
edX Django REST Framework Extensions Documentation

Release 2.3.0

edX

May 28, 2019

1	Requirements	3
2	Installation	5
3	Table of Contents	7
3.1	Settings	7
3.2	Authentication	8
3.3	Middleware	8
3.4	Permissions	8
3.5	Utility Functions	9
	Python Module Index	11

This package provides extensions for [Django REST Framework](#) that are useful for developing on the edX platform.

Requirements

- Python (2.7, 3.5, 3.6)
- Django (1.8, 1.9, 1.10, 1.11)
- Django REST Framework (3.2+)

Installation

Install using pip:

```
$ pip install edx-drf-extensions
```

Table of Contents

3.1 Settings

All settings for this package reside in a dict, `EDX_DRF_EXTENSIONS`. Within this dict, the following keys should be specified, depending on the functionality you are using.

3.1.1 BearerAuthentication

These settings are used by the `BearerAuthentication` class.

`OAUTH2_USER_INFO_URL`

Default: None

URL of an endpoint on the OAuth2 provider where `BearerAuthentication` can retrieve details about the user associated with the provided access token. This endpoint should return a JSON object with user details and HTTP 200 if, and only if, the access token is valid. See `BearerAuthentication.process_user_info_response()` for an example of the expected data format.

3.1.2 JwtAuthentication

These settings are used by the `JwtAuthentication` class. Since this class is based on `JSONWebTokenAuthentication`, most of its settings can be found in the documentation for `rest_framework_jwt` at <http://getblimp.github.io/django-rest-framework-jwt/#additional-settings>.

`JWT_AUTH['JWT_VERIFY_AUDIENCE']`

Default: True

If you do *not* want to verify the JWT audience, set the `'JWT_VERIFY_AUDIENCE'` key in the `JWT_AUTH` setting to `False`.

`JWT_PAYLOAD_USER_ATTRIBUTES`

Default: `('email',)`

The list of user attributes in the JWT payload that `JwtAuthentication` will use to update the local `User` model. These payload attributes should exactly match the names the attributes on the local `User` model.

3.2 Authentication

Authentication classes are used to associate a request with a user. Unless otherwise noted, all of the classes below adhere to the Django REST Framework's API for authentication classes.

3.3 Middleware

This module contains middleware to ensure best practices of DRF and other endpoints..

class `RequestMetricsMiddleware`

Adds various request related metrics.

Possible metrics include:

request_auth_type: Example values include: **no-user, unauthenticated,** jwt, bearer, other-token-type, or session-or-unknown

request_client_name: The client name from edx-rest-api-client calls. **request_referer** **request_user_agent:** The user agent string from the request header. **request_user_id:** The user id of the request user.

This middleware is dependent on the `RequestCacheMiddleware`. You must include this middleware later. For example:

```
MIDDLEWARE_CLASSES = (
    'edx_django_utils.cache.middleware.RequestCacheMiddleware',
    'edx_rest_framework_extensions.middleware.RequestMetricsMiddleware',
)
```

This middleware should also appear after any authentication middleware.

process_response (*request, response*)

Add metrics for various details of the request.

3.4 Permissions

Permissions determine whether a request should be granted or denied access. Unless otherwise noted, all of the classes below adhere to the Django REST Framework's API for permission classes.

class `IsStaff`

Allows access to “global” staff users..

class `IsSuperuser`

Allows access only to superusers.

class `IsUserInUrl`

Allows access if the requesting user matches the user in the URL.

class `JwtHasContentOrgFilterForRequestedCourse`

The JWT used to authenticate contains the appropriate content provider filter for the requested course resource.

has_permission (*request*, *view*)

Ensure that the `course_id` kwarg provided to the view contains one of the organizations specified in the content provider filters in the JWT used to authenticate.

class JwtHasScope

The request is authenticated as a user and the token used has the right scope.

class JwtHasUserFilterForRequestedUser

The JWT used to authenticate contains the appropriate user filter for the requested user resource.

has_permission (*request*, *view*)

If the JWT has a user filter, verify that the filtered user value matches the user in the URL.

class JwtRestrictedApplication

Returns whether the request was successfully authenticated with `JwtAuthentication` by a `RestrictedApplication`.

3.5 Utility Functions

This module contains useful utility functions.

e

`edx_rest_framework_extensions.middleware,`
 8
`edx_rest_framework_extensions.permissions,`
 8
`edx_rest_framework_extensions.utils,`9

E

`edx_rest_framework_extensions.middleware` (module), 8
`edx_rest_framework_extensions.permissions` (module), 8
`edx_rest_framework_extensions.utils` (module), 9

H

`has_permission()` (`JwtHasContentOrgFilterForRequestedCourse` method), 8
`has_permission()` (`JwtHasUserFilterForRequestedUser` method), 9

I

`IsStaff` (class in `edx_rest_framework_extensions.permissions`), 8
`IsSuperuser` (class in `edx_rest_framework_extensions.permissions`), 8
`IsUserInUrl` (class in `edx_rest_framework_extensions.permissions`), 8

J

`JwtHasContentOrgFilterForRequestedCourse` (class in `edx_rest_framework_extensions.permissions`), 8
`JwtHasScope` (class in `edx_rest_framework_extensions.permissions`), 9
`JwtHasUserFilterForRequestedUser` (class in `edx_rest_framework_extensions.permissions`), 9
`JwtRestrictedApplication` (class in `edx_rest_framework_extensions.permissions`), 9

P

`process_response()` (`RequestMetricsMiddleware` method), 8

R

`RequestMetricsMiddleware` (class in `edx_rest_framework_extensions.middleware`), 8