



AngularJS 필터(Filter)기능 및 Data 바인딩 개념

Table of Contents

- AngularJS 필터(Filter)기능 및 Data 바인딩 개념
 - Filter 및 Data 바인딩 전체 완성 소스
 - Controller.js 소스
 - Filter의 개념 및 적용
 - Data 바인딩 개념 및 적용
 - 실제 소스 동작 화면
 - plunker를 위한 실시간 테스트 방법
 - 최종 요약 정리

Filter 및 Data 바인딩 전체 완성 소스

우선 아래와 같은 간단한 예제를 작성해봅니다.

```
1 <!DOCTYPE html>
2 <html lang="ko" ng-app>
3 <head>
4   <meta charset="utf-8">
5   <title>AngularJS Filter And 데이터 바인딩</title>
6   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.10/angular.min.js"></script>
7   <script src="controller.js"></script>
8 </head>
9 <body ng-controller="StudyMemberListCtrl">
10
11   <input type="text" ng-model="searchContent"/>
12
13   <!-- 당연한 이야기지만, 위의 ng-model 과 아래 filter 의 Model 이름은 같아야 필터링 기능이 정상적으로 동작된다. -->
14
15   <ul>
16     <!-- filter 지시어를 유심히 살펴보자. -->
17
18     <li ng-repeat="member in members | filter:searchContent">
19
20       <!-- style css 적용해보자. -->
21
22       {{member.name}}
23       <p>{{member.message}}</p>
24     </li>
25   </ul>
26 </body>
27 </html>
```



지난 번 예제와 비교하여 추가된 부분이 어떤 곳인지 한 번 생각해 보자.

Controller.js 소스

자, 다음 Controller.js 파일을 아래와 같이 작성해보자. 서로 다른 키워드로 넉넉하게 입력해 두면 더욱 좋다.

```
1 function StudyMemberListCtrl($scope){
2   $scope.members = [
3     {"name":"이남희","message":"I love Groov"}.
```

```

4      {"name":"한용희","message":"I love Apple"},
5      {"name":"이준호","message":"I love Ipad"},
6      {"name":"윤정부","message":"I love Jobs"},
7      {"name":"김선준","message":"I love AirPlane"},
8      {"name":"김현하","message":"I love Computer"},
9      {"name":"손지성","message":"I love Money"},
10     {"name":"백성진","message":"I love Car"},
11     {"name":"김은기","message":"I love AngularJs"}
12 ];
13 }

```



저번 시간에도 이야기 했지만, Controller.js 파일의 내용은 굳이 **외부로 링크** 시키지 않고, 내부 **인라인**으로 작성해도 작동에 전혀 문제가 없음을 다시 한 번 알려드립니다.

Filter의 개념 및 적용

AngularJs 필터 함수 : input 박스와 연계된 기능이며, 입력된 값과 매치되는 것으로 새로운 배열을 생성하여 컨트롤러에 작성된 모델을 기준으로 새로운 배열을 생성하게 해주는 기능이다. 이 모든 기능은 개발자에게 투명하게 제공된다.

```
<li ng-repeat="member in members | filter:searchContent">
```

위 예제에서 필터 사용을 명시하는 부분은 빨간색 박스로 표시해 두었으며, 필터를 적용함으로써, 생성되는 모델에 관한 이름을 명시해준다.(임의로 지정가능)

Data 바인딩 개념 및 적용

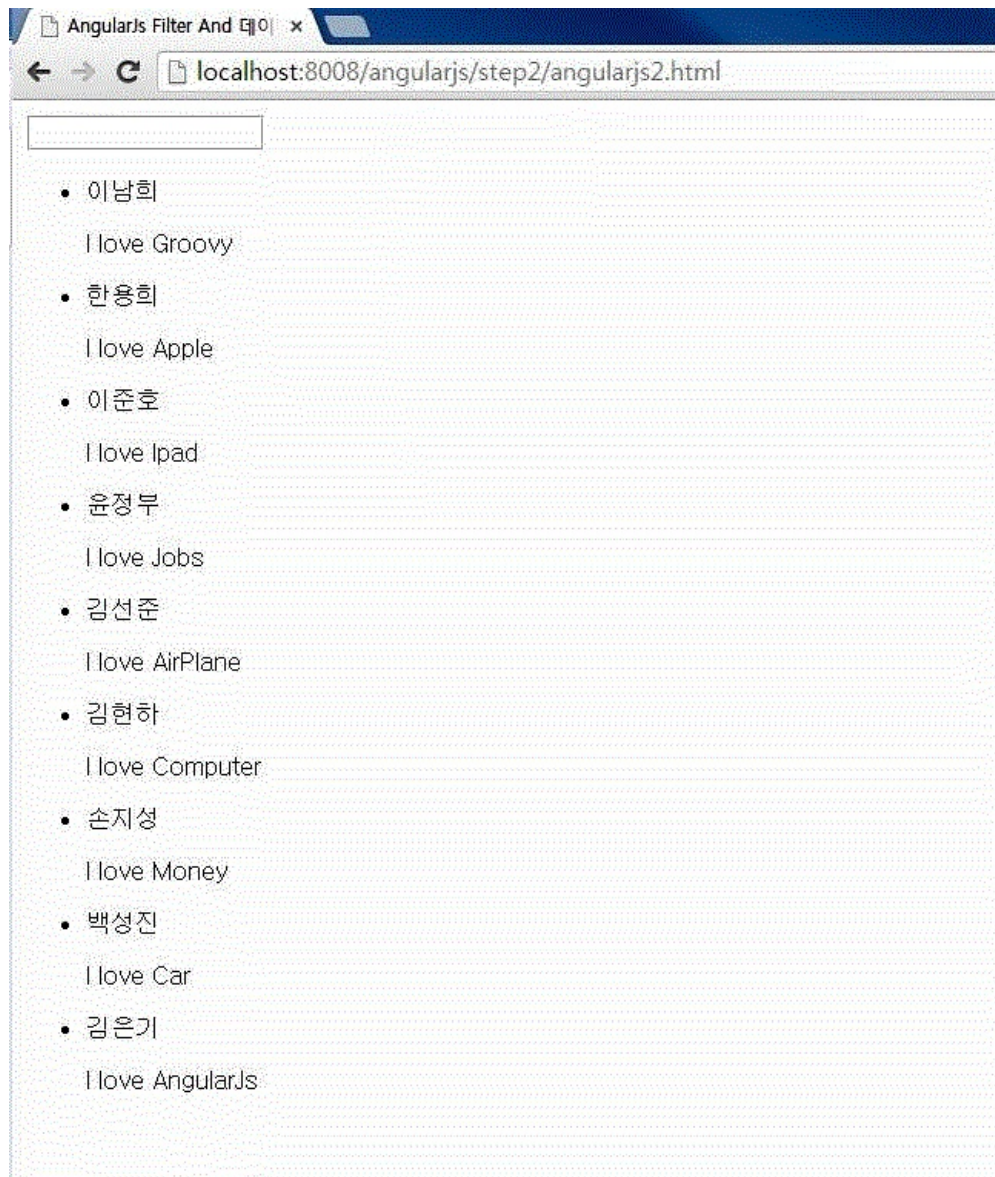
저번 시간에도 설명했듯이, **데이터 바인딩**은 AngularJs의 가장 중요한 기능이라고 볼 수 있다. 페이지 로드 후 input 박스 및 모델을 사용하는 영역에 대해 이들을 연결하고 뷰 <-> 컨트롤러를 연결시키고 **변화를 감지하여** 실시간으로 적용해 주는 역할을 하는 것이다.

```
<input type="text" ng-model="searchContent"/>
```

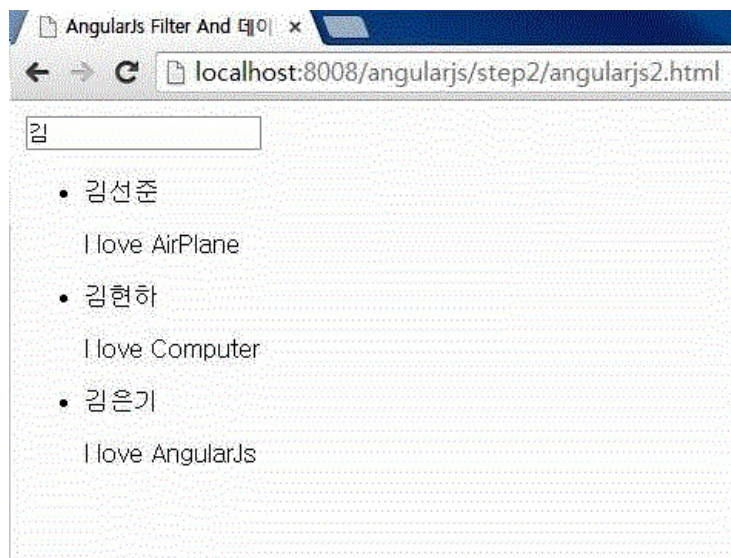
위 코드에서 필터에 의해 데이터 모델이 변경되면 새로운 Model을 생성함으로써, 효과적으로 DOM을 **업데이트 시켜주는** 역할을 한다고 볼 수 있다.

실제 소스 동작 화면

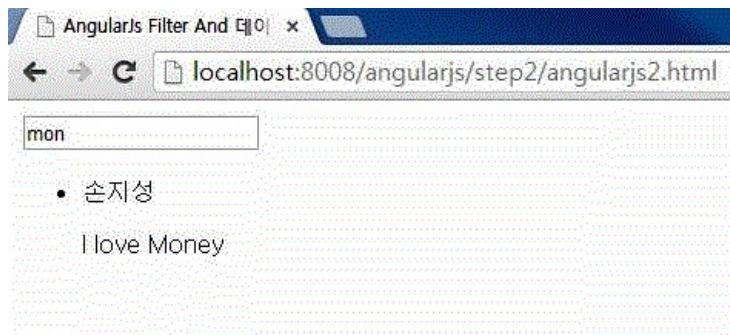
아래는 **실행 기본 화면**이다.



name으로 검색했을 경우



message로 검색했을 경우

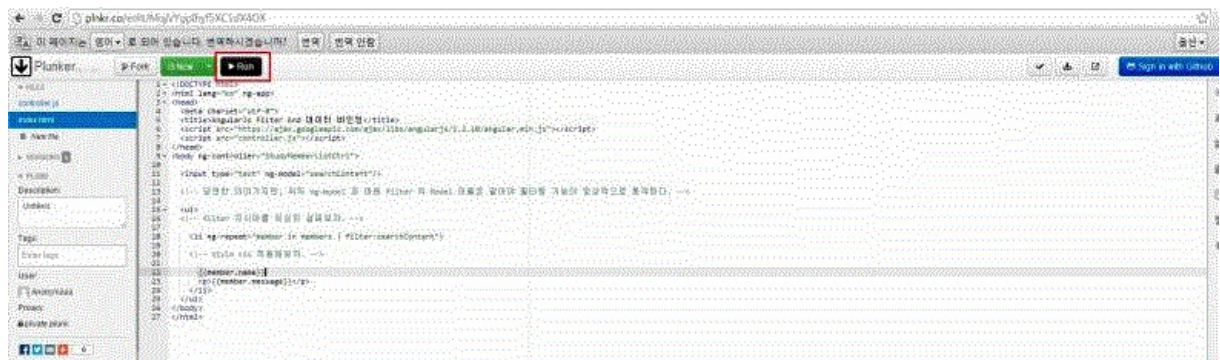


위 결과화면에서 확인할 수 있듯이, 실시간으로 필터를 통해 모델을 생성한 후 컨트롤러에 의하여 리스트가 동기화 되어 화면에 적용되는 것을 확인할 수 있다.

plunker를 위한 실시간 테스트 방법

<http://plnkr.co/edit/MqjVYgpIhyf5XC1dX4OX>

위 주소를 클릭하면 오늘 예제 소스를 작성한 후 실시간으로 변경, 테스트 할 수 있다. **가능하면 한 번 해보도록 하자.**



HTML 화면 또는 js파일 수정 후 위의 **RUN** 버튼을 클릭하면 된다.



Plunker는 jsfiddle와 같은 사이트의 일종으로 브라우저에서 자바스크립트 관련 소스코드를 **개발 및 실행, 공유** 할 수 있는 웹도구이다. 아카이브를 통해서 개발했던 자료를 쉽게 검색가능하며, Github을 통해서 공개된 소스코드를 다운 받는 것도 가능하다. 적극 활용에 보자.

최종 요약 정리

자, 다시 한번 오늘 예제 동작방식을 정리해 보도록 하자.

1. 필터에 변화가 감지되면, 입력된 문자 값으로 필터링 하여 **새로운 모델을 만들어** 낸다.
2. 변화된 모델은 즉시, **컨트롤러에 의해 감지되어 연결된 뷰에 즉시 반영된다.**
3. 반영됨과 동시에 DOM을 **자동으로 업데이트** 시키므로, 결과 값을 바로 확인 할 수 있다.



오늘 예제를 **AngularJS**를 사용하지 말고, **jQuery** 또는 **javascript**로 작성해보면, AngularJS가 주는 **편리성 및 이점**을 정확하게 확인할 수 있을 것이다. 한 번 해보자.

링크 목록

- <http://plnr.co/edit/MqjVYgplhyf5XC1dX4OX> - <http://plnr.co/edit/MqjVYgplhyf5XC1dX4OX>
-