
Django REST Framework Keycloak Documentation

Release 0.2.0

Marcos Pereira

Sep 27, 2017

Contents:

1	Welcome to Django REST Framework Keycloak's documentation!	3
2	Installation	5
3	Dependencies	7
3.1	Tests Dependencies	7
4	Bug reports	9
5	Documentation	11
6	Contributors	13
7	Usage	15

- [genindex](#)
- [modindex](#)
- [search](#)

CHAPTER 1

Welcome to Django REST Framework Keycloak's documentation!

django-rest-framework-keycloak package provides Keycloak support.

CHAPTER 2

Installation

Via Pypi Package:

```
$ pip install django-rest-framework-keycloak
```

Manually:

```
$ python setup.py install
```

Dependencies

django-rest-framework-keycloak depends on:

- Python 3
- `django`
- `djangoRESTframework`
- `python-keycloak`

Tests Dependencies

- `unittest`

CHAPTER 4

Bug reports

Please report bugs and feature requests at <https://bitbucket.org/agriness/django-rest-framework-keycloak/issues>

CHAPTER 5

Documentation

The documentation for **django-rest-framework-keycloak** is available on [readthedocs](#).

CHAPTER 6

Contributors

- Agriness Team

CHAPTER 7

Usage

1. Add “django_keycloak” to your INSTALLED_APPS setting like this:

```
INSTALLED_APPS = [  
    ...  
    'django_keycloak',  
]
```

2. Add “keycloak_django.middleware.KeycloakMiddleware” to your MIDDLEWARE setting like this:

```
MIDDLEWARE = [  
    ...  
    'keycloak_django.middleware.KeycloakMiddleware'  
    ...  
]
```

3. Add configure Keycloak:

```
KEYCLOAK_CONFIG = {  
    'KEYCLOAK_SERVER_URL': 'http://localhost/auth/',  
    'KEYCLOAK_REALM': 'your_realm',  
    'KEYCLOAK_CLIENT_ID': 'your_client',  
    'KEYCLOAK_CLIENT_SECRET_KEY': 'secret_key',  
    'KEYCLOAK_CLIENT_PUBLIC_KEY': 'public_key',  
    'KEYCLOAK_DEFAULT_ACCESS': 'DENY', # DENY or ALLOW (Default is DENY)  
    'KEYCLOAK_AUTHORIZATION_CONFIG': os.path.join(BASE_DIR, 'your-client-authz-  
→config.json'),  
    'KEYCLOAK_METHOD_VALIDATE_TOKEN': 'INTROSPECT', # INTROSPECT OR DECODE (Default_  
→is INTROSPECT)  
}
```

4. Map the scopes of the APIView:

```
from django.http.response import JsonResponse  
from rest_framework.views import APIView
```

```
class AdminView(APIView):
    keycloak_scopes = {'GET': 'read-only-admin-view',
                      'POST': 'edit-admin-view'}

    def get(self, request, **kwargs):
        return JsonResponse({"page": "Admin Resource"})

    def post(self, request, format=None):
        return JsonResponse({"page": "Edit Admin Resource"})
```