Editing Data

Reindert-Jan Ekker

@rjekker www.rjekker.nl

Summary

Edit, Save and Delete Cards

REST

- POST, PUT, DELETE

Creating a Directive for Cards

Styling and Interactivity

A Word About REST

Getting cards

GET
/scrumboard/cards/

Creating a card

POST
/scrumboard/cards/

Updating a card

PUT
/scrumboard/cards/5

Deleting a card

DELETE
/scrumboard/cards/5

Getting a card

GET
/scrumboard/cards/5

REST and Our Application

1. Backend: ViewSets and Router

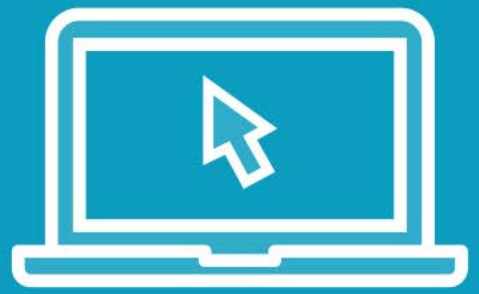
2. Saving Cards: POST

3. Editing Cards: PUT

(Directive)

4. Deleting Cards: DELETE

Demo



Saving New Cards With Post: Backend

- Viewsets
- Router

```
# In api.py
from rest_framework.viewsets import ModelViewSet


class CardViewSet(ModelViewSet):
    queryset = List.objects.all()
    serializer_class = ListSerializer
```

ModelViewSet

Support GET, PUT, POST, DELETE

Replace ListAPIView

Implements a number of views at once

```
# In urls.py


from .api import ListViewSet, CardViewSet
from rest_framework.routers import DefaultRouter


router = DefaultRouter()
```

Creating a Router

Easily configure URLS for our viewset

We simply use a DefaultRouter instance

All URLs end in a slash!

```
router.register(r'lists', ListViewSet)
router.register(r'cards', CardViewSet)

urlpatterns = router.urls
```

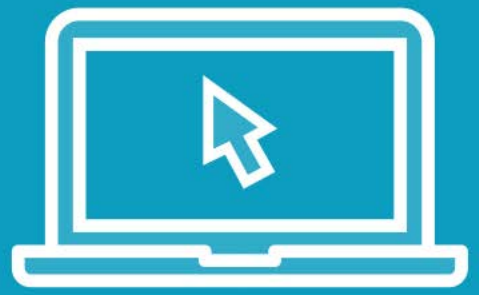
Registering the URLs

Call router.register with URL prefix and Viewset

This works with multiple Viewsets

Assign contents of router.urls to urlpatterns for the Django app

Demo



Saving New Cards With Post: Frontend

- `$http.post()`
- Problem: CSRF protection
- Error handling

```
$http.post(
  '/scrumboard/cards/',card)
    .then(function(response){
       list.cards.push(
            response.card
       );
    },
    function(){
       // handle error
}););
```

- POST the new card
- To the cards collection URL
- When server responds:
- Add card to model
  Using object from server

- Second function argument:
  Error handling

403 Forbidden

In case API calls return a “403 Forbidden” error

Caused by “CSRF protection”

Solution: Log out of the admin interface

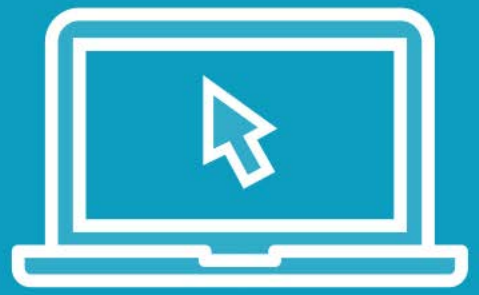
Custom Angular Directive

Reusable Component for Cards

HTML and JavaScript

Custom tag

Demo



Creating a Custom Angular Directive

- Template
- Binding
- Passing data to the directive
- CSS Styling

```
// new file card.directive.js


(function () {
    angular.module('scrumboard.demo')
      .directive('scrumboardCard', CardDirective);
})();
```

Creating an Angular Directive

IIFE – Immediately Invoked Function Expression

Call `angular.module` to retrieve module object

Call `directive` method on module object

```
function CardDirective() {
return {
        templateUrl: '/static/scrumboard/card.html',
        restrict: 'E‘,
};}
```

Implementing the Directive

`templateURL`: **location of HTML template**

`restrict`: **‘E’ for element, ‘A’ for attribute**

```
<ul>
  <li ng-repeat=
   "card in list.cards">

    <scrumboard-card>
    </scrumboard-card>

    </li>
</ul>
```

- Repeat for each card in list

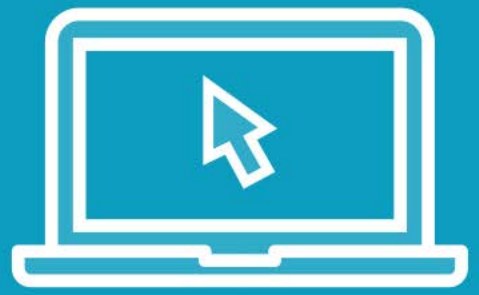
- Element: `scrumboard-card`

  Will contain template HTML

  Variable `card` visible in scope

  Name uses dash instead of CamelCase

Demo



Updating Cards With Put

- Make cards editable on click
- `ng-show` and `ng-hide` to show form
- Using `ng-change` for autosave

```
<div ng-hide="edit"
     ng-click="edit=true“ >
    .. Showing data ..
</div>

<div ng-show="edit">
    .. Controls for editing ..
</div>
```

- Hide element when expression is true

- Same, but shows element when true

```
<input type="text"
       ng-model=“card.title"
       ng-model-options=
         "modelOptions“
       ng-change=“update()"
/>
```

- ng-change: call update() when input text changes

```
controller:
  ['$scope', '$shttp'
    function ($scope, $http) {
      ...
  }]
```

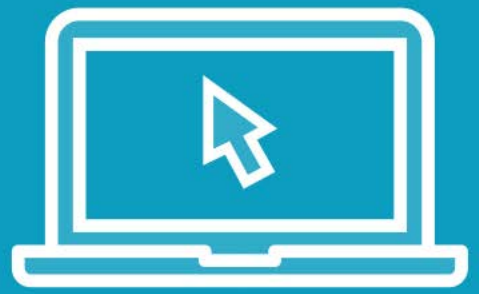
- Controller attribute for directive
- List syntax: dependencies first
- Controller implementations

Updating the Card

```
var url = '/scrumboard/cards/‘ +
          $scope.card.id + '/';

$scope.update = function () {
    $http.put(url, $scope.card);
};
```

Demo



Deleting Cards

`ng-click` **and** `$http.delete`

Summary

Edit, Save and Delete Cards

REST

- Client: $http.post(), .put(), .delete()
- Server: Using ViewSets and Router

Creating a Directive for Cards

- HTML tag
- Template
- Controller

Styling and Interactivity

- ng-show, ng-hide, ng-change
- CSS