AdafruitJWT Library Documentation

Release 1.0

Brent Rubell

Apr 07, 2020

Contents

JSON Web Token (JWT) Authentication module for CircuitPython. JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties.

This library currently supports the following signature algorithms for JWT generation and verification:

- No encoding ("none")
- RS256/SHA-256 (via Adafruit_CircuitPython_RSA)
- RS384/SHA-384 (via Adafruit_CircuitPython_RSA)
- RS512/SHA-512 (via Adafruit_CircuitPython_RSA)

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython
- Adafruit_CircuitPython_RSA
- Adafruit_CircuitPython_binascii

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

CHAPTER 2

Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally from PyPI. To install for current user:

```
pip3 install adafruit-circuitpython-jwt
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-jwt
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-jwt
```

CHAPTER 3

Usage Example

Generating encoded JWT

```
import adafruit_jwt
# Import Private RSA key from a secrets.py file
try:
    from secrets import secrets
except ImportError:
    print("WiFi secrets are kept in secrets.py, please add them there!")
    raise

# Create JWT Claims
claims = {"iss": "joe",
        "exp": 1300819380,
        "name": "John Doe",
        "admin": True}

# Generate JWT, sign with RSA private key and RS-256
encoded_jwt = adafruit_jwt.JWT.generate(
    claims, secrets["private_key"], algo="RS256")
print("Encoded JWT: ", encoded_jwt)
```

Validating a generated JWT, encoded_jwt.

```
import adafruit_jwt
decoded_jwt = adafruit_jwt.JWT.validate(encoded_jwt)
# The decoded JWT's JOSE header and claims set are returned as a tuple
print('JOSE Header: {}\nJWT Claims: {}'.format(decoded_jwt[0], decoded_jwt[1]))
```

CHAPTER 4

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER 5

Documentation

For information on building library documentation, please check out this guide.

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/jwt_simpletest.py

```
import adafruit_jwt

# Get private RSA key from a secrets.py file
try:
    from secrets import secrets
except ImportError:
    print("WiFi secrets are kept in secrets.py, please add them there!")
    raise

# Sample JWT Claims
claims = {"iss": "joe", "exp": 1300819380, "name": "John Doe", "admin": True}

# Generate a JWT
print("Generating JWT...")
encoded_jwt = adafruit_jwt.JWT.generate(claims, secrets["private_key"], algo="RS256")
print("Encoded JWT: ", encoded_jwt)

# Validate a provided JWT
print("Decoding JWT...")
decoded_jwt = adafruit_jwt.JWT.validate(encoded_jwt)
print("JOSE Header: {}\nJWT Claims: {}".format(decoded_jwt[0], decoded_jwt[1]))
```

6.2 adafruit_jwt

JSON Web Token Authentication

- Author(s): Brent Rubell

6.2.1 Implementation Notes

Hardware:

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: https://github.com/adafruit/circuitpython/releases
- Adafruit's RSA library: https://github.com/adafruit/Adafruit_CircuitPython_RSA
- Adafruit's binascii library: https://github.com/adafruit/Adafruit_CircuitPython_RSA

class adafruit_jwt.**JWT**

JSON Web Token helper for CircuitPython. Warning: JWTs are credentials, which can grant access to resources. Be careful where you paste them! :param str algo: Encryption algorithm used for claims. Can be None.

static generate(*claims*, *private_key_data=None*, *algo=None*)

Generates and returns a new JSON Web Token. :param dict claims: JWT claims set :param str private_key_data: Decoded RSA private key data. :rtype: str

static validate(*jwt*)

Validates a provided JWT. Does not support validating nested signing. Returns JOSE Header and claim set. :param str jwt: JSON Web Token. :returns: The message's decoded JOSE header and claims. :rtype: tuple

class adafruit_jwt.**STRING_TOOLS**

Tools and helpers for URL-safe string encoding.

static translate(*s*, *table*)

Return a copy of the string in which each character has been mapped through the given translation table. :param string s: String to-be-character-table. :param dict table: Translation table.

static urlsafe_b64decode(*payload*)

Decode bytes-like object or ASCII string using the URL and filesystem-safe alphabet :param bytes payload: bytes-like object or ASCII string

static urlsafe_b64encode(*payload*)

Encode bytes-like object using the URL- and filesystem-safe alphabet, which substitutes - instead of + and _ instead of / in the standard Base64 alphabet, and return the encoded bytes. :param bytes payload: bytes-like object.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index