

Login, Authentication, and Routing



Reindert-Jan Ekker

@rjekker www.rjekker.nl



Summary



Login and Logout

Backend

- REST API Views
- Django App
- CSRF protection

Frontend

- Login Page with Form
- Routing
- Reorganize HTML/JS Code



Demo



Backend Views for Login/Logout

- Create a new Django App
- Add Custom ApiView Classes



CSRF Protection



Requests that change data on the server

- POST, PUT, DELETE
- GET should never do this!

Prevent “cross-site” requests

Set a csrf-token cookie

Client should send token back as header

Demo



Configuring Authentication

- Restricted Views
- CSRF Protection



```
from rest_framework import status, views
from rest_framework.response import Response
class LogoutView(views.APIView):
    def get(self, request):
        # do some work
        return Response({}, status=status.HTTP_200_OK)
```

Custom REST Views

Inherit from `APIView`

Contains methods for HTTP methods `get`, `post`, ...

Return a `Response` object



```
from django.contrib.auth import authenticate, login

user = authenticate(
    username=request.data.get("username"),
    password=request.data.get("password"))

login(request, user)    # ONLY if user is not None
```

Logging the User In

`authenticate` checks credentials and returns `User` instance or `None`

Use `request.data.get` to retrieve form data

`login` creates the user login session

Of course there is also a `logout` method



```
from rest_framework import permissions

class ListViewSet(ModelViewSet):
    ...
    permission_classes = (permissions.IsAuthenticated,)
```

Restricting ViewSet Permissions

Add a `permission_classes` attribute to the `ViewSet`

Assign a tuple (don't forget the comma)

Tuple contains `permissions.IsAuthenticated`




```
angular.module('scrumboard.demo').run(['$http', run]);

function run($http) {
    $http.defaults.xsrfHeaderName = 'X-CSRFToken';
    $http.defaults.xsrfCookieName = 'csrftoken';
}
```

Angular and CSRF Tokens

New file `scrumboard.config.js`

Retrieve module and call `run()` with a function as argument

We configure both cookie and header name for CSRF token

Decorators for Django views: `csrf_protect` and `ensure_csrf_cookie` 

Login Frontend: Three Steps

Reorganize HTML/JS Code

Move everything to
top-level folders

Add Routing

Show different HTML
views depending on
URL

Add Login View

To be served at /login



Demo



Reorganizing the Frontend

- Top-level templates and static
- Uncoupled from Django Apps
- Organized by File Type



Demo



Routing

- Add the `ngRoute` module
- Configure URLs and Views
- `home.html` for global layout
- Use `ng-view` to include views



Demo



Login and Logout Frontend

- `ng-submit` to handle submit button
- `LoginController`
- Using `$location` to change path



```
TEMPLATES = [{  
    'DIRS': [os.path.join(BASE_DIR, 'templates')],  
    ...  
}, ]
```

```
STATICFILES_DIRS = (os.path.join(BASE_DIR, 'static'),)
```

Toplevel Folders

In `django/settings.py`

Setting for templates is a list of dictionaries

Change `'DIRS'` to contain top-level templates folder

Add `STATICFILES_DIRS`: a tuple containing top-level static folder



```
<body ng-app="scrumboard.demo">  
  <ng-view></ng-view>  
  <script src="https://ajax.googleapis.com/ajax/libs/  
    angularjs/X.Y.Z/angular-route.js"></script>  
</body>
```

ngRoute: main HTML file

When URL changes:

ngRoute will insert corresponding template inside ng-view tag



Routing Configuration Function

```
// scrumboard.config.js

angular.module('scrumboard.demo')
    .config(['$routeProvider', config]);

function config($routeProvider) {
    // routes go here
}
```




```
$routeProvider
    .when('/login', {
        templateUrl: '/static/html/login.html',
        controller: 'LoginController'
    })
    .otherwise('/');
```

ngRoute: Routing Configuration

`when()` associates a URL (string) with a route (object)

A route can have `templateURL` and `controller` `properties`

`otherwise()` redirects to a default URL

Matches part of URL after `#` - no page refresh!



ngRoute as a Dependency of Our App

```
// scrumboard.js
```

```
// this is the line that creates our module
```

```
// and sets the dependencies
```

```
angular.module('scrumboard.demo', ['ngRoute']);
```



ng-submit and \$location

```
ng-submit='login()'
```

Define the action for submitting form by clicking submit or pressing enter

```
$location.url = '/'
```

Used to change the URL: 'redirect' to another view



Summary



Login and Logout

Backend

- REST API Views
- Restricting views to Authorized Users
- CSRF protection

Frontend

- Reorganize HTML/JS Code
- Routing and \$location
- Ng-submit

