

JWT For Kali

To create a JSON Web Token (JWT) for Salesforce on **Kali Linux**, you can follow the steps below. Kali Linux is a Debian-based Linux distribution, so the process is similar to other Linux environments. You will use tools like **OpenSSL** for key generation and a programming language (e.g., Python) to generate the JWT.

Step 1: Install Required Tools

Ensure you have the necessary tools installed on Kali Linux:

1. **OpenSSL:**

- OpenSSL is pre-installed on Kali Linux. Verify by running:

```
>>> openssl version
```

■

- If not installed, use:

```
>>> sudo apt update
sudo apt install openssl
```

■

2. **Python:**

- Python is pre-installed on Kali Linux. Verify by running:

```
>>> python3 --version
```

■

- If not installed, use:

```
>>> sudo apt update
sudo apt install python3
```

■

3. **Python Libraries:**

- Install the `PyJWT` library for generating JWTs:

```
>>>
```

```
pip3 install pyjwt
```

■

Step 2: Generate a Private-Public Key Pair

Use OpenSSL to generate the private and public keys:

1. Generate Private Key:

```
>>>
```

```
openssl genrsa -out private.key 2048
```

■

2. Generate Public Key:

```
>>>
```

```
openssl rsa -in private.key -pubout -out public.key
```

■

3. Verify Keys:

- View the private key:

```
>>>
```

```
cat private.key
```

■

- View the public key:

```
>>>
```

```
cat public.key
```

■

4. Upload Public Key to Salesforce:

- Log in to Salesforce and navigate to **Setup > App Manager >** Select your Connected App > **Edit Policies >** Upload the public key under **Certificate and Key Management**.

Step 3: Create the JWT Using Python

Use Python to generate the JWT. Below is the Python script:

Python Script (generate_jwt.py)

python code block:

```
import jwt
import time

# Define the private key
private_key = """
-----BEGIN RSA PRIVATE KEY-----
YOUR_PRIVATE_KEY_HERE
-----END RSA PRIVATE KEY-----
"""

# Define the JWT payload
payload = {
    "iss": "YOUR_CONSUMER_KEY", # Consumer Key from Salesforce Connected App
    "sub": "YOUR_SALESFORCE_USERNAME", # Salesforce username
    "aud": "https://login.salesforce.com", # Use https://test.salesforce.com for sandbox
    "exp": int(time.time()) + 300 # Token expiration time (5 minutes from now)
}

# Generate the JWT
token = jwt.encode(payload, private_key, algorithm="RS256")
print("Generated JWT:")
print(token)
```

Steps to Run the Script

1. Save the script as generate_jwt.py.
2. Run the script:

```
python3 generate_jwt.py
```

3. The script will output the JWT token.

Step 4: Use the JWT to Obtain an Access Token

Send the JWT to Salesforce using `curl` to obtain an access token.

Example Command



```
curl -X POST https://login.salesforce.com/services/oauth2/token \  
  -d "grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer" \  
  -d "assertion=YOUR_JWT"
```



Replace `YOUR_JWT` with the JWT generated in the previous step.

Example Response

json code block:



```
{  
  "access_token": "00Dxx00000000000!AQEAQI...",  
  "instance_url": "https://yourInstance.salesforce.com",  
  "id": "https://login.salesforce.com/id/00Dxx00000000000/005xx000001Sv6e",  
  "token_type": "Bearer",  
  "issued_at": "1693142400",  
  "signature": "abcdef123456..."  
}
```



Step 5: Use the Access Token

Use the `access_token` to make authenticated API requests to Salesforce.

Example API Request



```
curl -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \  
  https://yourInstance.salesforce.com/services/data/v57.0/subjects/Account
```



Troubleshooting

1. **Invalid Grant Error:**

- Ensure the `sub` field matches the Salesforce username.
- Ensure the `aud` field matches the correct Salesforce environment (`login.salesforce.com` or `test.salesforce.com`).

2. **Expired Token:**

- Ensure the `exp` field is set to a future time (e.g., 5 minutes from now).

3. **Invalid Signature:**

- Ensure the private key matches the public key uploaded to Salesforce.

4. **Debugging:**

- Use verbose mode in `curl` to debug:

```
>>> curl -v -X POST https://login.salesforce.com/services/oauth2/token \
  -d "grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer" \
  -d "assertion=YOUR_JWT"
```

■

Additional Resources

- [Salesforce JWT Bearer Token Flow\(external, opens in a new tab or window\)](#)
- [PyJWT Documentation\(external, opens in a new tab or window\)](#)
- [OpenSSL Documentation\(external, opens in a new tab or window\)](#)

Let me know if you need further assistance!

Revision #2

Created 29 August 2025 12:53:29 by Richmond Abella

Updated 19 November 2025 08:05:19