Adding an AngularJS Frontend



Reindert-Jan Ekker
AUTHOR TITLE

@rjekker www.rjekker.nl



Summary



A Simple AngularJS Page

- Dummy data (no REST yet)
- MVC
- Serving the page with Django

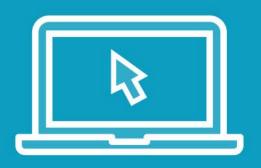
Directives and "real" data

- Ng-repeat to loop over lists
- Ng-model to handle input
- Ng-click to handle click event

Retrieving Data over REST



Demo



A Simple Angular Page

- Model: JavaScript variable
- View: HTML Template
- Controller

Using \$scope

- Using {{ }} expressions

Serving the Page with Django



```
(function () {
   'use strict';
   angular.module(
       'scrumboard.demo',
})();
<body ng-app=
   "scrumboard.demo">
```

- **◄** Code inside IIFE
- **◄** Strict mode

With a name

And no dependencies yet

- Immediately Invoke Function Expression
- Use ng-app in HTML

```
angular.module(
  'scrumboard.demo', [])
.controller(
    'ScrumboardController',
      '$scope',
       ScrumboardController
```

- **◄** List of dependencies

Scope

Reference to Ctrl. Function

```
function
ScrumboardController(
  $scope.person =
      name: "Joe",
      age: "35"
```

◄ Actual Controller function

■ Arguments as injected

■ Add data to the scope

Model data



```
<div ng-controller=
  "ScrumboardController">
     {{ person.name }} is
     {{ person.age }} years old
</div>
```

■ Using Our Controller in this div

■ Render \$scope properties in HTML



Django has its own template language

It uses {{ ... }} syntax like angular

■ Surround angular code with

```
{% verbatim %}
```

Will prevent conflicts

```
# In scrumboard/urls.py
from django.views.generic
   import TemplateView
urlpatterns = [
  # ...
 url(r'^home',
    TemplateView.as_view(
    template_name =
      "scrumboard/home.html"
))]
```

- Serving home.html with Django
- Import TemplateView

- Add a line to urlpatterns
- Map to url home
- **■** Use the TemplateView class
- **◄ It will serve the template in**
- templates/scrumboard/home.html

Demo



Directives

- Using ng-repeat to loop over lists
- Using ng-click to call a function
- Using ng-model to bind input to scope

Serving JavaScript as Static Files



```
<div ng-controller=
 "ScrumboardController">
 <div ng-repeat=
   "list in data">
   <h3>{{ list.name }}</h3>
   ng-repeat=
       "card in list.cards">
</div></div>
```

■ Using Our Controller in this div

■ Loop over \$scope.data

■ Render list.name in HTML

■ Loop over cards in list

■ Render title in HTML



```
<input
 ng-model="new_title"/>
<button
 ng-click=
  "add(list, new_title)">
```

■ Ng-model binds this input to \$scope.new_title

■ Ng-click runs the expression
When button is clicked
This runs \$scope.add()

■ Render new_title Automatically updated when input changes



```
$scope.add =
  function (list, title) {
    var card = {
      title: title
    list.cards.push(card);
```

■ Defining the add function as a property on \$scope

◄ Create the new card object

■ Add it to the list
HTML will automatically update

```
{% load staticfiles %}
<script src=</pre>
  "{% static
   "scrumboard/scrumboard.js"
  %}">
</script>
```

- Make static tag available

 Should be at top of template
- ◄ Include the script
- Using the {% static %} tag
- Referencing the file Relative to the static folder This will generate a URL

Demo



Retrieving Data over REST
Injecting the \$http service
Using the \$http service



```
angular.module(
  'scrumboard.demo', [])
  .controller(
    'ScrumboardController',
    ['$scope',
     '$http',
     ScrumboardController
    ]);
```

- Angular.module creation
- **◄** Creating the Controller

- ◀ Inject \$http
- **◄** Controller Function last

```
function
ScrumboardController(
 $scope, $http
  $http.get(
    '/scrumboard/lists')
    .then(function(resp)
    $scope.lists = resp.data;
  });
```

■ Inject \$http as 2nd argument

- **← Call** \$http.get
- ◆ Pass REST API URL
- When data is received

■ Save data in scope
 View will auto-update

Summary



A Simple Angular Page

- MVC
- Scope
- Directives: ng-app, ng-repeat

Serving the Page with Django

Two-way Binding

- ng-model and ng-click

Retrieving Data over REST

- \$http

