

software pilots

TRIFORK.



ANGULARJS

by Google™

Features i

◦ helikopterperspektiv

- 2-vejs-databinding mellem view og model.
- “Plain JavaScript”.
- Dependency Injection.
- Filtre til GUI-relaterede opgaver.
- Simpel REST-klient.
- Egne tags og attributter i HTML.
- Notifikationer.
- Testbart.

Databinding

- Eksempel med en controller med en property der synkroniseres med viewet.
- Bemærk: “Plain Old JavaScript”.
- ng-app, ng-controller, {{...}}, \$scope...

Filtre

- Et meget simpelt koncept.
- Nemt at lave sit eget filter.
- Eksempel med liste af brugere, hvor der filtreres på navn. Og måske sorteres? Det er alt sammen nemt!

...og med REST-kald!

Vi skal bare lige inkludere angular-resource, og vi skal strukturere vores applikation lidt, så kan vi via DI få en fin REST-klient.

```
app.factory('User', function($resource) {  
    return $resource('/api/users/:id');  
})
```

Men... hvad med flere sider?

- Routes binder “sider” til controllers og “fragments”.
- `<ng-view>` angiver hvor den givne “fragment” skal indgå.
- Som udgangspunkt bruges “#/page1” osv. til “siderne”. Mere herom senere...

Routes

```
app.config(function($routeProvider) {
  $routeProvider.
  when('/users', {
    templateUrl: '/app/partials/users.html',
    controller: UsersCtrl
  }).
  when('/groups', {
    templateUrl: '/app/partials/groups.html',
    controller: GroupsCtrl
  }).
  otherwise({
    redirectTo: '/users'
  });
});
```

Men hele tiden nye side-hentninger?

Så slår vi (dam-dam-dammm!) HTML5-mode til!

```
$locationProvider.html5Mode(true);
```

...og så kører alt client-side.

Kræver dog en server der kan klare det!

...og et base-tag: `<base href="/">`

Vi kan så fjerne #'erne fra links'ene.

Så til det sjove... tags og attributter!

- Nogle gange er HTML altså bare lidt bøvlet at arbejde med.
- Man gentager sig selv, og det hele er meget lidt dynamisk.
- Så er det godt at Angular lader dig definere din egen “HTML-DSL”.

Simpelt eksempel: Hello World

```
app.directive('helloWorld', function() {  
  return {  
    template: '<h1>Hello world!</h1>',  
    restrict: 'E'  
  };  
});
```

Lidt mere komplekst: Navigationsbar

```
app.directive('navbarItem', function($location) {
  return {
    template: '<li><a ng-href="{{route}}">{{name}}</a></li>',
    restrict: 'E',
    replace: true,
    scope: { route:'@', name:'@' },
    link: function(scope, element, attributes, controller) {
      scope.$on('$routeChangeSuccess', function() {
        var path = $location.path();
        if (path === scope.route) {
          element.addClass('active');
        } else {
          element.removeClass('active');
        }
      });
    }
  };
});
```

Organisering af applikation

- app.js: Konfiguration af hele app'en (routes, HTML5-mode, ...).
- controllers.js
- services.js
- directives.js (eller i app.js hvis der er få)
- filters.js (eller i app.js hvis der er få)

jQuery widgets

- AngularJS kan (som regel) godt arbejde sammen med andre JavaScript widgets.
- Bedste måde er at wrappe i et direktiv
- datepicker eksempel

```
module.directive('datepicker', function() {  
    return {  
        link: function(scope, element, attributes) {  
            $(element).datepicker();  
        }  
    };  
});
```

jQuery widgets

Ikke alt integrerer naturligt med Angular.

Angular forventer en stabil DOM - de DOM-elementer som Angular linker kan ikke bare slettes udenom Angular.

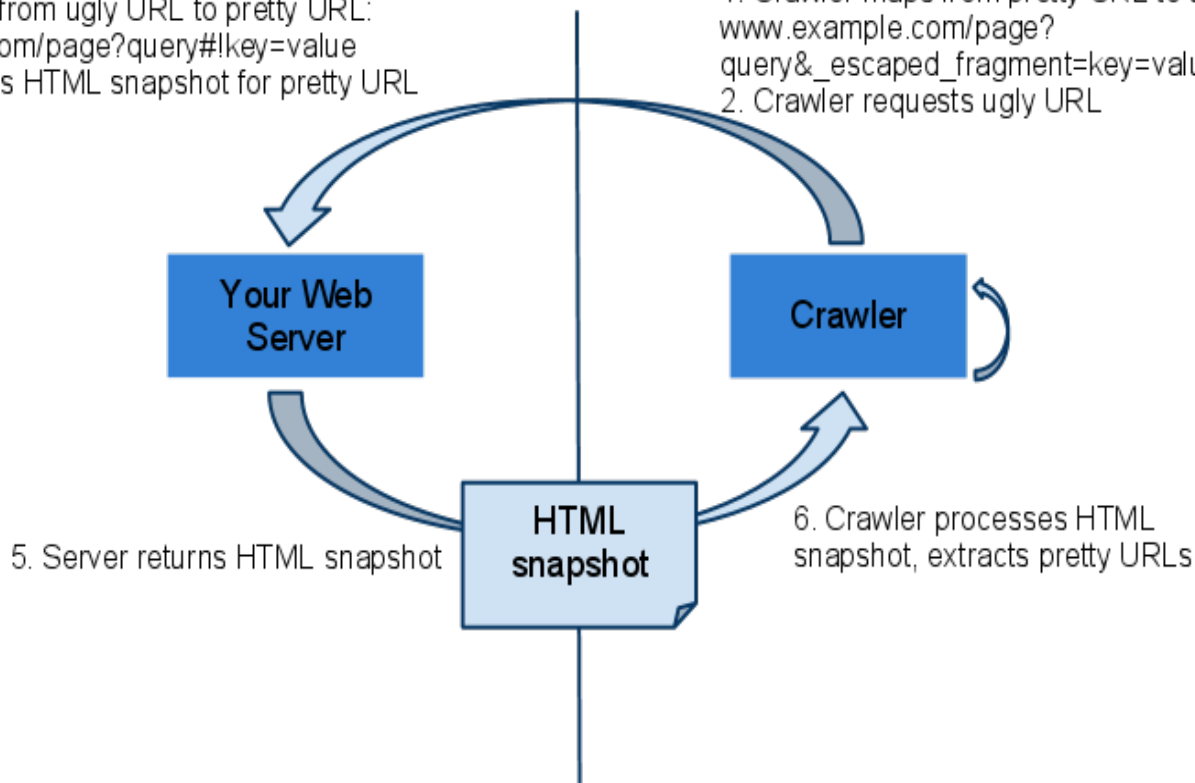
F.eks. vil man nok helst skrive angular-specifik kode til at åbne en modal dialog.

Søgemaskineoptimering

Indeksering af JavaScript-enablede sider kræver særlige tricks.

3. Server maps from ugly URL to pretty URL:
`www.example.com/page?query#!key=value`
4. Server creates HTML snapshot for pretty URL

1. Crawler maps from pretty URL to ugly URL:
`www.example.com/page?query&_escaped_fragment=key=value`
2. Crawler requests ugly URL



<https://developers.google.com/webmasters/ajax-crawling/docs/getting-started>

Søgemaskineoptimering

- Generering af HTML-snapshots kan give noget fnidder, men der er værktøjer til formålet:
- HtmlUnit (Java) (Anbefales IKKE)
- Zombie.js (Node, JavaScript - ikke afprøvet)
- Selenium + headless firefox eller webkit
- PhantomJS (Webkit)

Søgemaskineoptimering

Søgemaskiner vil gerne have en HTTP status kode **404 (Not Found)** på URLs som ikke (længere) findes - og en **301 (Moved Permanently)** på URLer som er flyttet.

Det kræver server-side logik!

Søgemaskineoptimering

SEO-huskeliste:

- Semantisk markup
- <title>, <meta> tags kan også fint sættes med JavaScript
- Links mellem "sider" som hrefs, ikke kun clickhandlers
- Gode URL'er
- Indhold som ikke har en URL eller som ikke er vist når siden er indlæst kan ikke indekseres!

Søgemaskineoptimering

Udfordringer:

- Får AJAX-sider lige så god page rank som "almindelige" sider? (tilsyneladende JA)
- Andre søgemaskiner? (Bing ser ikke ud til at virke ordentligt)
- SEO værktøjer understøtter som regel ikke AJAX crawling.
- Ved HTML5-mode får IE-brugere og andre ikke den samme URL - kan måske løses med angivelse af [Canonical URL](#) i headeren.

Tool Support

- Chrome plugin (Batarang) - findes i Chrome web store
- IDE plugins: IntelliJ, Netbeans

Referencer

- Officiel hjemmeside:
 - <http://angularjs.org>
- Kode fra i dag:
 - <https://github.com/andershessellund/angular-seed>
- SimplePVR:
 - <https://github.com/olefriis/simplepvr>
- Ajax Crawling:
 - <https://developers.google.com/webmasters/ajax-crawling/docs/getting-started>
- Datepicker:
 - <https://github.com/eternicode/bootstrap-datepicker>
- Sammenlign JavaScript MVC frameworks:
 - <https://github.com/addyosmani/todomvc>

Tak!

- Anders Hessellund Jensen
 - ahj@trifork.com
- Ole Friis Østergaard
 - of@silverbullet.dk