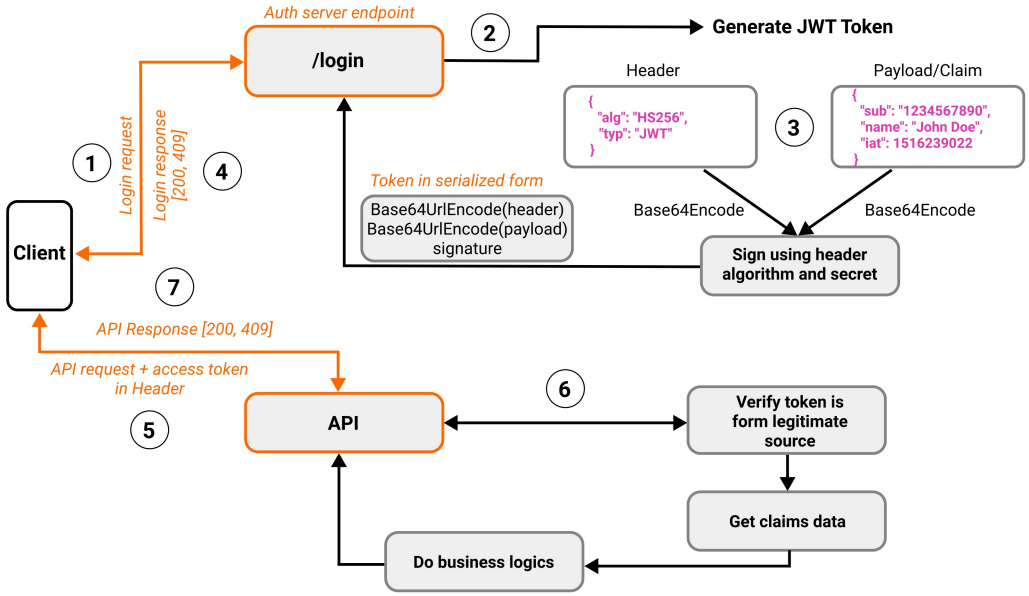


FIGURE 2.6: JWT Authorization concept diagram.

By steps, the process is

- **Step 1.** *Client* application sends authentication request to the *Auth server endpoint* provided user credentials in request body.
- **Step 2.** *Auth server endpoint* responses to the *Client* with the following HTTP response codes:
  - `409CONFLICT`: Invalid credentials.
  - `200SUCCESS`: Returns a pair of access and refresh tokens.
    - **Step 3.** *Auth server* generates a pair of access and refresh tokens
      - *Auth server* fetches user data and claims.
      - *Auth server* creates new session instance in database.
      - *Auth server* Base64 encodes access token's Header.
      - *Auth server* Base64 encodes access token's Payload.
      - *Auth server* generates access token's Signature using encoded token's Header and Payload signed by means of the `HMACSHA256` algorithm and secret.

- **Step 4.** JWT access token in serialized form and refresh token in form of GUID are returned in response with `200SUCCESS` http status code to the *Client* from the *Auth server*.
- **Step 5.** *Client* queries the *API* providing access token as Bearer in request header.
- **Step 6.** *API* validates the token claims in order to authorize user
  - If authorized: *API* handles the request, goes to **Step 7**.
  - Otherwise: returns error with `401UNAUTHORIZED` http status code.
- **Step 7.** returns response with `200SUCCESS` or `409CONFLICT` http status codes to the client, according to business logic layer implementation.