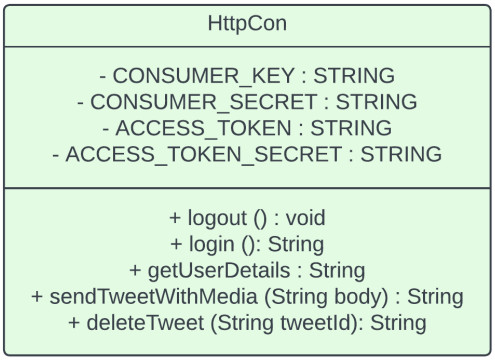
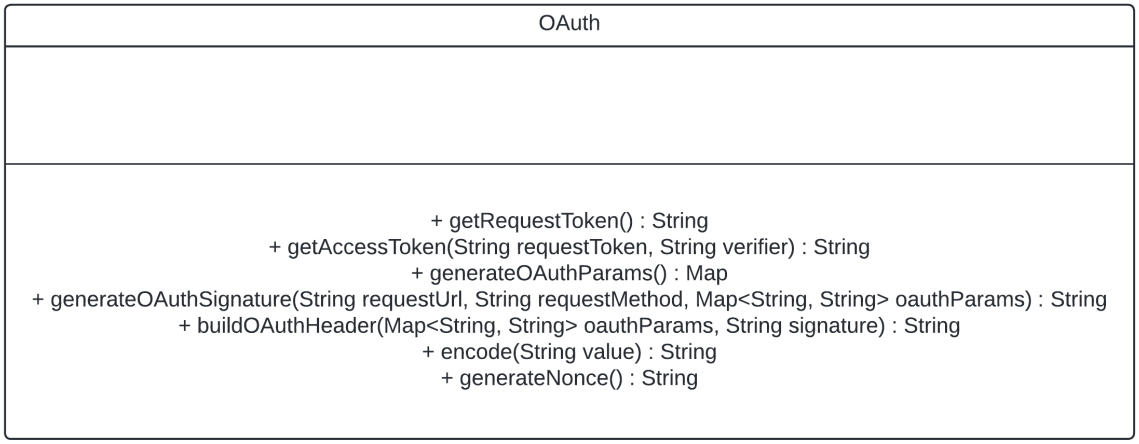
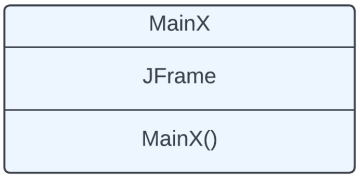
PBO

TWITTER / X API Oauth 1.0a dengan Java

A11.2022.14795 - NUR IKHSANUDIN







Class Diagram

```
private static String CONSUMER_KEY  = "CONSUMER_KEY";
private static String CONSUMER_SECRET  = "CONSUMER_SECRET";
private static String ACCESS_TOKEN  = "ACCESS_TOKEN";
private static String ACCESS_TOKEN_SECRET  = "ACCESS_TOKEN_SE
```

OAuth 1.0 credentials

Didapatkan dengan mendaftar ke https://developer.x.com/ kemudian melakukan regenerate untuk Keys dan Token.

```
 private Map<String, String> generateOAuthParams() {
        Map<String, String> params = new HashMap<>();
        params.put("oauth_consumer_key", CONSUMER_KEY);
        params.put("oauth_token", ACCESS_TOKEN);
        params.put("oauth_signature_method", "HMAC-SHA1");
        params.put("oauth_timestamp", String.valueOf(System.
        params.put("oauth_nonce", generateNonce());
        params.put("oauth_version", "1.0");
        return params;
    }

 private static String generateNonce() {
        String characters = "abcdefghijklmnopqrstuvwxyzABCDE
        Random rand = new Random();
        StringBuilder nonceBuilder = new StringBuilder();
        for (int i = 0; i < 32; i++) {
            nonceBuilder.append(characters.charAt(rand.nextI
        }
        return nonceBuilder.toString();
    }
```

Meng-Generate parameter OAuth yang diperlukan untuk otentikasi. Parameter-parameter ini mencakup consumer key, access token, signature method, timestamp, nonce, and version.

```
private String generateOAuthSignature(String requestUrl,
String requestMethod, Map<String, String> oauthParams) throws
        // Combine the request parameters and OAuth paramete
```

```
        Map<String, String> allParams = new HashMap<>(oauthP

        // Sort the parameters alphabetically by key
        String[] sortedKeys = allParams.keySet().toArray(new
        Arrays.sort(sortedKeys);

        // Construct the parameter string
        StringBuilder paramBuilder = new StringBuilder();
        for (String key : sortedKeys) {
            if (paramBuilder.length() > 0) {
                paramBuilder.append("&");
            }
            paramBuilder.append(key).append("=").append(allP
        }
        String parameterString = paramBuilder.toString();

        // Construct the base string
        String baseString = requestMethod + "&" + encode(req
        + "&" + encode(parameterString);

        // Construct the signing key
        String signingKey = encode(CONSUMER_SECRET) + "&" + (

        // Generate the HMAC-SHA1 signature
        Mac mac = Mac.getInstance("HmacSHA1");
        SecretKey secretKey = new SecretKeySpec(signingKey.g
        mac.init(secretKey);
        byte[] baseStringBytes = baseString.getBytes("UTF-8"
        byte[] signatureBytes = mac.doFinal(baseStringBytes)
        String signature = new String(Base64.getEncoder().en
        return signature;
    }
```

Generate OAuth Signature

Menggunakan metode generateOAuthSignature. Ini melibatkan penggabungan request parameters and OAuth parameters, mengurutkannya secara alfabet, membangun base string, dan menghasilkan signature HMAC-SHA1.

```
private String buildOAuthHeader(Map<String, String> oauthPara
        StringBuilder headerBuilder = new StringBuilder();
        headerBuilder.append("OAuth ");

        List<String> encodedParams = new ArrayList<>();

        encodedParams.add("oauth_consumer_key=\""
        + encode(oauthParams.get("oauth_consumer_key")) + "\
        encodedParams.add("oauth_token=\""
        + encode(oauthParams.get("oauth_token")) + "\"");
        encodedParams.add("oauth_signature_method=\""
        + encode(oauthParams.get("oauth_signature_method"))
        encodedParams.add("oauth_timestamp=\""
        + encode(oauthParams.get("oauth_timestamp")) + "\"")
        encodedParams.add("oauth_nonce=\""
        + encode(oauthParams.get("oauth_nonce")) + "\"");
        encodedParams.add("oauth_version=\""
        + encode(oauthParams.get("oauth_version")) + "\"");
        encodedParams.add("oauth_signature=\""
        + encode(signature) + "\"");

        String header = String.join(", ", encodedParams);
        headerBuilder.append(header);

        return headerBuilder.toString();
    }

private String encode(String value) {
        try {
            return URLEncoder.encode(value, StandardCharsets
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("Failed to encode par
        }
    }
```

Membuat OAuth Authorization Header

Dengan parameter OAuth dan signature, kemudina membangun header otorisasi OAuth menggunakan metode buildOAuthHeader. Header ini mencakup semua parameter OAuth yang telah dienkripsi dan generated signature.

Contoh Header :

```
 --header 'authorization: OAuth oauth_consumer_key="CONSUMER_
  oauth_nonce="OAUTH_NONCE",
  oauth_signature="OAUTH_SIGNATURE",
  oauth_signature_method="HMAC-SHA1",
  oauth_timestamp="OAUTH_TIMESTAMP", oauth_token="ACCESS_TOKE
  oauth_version="1.0"' \
```

Send Tweet

```
public static void sendTweet(String body) throws IOException
        String tweetUrl = "https://api.twitter.com/2/tweets";

        // Generate the OAuth parameters
        Map<String, String> oauthParams = OAuth.generateOAuth

        // Generate the OAuth signature
        String signature = OAuth.generateOAuthSignature(tweet

        // Build the OAuth authorization header
        String oauthHeader = OAuth.buildOAuthHeader(oauthPara

        // System.out.println(oauthHeader);

                // Membuat HTTP Request
        URL url = new URL(tweetUrl);
        HttpURLConnection connection = (HttpURLConnection) ur
        connection.setRequestMethod("POST");
        connection.setRequestProperty("Authorization", oauthH
        connection.setRequestProperty("Content-Type", "applic
        connection.setDoOutput(true);
```

```
        OutputStreamWriter writer = new OutputStreamWriter(co
        writer.write(body);
        writer.flush();

        int responseCode = connection.getResponseCode();
        System.out.println("Response code: " + responseCode);


        writer.close();
        connection.disconnect();
    }
```

Set up HTTP Connection

Twitter API endpoint : https://api.twitter.com/2/tweets digunakan untuk request upload status ke X

Mengatur metode request sebagai "POST", dan menambahkan header otorisasi OAuth serta jenis konten yang akan dikirim, dalam hal ini adalah JSON.
Data Tweet yang ingin dikirimkan diatur dalam variabel body dan dikirimkan melalui koneksi yang telah dibuka.
Menghandle respons dari server Twitter untuk mengetahui apakah Tweet berhasil dikirim atau tidak. Jika mendapatkan kode kurang dari 200 maka request berhasil.

Login

```
    public static String login() throws IOException {
        String authorizationUrl = "https://api.twitter.com/oa
        String requestToken = OAuth.getRequestToken();

        String authorizationUrlWithToken = authorizationUrl +
        System.out.println("Buka Url dan beri Akses Aplikasi:
        System.out.println(authorizationUrlWithToken);

        System.out.println("Access token: " + ACCESS_TOKEN);
        System.out.println("Access token secret: " + ACCESS_T
```

```
        return authorizationUrl + "?oauth_token=" + requestTo
    }
```

Twitter API endpoint : https://api.twitter.com/oauth/authorize digunakan untuk agar user dapat mengakses dan memberikan izin aplikasi.

Fungsi ini mengembalikan URL otorisasi beserta dengan request token.

Delete Tweet by ID

```
    public static void deleteTweet(String tweetId) throws IOE
        String apiUrl = "https://api.twitter.com/2/tweets/" +

        // Generate the OAuth parameters
        Map<String, String> oauthParams = OAuth.generateOAuth

        // Generate the OAuth signature
        String signature = OAuth.generateOAuthSignature(apiUr

        // Build the OAuth authorization header
        String oauthHeader = OAuth.buildOAuthHeader(oauthPara

        // Make the DELETE request to delete the tweet
        URL url = new URL(apiUrl);
        HttpURLConnection connection = (HttpURLConnection) ur
        connection.setRequestMethod("DELETE");
        connection.setRequestProperty("Authorization", oauthH

        // Handle the response
        int responseCode = connection.getResponseCode();
        System.out.println("Response code: " + responseCode);


        connection.disconnect();
    }
```

Twitter API endpoint : https://api.twitter.com/2/tweets/ digunakan untuk menghapus tweet tertentu berdasarkan ID yang diberikan.

Mengatur metode request sebagai "DELETE".

Get User ME

```
public static String getUserDetails() throws IOException {
        String apiUrl = "https://api.twitter.com/2/users/me";

        // Generate the OAuth parameters
        Map<String, String> oauthParams = OAuth.generateOAuth

        // Generate the OAuth signature
        String signature = OAuth.generateOAuthSignature(apiUr

        // Build the OAuth authorization header
        String oauthHeader = OAuth.buildOAuthHeader(oauthPara

        // Make the request
        URL url = new URL(apiUrl);
        HttpURLConnection connection = (HttpURLConnection) ur
        connection.setRequestMethod("GET");
        connection.setRequestProperty("Authorization", oauthH

        // Handle response
        int responseCode = connection.getResponseCode();
        if (responseCode < 299) {
            BufferedReader reader = new BufferedReader
            (new InputStreamReader(connection.getInputStream(
            StringBuilder response = new StringBuilder();
            String line;
            while ((line = reader.readLine()) != null) {
                response.append(line);
            }
            reader.close();

            System.out.println(responseCode);
            System.out.println("User details:");
            System.out.println(response);
```

```
            return response.toString();
        } else {
            return "Gagal mendapatkan detail user. Response c
        }
    }
```

Twitter API endpoint : https://api.twitter.com/2/users/me digunakan untuk mendapatkan profile dari user yang telah login.
Data yang diterima dari API dibaca baris demi baris menggunakan BufferedReader dan disusun menjadi satu string menggunakan StringBuilder.