flask-jwt Documentation

Release 0.3.2

Dan Jacob

May 21, 2017

Contents

Add basic JWT features to your Flask application.

CHAPTER 1

Links

- documentation
- source
- *changelog*

CHAPTER 2

Installation

Install with **pip** or **easy_install**:

```
pip install Flask-JWT
```

or download the latest version from version control:

```
git clone https://github.com/mattupstate/flask-jwt.git ./flask-jwt
pip install ./flask-jwt
```

CHAPTER 3

Quickstart

Minimum viable application configuration:

```
from flask import Flask
from flask_jwt import JWT, jwt_required, current_identity
from werkzeug.security import safe_str_cmp

class User(object):
    def __init__(self, id, username, password):
        self.id = id
        self.username = username
        self.password = password

    def __str__(self):
        return "User(id='%s')" % self.id

users = [
    User(1, 'user1', 'abcxyz'),
    User(2, 'user2', 'abcxyz'),
]

username_table = {u.username: u for u in users}
userid_table = {u.id: u for u in users}

def authenticate(username, password):
    user = username_table.get(username, None)
    if user and safe_str_cmp(user.password.encode('utf-8'), password.encode('utf-8')):
        return user

def identity(payload):
    user_id = payload['identity']
    return userid_table.get(user_id, None)

app = Flask(__name__)
app.debug = True
app.config['SECRET_KEY'] = 'super-secret'
```

```
jwt = JWT(app, authenticate, identity)

@app.route('/protected')
@jwt_required()
def protected():
    return '%s' % current_identity

if __name__ == '__main__':
    app.run()
```

To get a token make a request to the auth resource:

```
POST /auth HTTP/1.1
Host: localhost:5000
Content-Type: application/json

{
    "username": "joe",
    "password": "pass"
}
```

The response should look similar to:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
↪eyJpZGVudGl0eSI6MSwiaWF0IjoxNDQ0OTE3NjQwLCJuYmYiOjE0NDQ5MTc2NDAsImV4cCI6MTQ0NDkxNzk0MH0.
↪KPmI6WSjRjlpzecPvs3q_T3cJQvAgJvaQAPtk1abC_E"
}
```

This token can then be used to make requests against protected endpoints:

```
GET /protected HTTP/1.1
Authorization: JWT eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
↪eyJpZGVudGl0eSI6MSwiaWF0IjoxNDQ0OTE3NjQwLCJuYmYiOjE0NDQ5MTc2NDAsImV4cCI6MTQ0NDkxNzk0MH0.
↪KPmI6WSjRjlpzecPvs3q_T3cJQvAgJvaQAPtk1abC_E
```

Within a function decorated by *jwt_required()*, you can use the *current_identity* proxy to access the user whose token was passed into this request context.

CHAPTER 4

Configuration Options

JWT_DEFAULT_REALM	The default realm. Defaults to `Login Required`
JWT_AUTH_URL_RULE	The authentication endpoint URL. Defaults to `/auth`.
JWT_AUTH_ENDPOINT	The authentication endpoint name. Defaults to `jwt`.
JWT_AUTH_USERNAME_KEY	The username key in the authentication request payload. Defaults to `username`.
JWT_AUTH_PASSWORD_KEY	The password key in the authentication request payload. Defaults to `password`.
JWT_ALGORITHM	The token algorithm. Defaults to `HS256`
JWT_LEEWAY	The amount of leeway given when decoding access tokens specified as an integer of seconds or a `datetime.timedelta` instance. Defaults to `timedelta(seconds=10)`.
JWT_VERIFY	Flag indicating if all tokens should be verified. Defaults to `True`. It is not recommended to change this value.
JWT_AUTH_HEADER_PREFIX	The Authorization header value prefix. Defaults to `JWT` as to not conflict with OAuth2 Bearer tokens. This is not a case sensitive value.
JWT_VERIFY_EXPIRATION	Flag indicating if all tokens should verify their expiration time. Defaults to `True`. It is not recommended to change this value.
JWT_LEEWAY	A token expiration leeway value. Defaults to `0`.
JWT_EXPIRATION_DELTA	A `datetime.timedelta` value indicating how long tokens are valid for. This value is added to the `iat` (issued at) claim. Defaults to `timedelta(seconds=300)`
JWT_NOT_BEFORE_DELTA	A `datetime.timedelta` value indicating a relative time from the `iat` (issued at) claim that the token can begin to be used. This value is added to the `iat` (issued at) claim. Defaults to `timedelta(seconds=0)`
JWT_VERIFY_CLAIMS	A list of claims to verify when decoding tokens. Defaults to `['signature', 'exp', 'nbf', 'iat']`.
JWT_REQUIRED_CLAIMS	A list of claims that are required in a token to be considered valid. Defaults to `['exp', 'iat', 'nbf']`

CHAPTER 5

API

flask_jwt.**current_identity**
A proxy for the current identity. It will only be set in the context of function decorated by *jwt_required()*.

class flask_jwt.**JWT**(*app=None*, *authentication_handler=None*, *identity_handler=None*)

auth_request_handler(*callback*)
Specifies the authentication response handler function.

Parameters **callback** (*callable*) – the auth request handler function

auth_response_handler(*callback*)
Specifies the authentication response handler function.

Parameters **callback** (*callable*) – the auth response handler function

authentication_handler(*callback*)
Specifies the identity handler function. This function receives two positional arguments. The first being the username the second being the password. It should return an object representing an authenticated identity. Example:

```
@jwt.authentication_handler
def authenticate(username, password):
    user = User.query.filter(User.username == username).scalar()
    if bcrypt.check_password_hash(user.password, password):
        return user
```

Parameters **callback** – the identity handler function

identity_handler(*callback*)
Specifies the identity handler function. This function receives one positional argument being the JWT payload. For example:

```
@jwt.identity_handler
def identify(payload):
    return User.query.filter(User.id == payload['identity']).scalar()
```

Parameters **callback** – the identity handler function

jwt_decode_handler(*callback*)
Specifies the decoding handler function. This function receives a signed payload and decodes it.

Parameters **callback** (*callable*) – the decoding handler function

jwt_encode_handler(*callback*)
Specifies the encoding handler function. This function receives a payload and signs it.

Parameters **callback** (*callable*) – the encoding handler function

jwt_error_handler(*callback*)
Specifies the error handler function. Example:

```
@jwt.error_handler
def error_handler(e):
    return "Something bad happened", 400
```

Parameters **callback** – the error handler function

jwt_headers_handler(*callback*)
Specifies the JWT header handler function. This function receives the return value from the `identity_handler` function.

Example:

```
@jwt.payload_handler
def make_payload(identity):
    return {'user_id': identity.id}
```

Parameters **callback** (*callable*) – the payload handler function

jwt_payload_handler(*callback*)
Specifies the JWT payload handler function. This function receives the return value from the `identity_handler` function

Example:

```
@jwt.payload_handler
def make_payload(identity):
    return {'user_id': identity.id}
```

Parameters **callback** (*callable*) – the payload handler function

request_handler(*callback*)
Specifieds the request handler function. This function returns a JWT from the current request.

Parameters **callback** (*callable*) – the request handler function

flask_jwt.**jwt_required**(*realm=None*)
View decorator that requires a valid JWT token to be present in the request

Parameters **realm** – an optional realm

CHAPTER 6

Changelog

Flask-JWT Changelog

Here you can see the full list of changes between each Flask-JWT release.

Version 0.3.2

Released November 3rd 2015

- Fixed an Authorization header conditional bug

Version 0.3.1

Released October 26th 2015

- Fix a bug with `auth_request_handler`
- Deprecate `auth_request_handler`

Version 0.3.0

Released October 15th 2015

Note: This release includes many breaking changes

- Fix major implementation issue with encoding/decoding tokens
- Changed new configuration options to align with PyJWT
- Changed `current_user` to `current_identity`

Version 0.2.0

Released June 10th 2014

- Fixed an issue where `current_user` was not None
- Added a response handler hook to be able to adjust auth response(s)
- Removed the configurable handlers in favor of decorators
- Removed pyjwt dependency

Version 0.1.0

Released March 5th 2014

- Initial release

Python Module Index

f

Index