

[Home](#) / [Resources](#) / JWT Auth Playground

JWT Auth Payload Verifier

APEX Cloud supports **JWT Auth** to secure APIs transactions. Understand the process of creating hashed API payloads to be used in our JWT Auth

1. Pick the Content-Type of the payload you want to send

Welcome to the JWT Auth Payload Verifier Playground! To get started, select the content-type of the JWT payload you'd like to try out. Different content-types represent various data formats that may be used within JWTs. Choose the one that aligns with your use case.

Suggestion:

JWT Auth Payload Verifier

Welcome to the JWT Auth Payload Verifier Playground!

APEX Cloud supports JWT Auth to secure APIs transactions. Explore this playground to learn how to create hashed API payloads for our JWT authentication process.

1. Select the content type of the payload you want to send

Different content-types represent various data formats that may be used within JWTs. Choose the one that aligns with your use case.

2. Enter the contents of your JWT payload

Enter the contents of your JWT payload below to start the verification process. Upon entering your JWT payload, the expected payload will be generated, which is the properly formatted version of your payload according to our JWT auth policy specifications.

Note: The displayed expected payload may be broken up into multiple lines for readability purposes, even if the actual payload is a single line.

Payload verifications for application/json

1. The payload must be a valid JSON string
2. The payload must not have leading whitespace
3. The payload must not have trailing whitespace
4. The payload must not have Carriage Returns (\r) or Linefeeds (\n)
5. The payload must not have whitespace between any of the structural characters of JSON

Your Payload JSON

View the [sample code snippet](#) to learn more about formatting application/json payloads for hashing

1

2. Add your JWT payload

To start the verification process, add your JWT payload below. The expected payload will be automatically generated. This payload is formatted according to our JWT authentication policy.

Note: The generated expected payload may be broken up into multiple lines for readability purposes even if the actual payload is a single line.

Payload verifications for application/json:

- The payload must be a valid JSON string.
- The payload must not have leading whitespace.
- The payload must not have trailing whitespace.
- The payload must not have Carriage Returns (\r) or Linefeeds (\n).
- The payload must not have whitespace between any of the structural characters of JSON.

Your Payload

View the [sample code snippet](#) to learn more about formatting application/json payloads for hashing.



Payload must not contain Carriage Return(\r) or Newline(\n) characters

For more information, visit the [API Payload Hash](#)  of our JWT guide

Payload must not contain Carriage Return(\r) or Newline(\n) characters

For more information, refer to [API Payload Hash](#) in our JWT guide.

3. Enter the SHA256 hash of your payload

Next, calculate the SHA256 hash of your entered payload and input it below. The expected hash of **the payload you entered** will be generated for you, which you can use to verify against your own hash.

Your Payload Hash

Expected Payload Hash

 Copy


ecca23232a06c6d13a4d6149f8d6b9334e6130d4e02af4a0805030751311774a

Next, calculate the SHA256 hash of your payload and enter it below.

We'll generate the expected hash for you to compare with your own.

(Optional) Try out the JWT Auth Flow

This is the end of the guide on generating a JWT payload hash. If you need more information on how our JWT auth flow works, please visit our [JWT Auth Playground](#).

You can also try out our [Hello World! APIs](#)  to test out your JWT auth flow with an actual application.

(Optional) Try out the JWT Auth Flow

This concludes the JWT Auth Payload Verifier guide. For more details on how our JWT auth flow works, visit our [JWT Auth Playground](#). You can also try out our [Hello World! APIs](#) to apply the JWT auth flow with an actual application.