLO

Java Brukerforening i Norge c/o javaBin Økonomi Postboks 5163, Majorstuen, OSLO

Java Consulting Arvid Hansen Postboks 414 Økern, OSLO

Java Eid DA c/o May Iren Gill Postboks 61, BRYGGJA

Java Espressoar & Kaffeforretning AS Ullevålsveien 47, OSLO

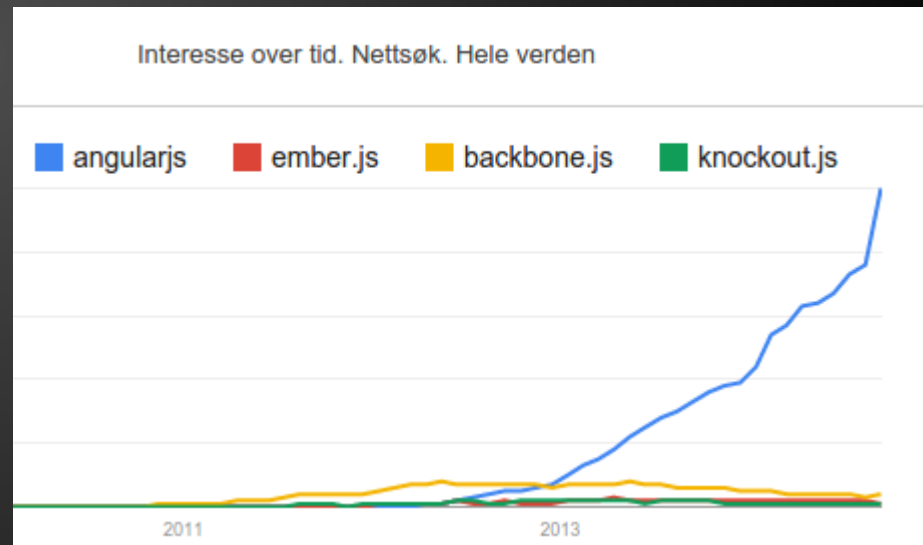
Java Holding AS Tullingsgate 6, OSLO

Fortige Side 1 av 4 Neste

Opprett "javab"

# Hvorfor ANGULARJS?

- Testbart
- MVC - deklarativ view
  - HTML for CRUD-applikasjoner
- Non-intrusiv
  - POJO



# INTRO

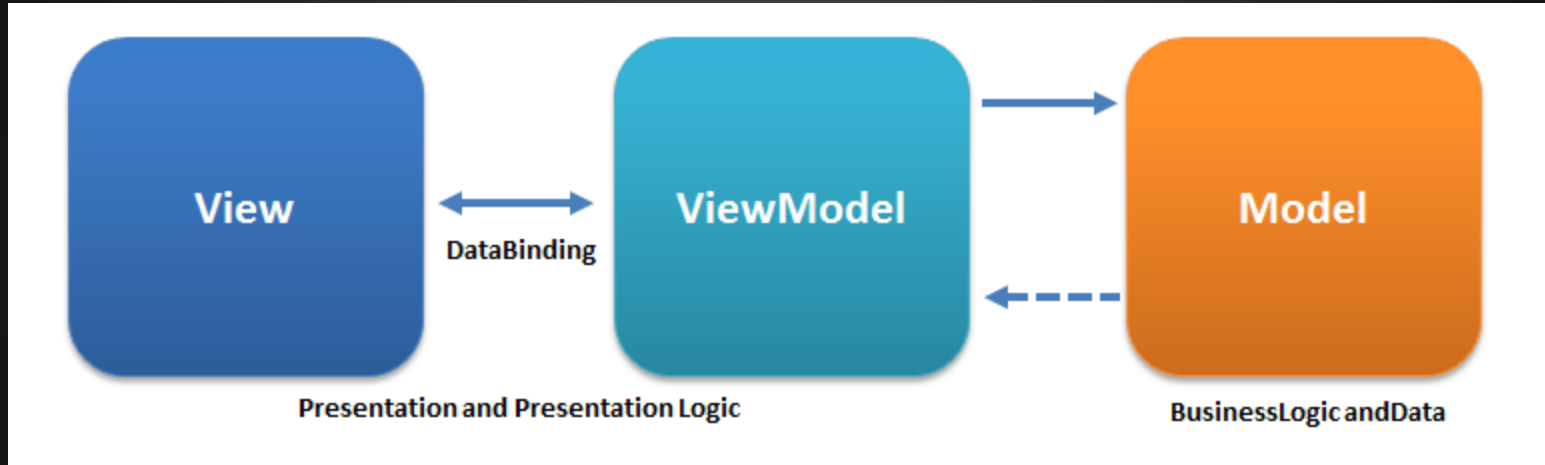
```
<html>
<head>
  <script type="application/javascript" src="angular.js"></script>
</head>
<body ng-app>

  <h1>Hello: {{myText}}</h1>

  <input type="text" ng-model="myText"/>

</body>
</html>
```

demo



MVVM: Model ViewModel Model



```
function myCtrl() {  
    this.name = 'JavaZone 2014';  
}
```

```
<html>
<head>
  <script type="application/javascript" src="angular.js"></script>
</head>
<body ng-app>
<div ng-controller="myCtrl as ctrl">
  <p>Hello {{ctrl.name}}!</p>
</div>
</body>
</html>
```

```
var ctrl = new myCtrl();
```

```
function myCtrl() {  
  this.names = [  
    'Arne',  
    'Bjarte',  
    'Conrad'  
  ];  
}
```

```
<html>
<head>
  <script type="application/javascript" src="angular.js"></script>
</head>
<body ng-app>
<div ng-controller="myCtrl as ctrl">
  <ul>
    <li ng-repeat="name in ctrl.names">{{name}}</li>
  </ul>
</div>
</body>
</html>
```

Old: "attach to \$scope":

```
function myCtrl($scope) {  
  $scope.names = [ .. ];  
}
```

```
<div ng-controller="myCtrl">  
  <ul>  
    <li ng-repeat="name in names">  
      {{name}}  
    </li>  
  </ul>  
</div>
```

New: "controller as":

```
function myCtrl() {  
  this.name = [ .. ];  
}
```

```
<div ng-controller="myCtrl as ctrl">  
  <ul>  
    <li ng-repeat="name in ctrl.names">  
      {{name}}  
    </li>  
  </ul>  
</div>
```

# Event

```
<div ng-controller="myCtrl as ctrl">
  <ul>
    <li ng-repeat="name in ctrl.names">{{name}}</li>
  </ul>

  <form>
    <input type="text" ng-model="ctrl.newName">

    <button ng-click="ctrl.add()">Add</button>
  </form>
</div>
```

```
function myCtrl() {  
  var ctrl = this;  
  
  ctrl.names = [ .. ];  
  
  ctrl.newName = '';  
  ctrl.add = function() {  
    ctrl.names.push(ctrl.newName);  
  }  
}
```



# Global Scope

```
<script ..>  
function myCtrl() {  
    ...  
}  
</script>
```

```
<script ..>  
(function() {  
    function myCtrl() {  
        ...  
    }  
})();  
</script>
```

Error: ng:areq

Bad Argument: Argument 'MyCtrl' is not a  
function, got undefined

```
(function () {  
  function myCtrl() {  
    this.names = [  
      'Arne',  
      'Bjarte',  
      'Contrad'  
    ];  
  }  
}
```

```
angular.module('MyApp', []).  
  controller('myCtrl', myCtrl);  
})();
```

```
<html ng-app="MyApp">  
<head> .. </head>  
<body>  
  <div ng-controller="myCtrl">  
    ..  
  </div>  
</body>  
</html>
```

# Oppgavetid

Oppgaver:

- [github.com/fiken/javazone-2014](https://github.com/fiken/javazone-2014)



# **FILTERS**

**{{expressions}}**



## Javascript-ish:

- {{navn}}
- {{bruker.adresse.postnummer}}
- {{“En og en er: “ + (1+1)}}
- {{brukere.length}}
- {{brukere[i].navn.substring(2)}}

- ... men med noen forskjeller
  - Context er scope, ikke window
  - Forgiving, ingen Reference eller Type-errors
  - Ingen flow control statements
- Filters!

- {{navn | uppercase}}
- {{brukere | filter: search}}
- {{brukere | orderBy: "navn" : true }}
- {{total | currency }}

Syntax: {{ data | filter : 'evt arg1' : 'evt argn' }} ->  
filter(data, 'evt arg1', 'evt argn')

Chaining: {{ liste | orderBy : 'name' : true | limitTo : 5 }} ->  
limitTo( orderBy(liste, 'name', true), 5)

```
module.filter('sqrt', function(){  
  return function(num){  
    return Math.sqrt(num);  
  }  
});
```

# Obs!

- Pure - gjentatte kall med lik input skal gi likt resultat

```
module.filter('random', function(){  
  return function(data){  
    return Math.random();  
  }  
});
```



# Deklarative view

```
<div ng-repeat="name in names">  
  Rad: {{$index}}.  
  Første rad? {{$first}}, eller siste? {{$last}}, ingen av delene? {{$middle}}.  
  Partall-rad? {{$even}} eller oddetall-rad {{$odd}}  
</div>
```



```
<div ng-show="customer.vip">
  <!-- Blir skjult med display:none hvis ikke customer.vip -->
  Rabattkode: {{discountCode}}
</div>
```

```
<div ng-hide="!customer.vip"> <!-- Nøyaktig samme som over -->
  Rabattkode: {{discountCode}}
</div>
```

```
<div ng-if="customer.vip">
  <!-- Blir ikke med i DOM-en med mindre customer.vip -->
  Rabattkode: {{discountCode}}
</div>
```

```
<div ng-class="{ 'large-text': customer.age > 70 }">
  Velkommen
</div>
```



# Hei Angular

# Dirty checking

# \$digest()

- Sjekker for eventuelle endringer i modellen
  - Må gjøres felt-for-felt
- Dersom en endring oppdages kalles watchers
- Dette kan igjen føre til nye endringer
- Går fint så lenge sjekkene ikke tar for lang tid eller antall elementer blir for mange
- ES7: Object.observe() - Jedi power!
- Som regel trenger man ikke å tenke på \$digest
  - ... med mindre man vil gjøre ting utenfor Angular-verden

# \$apply()

\$apply() setter i gang en \$digest() i \$rootScope som propaganderes til alle aktive \$scopes

```
var scope = angular.element(domElement).scope();
```

```
scope.$apply(function () {  
    //gjør endringer på scope  
});
```

# \$scope.\$watch()

Få beskjed av Angular når en property i \$scope har endret seg

```
$scope.$watch('discount',  
  function(newVal, oldVal) {  
    $scope.total = $scope.sum*newVal;  
  });
```





D.I.

```
function myController($scope, $http) {  
}
```

```
module.filter('doSomething', function(myService){  
    return function(num){  
        return myService.foo(num);  
    }  
});
```

```
//eller - minifyervennlig:
```

```
module.filter('doSomething', ['myService', function(m) { ... }]);
```

# Moduler

```
// ny modul
angular.module('jzConstants', []);

// hent opp en modul tidligere definert:
var mod = angular.module('jzConstants');
// registrere en konstant:
mod.constant('pi', 3.141592654);

// registrere en ny modul med et filter med dependency
angular.module('jzServices', ['jzConstants'])
    .filter('circleArea', function(pi) {
        return function(r) {
            return pi * r * r;
        }
    });
```

```
// servicer med factory
angular.module('jzServices', [])
    .factory('math', function() {
        return {
            increment:
                function(val) {
                    return val + 1
                }
        }
    });

//brukes feks i en controller
angular.module('jzCtrl', ['jzServices'])
    .controller('fubarCtrl', function(math) {
        //...
        var foo = math.increment(bar);
    });
```



**\$HTTP**

**\$http**



```
$http.get('/someUrl')  
$http.post('/someUrl', data)  
$http.put('/someUrl', data)  
$http.delete('/someUrl')  
$http.patch('/someUrl', data)  
  
$http.jsonp('/someUrl')
```

```
var config = {  
  method: 'GET',  
  url: '/someUrl',  
  params: { .. },  
  headers: { .. },  
  cache: ..,  
  timeout: ..,  
  ..  
}:  
var promise = $http(config)
```

```
var req = $http({method: 'GET', url: '/someUrl'})

req.
  success(function(data, status, headers, config) {
    ..
  }).
  error(function(data, status, headers, config) {
    ..
  });

req.
  then(function(response) {
    ..
  },
  function(response) {
    ..
  });
```



# DIRECTIVES

- Utvidelse av HTML
- ng-app, ng-repeat, ng-click, ng-model etc...
- Attributes eller elements (class eller comment)
  - Utvidelse av funksjonalitet
  - Komponenter
- Grunnlag for kommende WebComponents (custom elements, shadow dom, html imports og templates)

```
module.directive('datePicker', datePickerDirective)
```

```
function datePickerDirective(){
```

```
  return {
```

```
    restrict: 'E',
```

```
    scope: { date: '=model' },
```

```
    template: '<input ng-model="date"><button ng-click="showDatePicker()"><i class="calenar-icon"/></button>',
```

```
    controller: function($scope){
```

```
      $scope.showDatePicker = function(){ ... }
```

```
    } } }
```

```
<date-picker model="user.date"></date-picker>
```

```
module.directive('trimOnBlur', trimOnBlurDirective)
```

```
function trimOnBlurDirective(){
```

```
  return {
```

```
    controller: function($scope, $element){
```

```
      $element.bind('blur', function(){
```

```
        $element.val() = $element.val().trim();
```

```
      });
```

```
    }
```

```
  } }
```

```
<input type="text" trim-on-blur />
```

- Isolasjon
- Templates
- DOM-interaksjon og manipulering



# Scopes

- Alle scopes har en \$parent (bortsett fra \$rootScope)

## Directive config

- scope: false (default)
- scope: true
  - Nytt barnescope hvor Object.getPrototypeOf(scope) === scope.\$parent
- scope: { }
  - Nytt isolert scope hvor Object.getPrototypeOf(scope) !== scope.\$parent
  - Object.getPrototypeOf(scope) === Object.getPrototypeOf(scope.\$root)

- Mange directives på samme element
- Forventet oppførsel i view
- Ytelse (kompilering)

Enkle ting er enkelt, vanskelige ting er mulig :-)

```
.directive('directiveName', function factory(injectables) {  
    var directiveDefinitionObject = {  
        priority: 0,  
        template: '<div></div>', // or // function(tElement, tAttrs) { ... },  
        // or templateUrl: 'directive.html', // or // function(tElement, tAttrs) { ... },  
        transclude: false,  
        restrict: 'A',  
        scope: false,  
        controller: function($scope, $element, $attrs, $transclude, otherInjectables) { ... },  
        controllerAs: 'stringAlias',  
        require: 'siblingDirectiveName', // or // ['^parentDirectiveName', '?optionalDirectiveName', '?^optionalParent'],  
        compile: function compile(tElement, tAttrs, transclude) {  
            return {  
                pre: function preLink(scope, iElement, iAttrs, controller) { ... },  
                post: function postLink(scope, iElement, iAttrs, controller) { ... }  
            } // or return function postLink( ... ) { ... }  
        },  
        // or link: {  
        //   pre: function preLink(scope, iElement, iAttrs, controller) { ... },  
        //   post: function postLink(scope, iElement, iAttrs, controller) { ... }  
        // } // or link: function postLink( ... ) { ... }  
    };  
    return directiveDefinitionObject;  
});
```

\$parse - service som gjør et Angular expression til en funksjon

```
var getUsernameFn = $parse('user.name');  
var userName = getUsernameFn($scope);
```

# jz-reset

```
angular.module('MyJzApp', [])  
  .directive('jzReset', jzResetDirective);  
  
function jzResetDirective($parse) {  
  return {  
    controller: function ($element, $scope, $attrs) {  
      var fn = $parse($attrs.jzReset);  
      $element.bind('reset', function (event) {  
        $scope.$apply(function () {  
          fn($scope);  
        });  
      })  
    }  
  }  
}
```

# jz-reset

```
<div class="container" ng-init="status = 'pristine'">
  <form jz-reset="status = 'reset'">
    <label>
      Name: <input class="form-control" type="text" ng-model="name">
    </label>

    <button type="reset">Reset</button>
  </form>

  <p>Status: {{ status }}</p>
</div>
```

# jzLabel

```
angular.module('MyJzApp', [])  
  .directive('jzLabel', jzLabelDirective);  
  
function jzLabelDirective() {  
  return {  
    restrict: 'E',  
    scope: {jzText: '='},  
    template: '<span class="label label-default">{{jzText}}</span>'  
  }  
}
```

Status: <jz-label jz-text="status"></jz-label>





# NG-ROUTE

todo: husk å nevne templates

# **SPA - Single Page Application**

ngRoute

<https://docs.angularjs.org/api/ngRoute>

.../index.html#/hash/path

```
<html>
<head>
  <script src="angular.js"></script>
  <script src="angular-route.js"></script>
  <script>
    (function () {
      var module = angular.module('MyApp', ['ngRoute']);
      ...
    })();
  </script>
</head>
<body ng-app="MyApp">
<div ng-view></div>
</body>
</html>
```

```
module.  
  config(function($routeProvider) {  
    $routeProvider.  
      when(path, config).  
      when(path, config).  
      otherwise(config);  
  });
```

```
when(path, config)
```

```
'/'
```

```
'/foo'
```

```
'/foo/:fooId'
```

```
'/foo/:fooId/edit'
```

```
'/foo/:fooId*/edit'
```

```
'/foo/:fooId/:mode?'
```

```
when(path, config)
```

```
{  
  controller: string|function,  
  controllerAs: string|function,  
  template: string|function,  
  templateUrl: string|function,  
  resolve: object,  
  ...  
}
```

```
$routeProvider.  
  when('/', {  
    templateUrl: 'list.html',  
    controller: 'listCtrl as ctrl'  
  }).  
  when('/:personId/:mode?', {  
    ..  
  });
```



front-page.html:

```
<div>
  Hello {{ctrl.name}}!
</div>
```

index.html:

```
<html>
..
  <script
    type="text/ng-template"
    id="front-page.html">
    <div>
      Hello {{ctrl.name}}!
    </div>
  </script>
..
</html>
```

```
$routeProvider.  
  when('/:personId/:mode?', {  
    controller: 'myCtrl as ctrl'  
  });
```

```
function myCtrl($routeParams) {  
  var personId = $routeParams.personId;  
}
```

```
<a ng-href='#/{{person.id}}'>Details</a>  
<a ng-href='#/{{person.id}}/edit'>Edit</a>
```

```
function myCtrl($location) {  
  var ctrl = this;  
  
  ctrl.view = function() {  
    $location.path('/' + id);  
  }  
}
```

# Resolve

Preventing rendering of “empty” pages

```
function MyCtrl(PersonService) {  
    var ctrl = this;  
  
    PersonService.list().then(function(res) {  
        ctrl.persons = res.data;  
    });  
}
```

```
module.  
  config(function ($routeProvider) {  
    $routeProvider.  
      when('/', {  
        ..  
        resolve: {  
          persons: function(PersonService) {  
            return PersonService.list();  
          }  
        }  
      });  
    });  
    function myCtrl(PersonService, persons) {  
      var ctrl = this;  
  
      ctrl.persons = persons.data;  
    }  
  });
```

```
module.  
  config(function ($routeProvider) {  
    $routeProvider.  
      when('/:personId', {  
        controller: 'myCtrl as ctrl',  
        resolve: {  
          person: function(PersonService, $route) {  
            return PersonService.get($route.current.params.personId);  
          }  
        }  
      });  
    });  
    function myCtrl(PersonService, person) {  
      var ctrl = this;  
  
      ctrl.person = person.data;  
    }  
  });
```

