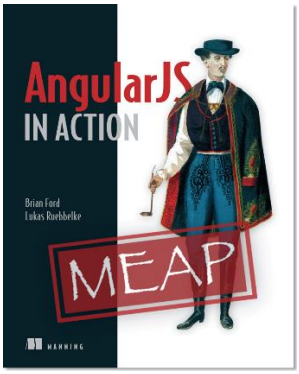
AngularJS: Display Stories with ngRepeat

By Brian Ford and Lukas Ruebbelke, *AngularJS in Action*



Wouldn't it be nice if we could define a template once and then just repeat it over and over for each item in the collection? This is exactly the role that ngRepeat was designed to play. This article, excerpted from AngularJS in Action, walks you through an example.

How would we take a collection of objects and display them on the page without having to define a layout for each item? Wouldn't it be nice if we could define a template once and then just repeat it over and over for each item in the collection? This is exactly the role that ngRepeat was designed to play. We will set the stage by defining the data structures that we are going to be working with for this example. We have an array that contains stories story objects as well as a statuses array that we will use to define our status columns in the view.

```
// client/src/angello/storyboard/controllers/StoryboardController.js
angular.module('Angello.Storyboard')
    .controller('StoryboardCtrl', function () {
        var storyboard = this;

        storyboard.stories = [
            {
                  "assignee": "1",
                  "criteria": "It tests!",
                  "description": "This is a test",
                  "id": "1", "reporter": "2",
                  "status": "To Do",
                  "title": "First Story",
                  "type": "Spike"
            },
            {
                  "assignee": "2",
                  "description": "testing something",
                  "id": "2",
                  "reporter": "1",
                  "status": "In Progress",
                  "title": "Second Story",
                  "type": "Enhancement"
            }
      ];
      storyboard.statuses = [

            {name: 'To Do'},
            {name: 'In Progress'},
            {name: 'Code Review'},
            {name: 'QA Review'},
```

```
            {name: 'Verified'}
        ];
});
```

We will start out by creating a column for each status in `storyboard.statuses` using `ngRepeat` on a `ul` element. The `ngRepeat` directive duplicates the element that it was declared on for each item in the collection that is providing data to the directive. Because it duplicates the child elements as well, we are able to define the layout once and just repeat it over and over. The expression "`status in storyboard.statuses`" essentially is telling AngularJS to repeat over the `storyboard.statuses` array, assign each item in the array, and refer to the current item in the iteration as status. This allows us to bind to a specific item in the array within the template like `{{status.name}}`, for instance.

NOTE *AngularJS is able to keep the instances of each item separate by implicitly creating a child scope for each template that is created by* `ngRepeat`*. Scope does an excellent job of providing context so we do not have to worry about those types of things colliding.*

```
// client/src/angello/storyboard/tmpl/storyboard.html
<div class="list-area">
      <div class="list-wrapper">
            <ul class="list my-repeat-animation" ng-repeat="status in
      storyboard.statuses">
            <h3 class="status">{{status.name}}</h3>
      <hr>
            </ul>
      </div>
</div>
```

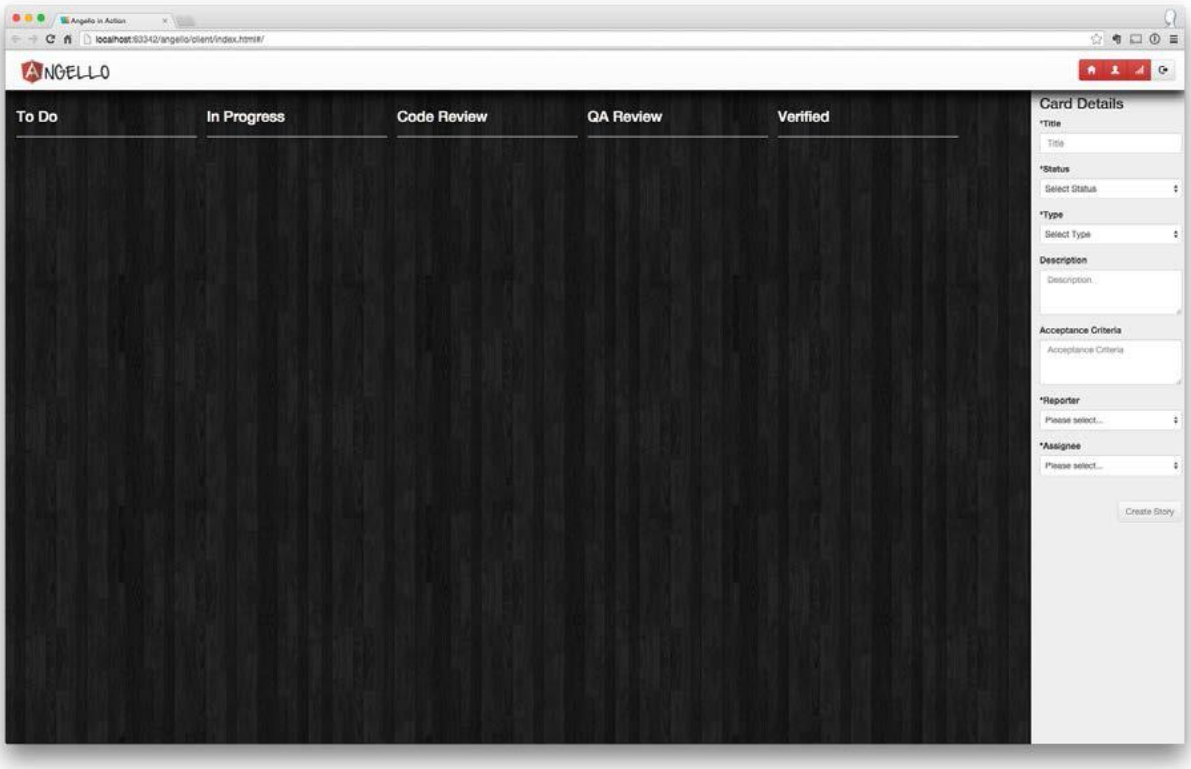


Figure 1 The status columns

Because we have five statuses in the `storyboard.statuses` array, the final result is five columns with the `status` name on top.

We are going to nest another `ngRepeat` within our first `ngRepeat` to add a list item for every story in `storyboard.stories`.

```
// client/src/angello/storyboard/tmpl/storyboard.html
<div class="list-area">
      <div class="list-wrapper">
            <ul class="list my-repeat-animation" ng-repeat="status in
      storyboard.statuses">
            <h3 class="status">{{status.name}}</h3>
            <hr>
            <li class="story" ng-repeat="story in storyboard.stories">
                  <article>
                        <div>
                              <button type="button" class="close">×</button>
                              <p class="title">{{story.title}}</p>
                        </div>
                        <div class="type-bar {{story.type}}"></div>
                        <div>
                              <p>{{story.description}}</p>
                        </div>
                  </article>
            </li>
      </ul>
   </div>
</div>
```

We will use the same convention as before and define our second `ngRepeat` as which `ng-repeat="story in storyboard.stories"` will loop over the `storyboard.stories` array and create a reference to the individual element as story. We then will use that reference to bind to and display `story.title` and `story.description` as well as assign a class based on `story.type`.

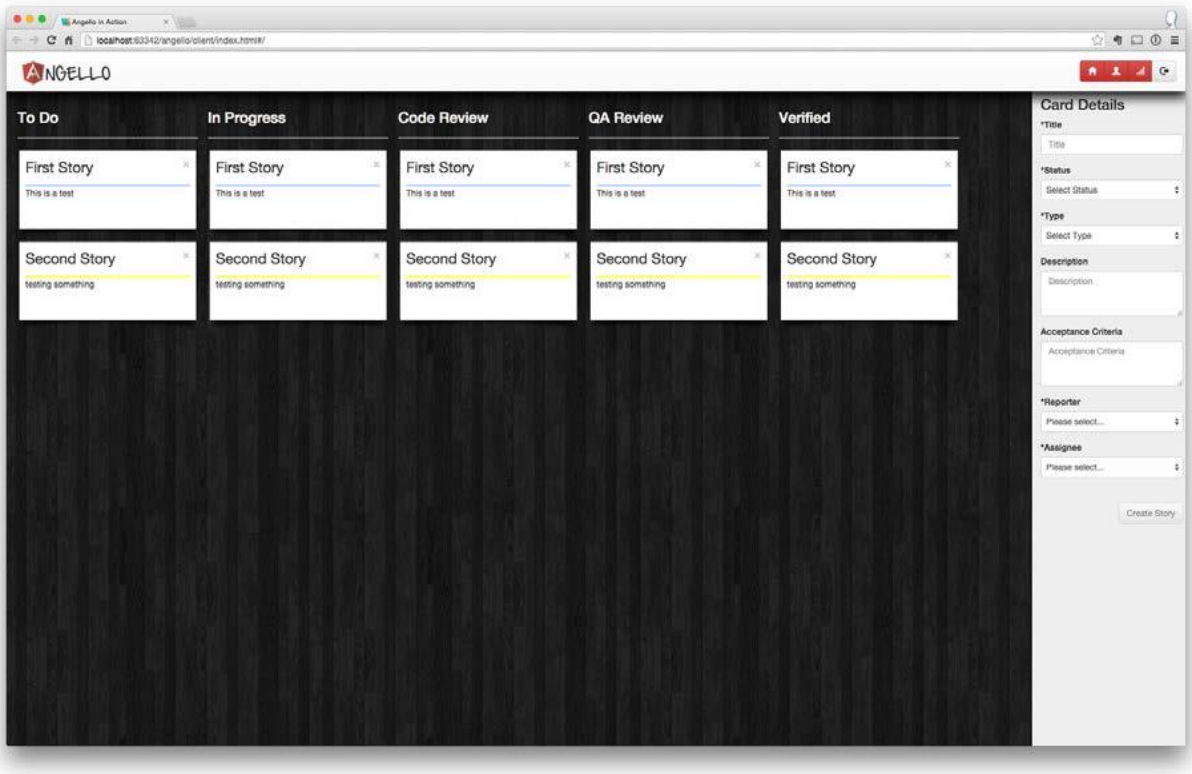


Figure 2 Stories in every column

The result is that we now have created a story element for each story in each of the status columns. This is obviously visually incorrect as we need a way to filter what we are displaying to show only the stories that match the status of the column they are in.