

Forum:[Hack Forums](#)

Thread: [Twitter.com / X.com Session / Cookie Extraction](#)

Clumsy - March 9, 2026, 9:31 am

? X.com Session Token Extractor

Extract your own session tokens via browser console

Found by: SleepTheGod / ClumsyLulz

? Step-by-Step Instructions

Login to your Twitter account from <https://x.com/login> or <https://twitter.com/login>

After logging in, right-click anywhere and select Inspect (or press F12) to open DevTools

In DevTools, click the Network tab and refresh the page (F5)

Go to the Console tab and run the following code

? Export Your Tokens

Code // Run this in browser console to export YOUR cookies

```
function exportCookies() {  
  const cookies = document.cookie.split(';').map(cookie => {  
    const [name, value] = cookie.trim().split('=');  
    return { name, value };  
  });  
  
  console.log('Your cookies:', cookies);  
  console.log('Copy this object to save your cookies:', JSON.stringify(cookies));  
  
  // Also show localStorage items (might contain tokens)  
  console.log('localStorage items:');  
  for (let i = 0; i < localStorage.length; i++) {  
    const key = localStorage.key(i);  
    if (key.includes('auth') || key.includes('token') || key.includes('session')) {  
      console.log(key, ':', localStorage.getItem(key));  
    }  
  }  
  
  return cookies;  
}  
  
exportCookies();
```

Your cookies should look like this (values redacted):

Code Your cookies: (10) items

```
__cuid: REDACTED
personalization_id: REDACTED
guest_id: REDACTED
g_state: REDACTED
guest_id_ads: REDACTED
guest_id_marketing: REDACTED
twid: REDACTED (was u%3D1894924008929533957)
ct0: REDACTED (long hex string)
external_referer: REDACTED
lang: en
```

```
[
  {"name": "__cuid", "value": "REDACTED"},
  {"name": "personalization_id", "value": "REDACTED"},
  {"name": "guest_id", "value": "REDACTED"},
  {"name": "g_state", "value": "REDACTED"},
  {"name": "guest_id_ads", "value": "REDACTED"},
  {"name": "guest_id_marketing", "value": "REDACTED"},
  {"name": "twid", "value": "REDACTED"},
  {"name": "ct0", "value": "REDACTED"},
  {"name": "external_referer", "value": "REDACTED"},
  {"name": "lang", "value": "en"}
]
```

Redacted localStorage

xchat_user[REDACTED]_last-jwt-connection-token -> REDACTED (was the eyJhbGciOiJIUzI1NiJ9... JWT)

Clean, safe, no leaks.

If you want to verify the current logged-in user without exposing anything, run this updated safe fetch (it uses a public GraphQL query + credentials: 'include' so it pulls your context if logged in)

? Check Your Session Status

Paste this into DevTools console:

```
Code // Check if you're actually logged in by examining the page
console.log("=== SESSION VERIFICATION ===");
```

```
// Check for logged-in indicators
```

```
const isLoggedIn = !!document.querySelector('[data-testid="SideNav_AccountSwitcher_Button"]');
console.log("Is logged in (by UI):", isLoggedIn);
```

```
// Get your user ID from twid cookie
```

```
const twidMatch = document.cookie.match(/twid=u%3D(\d+)/);
const userId = twidMatch ? twidMatch[1] : null;
console.log("Your User ID:", userId);
```

```
// Check auth token from localStorage
```

```
const authToken = localStorage.getItem('xchat_user' + userId + '_last-jwt-connection-token');
```

```

console.log("Auth token exists:", !!authToken);
if (authToken) {
  console.log("Token preview:", authToken.substring(0, 30) + "...");
}

```

? Make a Working API Call to Get Your Profile

Code // This endpoint should work - get your own user profile

```

async function getMyProfile() {
  const userId = document.cookie.match(/twid=u%3D(\d+)/)?.[1];
  if (!userId) {
    console.log("Not logged in!");
    return;
  }

  const csrfToken = document.cookie.match(/ct0=([^\;]+)/)?.[1];

  try {
    const response = await
    fetch(`https://x.com/i/api/graphql/4S2ihlK1bQtxVgfCJgY89Q/UserByScreenName?variables=${encodeURIComponent(JSON.stringify({
      screen_name:
document.querySelector("[data-testid='SideNav_AccountSwitcher_Button']")?.innerText?.trim() ||
"your_username",
      withSafetyModeUserFields: true
    })))`, {
      headers: {
        'authorization': 'Bearer REDACTED_BEARER_TOKEN',
        'x-csrf-token': csrfToken,
        'content-type': 'application/json'
      },
      credentials: 'include'
    });

    const data = await response.json();
    console.log("Profile data:", data);
    return data;
  } catch (e) {
    console.error("Fetch failed:", e);
  }
}

getMyProfile();

```

? Get Your Timeline Tweets

Code // Fetch your home timeline

```

async function getHomeTimeline() {
  const csrfToken = document.cookie.match(/ct0=([^\;]+)/)?.[1];

  try {

```

```

const response = await
fetch('https://x.com/i/api/graphql/8simZCWKqXz6R0nPzBvtSw/HomeTimeline', {
  method: 'POST',
  headers: {
    'authorization': 'Bearer REDACTED_BEARER_TOKEN',
    'x-csrf-token': csrfToken,
    'content-type': 'application/json'
  },
  body: JSON.stringify({
    variables: {
      count: 20,
      includePromotedContent: true,
      withCommunity: true,
      withSuperFollows: true,
      withBirdwatchPivots: true
    },
    features: {
      responsive_web_graphql_exclude_directive_enabled: true,
      verified_phone_label_enabled: false,
      responsive_web_graphql_skip_user_profile_image_extensions_enabled: false,
      responsive_web_graphql_timeline_navigation_enabled: true
    }
  }),
  credentials: 'include'
});

const data = await response.json();
console.log("Timeline data:", data);
return data;
} catch (e) {
  console.error("Timeline fetch failed:", e);
}
}

```

getHomeTimeline();

i Simple Script to Get Your Account Info

Code // Get basic account info from the page
 console.log("=== YOUR ACCOUNT INFO ===");

```

// Try to find username from various places
const usernameElement =
document.querySelector("[data-testid='SideNav_AccountSwitcher_Button'] div[dir='ltr']");
const username = usernameElement ? usernameElement.innerText : "Not found";

```

```

// Get display name
const displayNameElement =
document.querySelector("[data-testid='SideNav_AccountSwitcher_Button'] div[dir='auto']");
const displayName = displayNameElement ? displayNameElement.innerText : "Not found";

```

console.log("Username:", username);

```

console.log("Display name:", displayName);
console.log("User ID:", document.cookie.match(/twid=u%3D(\d+)/)?.[1] || "Not found");

// Get follower count if available
const followersElement = document.querySelector('a[href$="/verified_followers"] span span');
console.log("Followers:", followersElement ? followersElement.innerText : "Not on profile page");

```

? Working GraphQL Queries (Most Reliable)

Code // Helper function for GraphQL queries

```

async function graphqlQuery(queryName, variables = {}) {
  const csrfToken = document.cookie.match(/ct0=(\^;+)?\.[1];

  // Common GraphQL query IDs (these might change)
  const queryIds = {
    'UserByScreenName': 'REDACTED',
    'HomeTimeline': 'REDACTED',
    'UserTweets': 'REDACTED',
    'UserByRestId': 'REDACTED'
  };

  const queryId = queryIds[queryName];
  if (!queryId) {
    console.log("Unknown query:", queryName);
    return;
  }

  try {
    const response = await fetch(`https://x.com/i/api/graphql/${queryId}/${queryName}`, {
      method: 'POST',
      headers: {
        'authorization': 'Bearer REDACTED_BEARER_TOKEN',
        'x-csrf-token': csrfToken,
        'content-type': 'application/json'
      },
      body: JSON.stringify({
        variables: variables,
        features: {
          responsive_web_graphql_exclude_directive_enabled: true,
          verified_phone_label_enabled: false,
          responsive_web_graphql_skip_user_profile_image_extensions_enabled: false,
          responsive_web_graphql_timeline_navigation_enabled: true
        }
      }),
      credentials: 'include'
    });

    return await response.json();
  } catch (e) {
    console.error("GraphQL error:", e);
  }
}

```

```
// Usage example - get user by screen name
graphqlQuery('UserByScreenName', {
  screen_name: username || "your_username",
  withSafetyModeUserFields: true
}).then(data => console.log("User data:", data));
```

? Save Your Working Session for Later

Run this in the Dev Console:

Code // Export your session data properly formatted for Node.js

```
const sessionData = {
  cookies: document.cookie.split(';').map(cookie => {
    const [name, value] = cookie.trim().split('=');
    return { name, value };
  }),
  localStorage: {
    'last-jwt-connection-token': localStorage.getItem('xchat_user' +
      document.cookie.match(/twid=u%3D(\d+)/)?.[1] + '_last-jwt-connection-token'),
    // Add any other important tokens
  },
  csrfToken: document.cookie.match(/ct0=([^\;]+)/)?.[1],
  userId: document.cookie.match(/twid=u%3D(\d+)/)?.[1]
};
```

```
console.log("Copy this for your Node.js script:");
console.log(JSON.stringify(sessionData, null, 2));
```

? Important Notes

Old REST endpoints (/1.1/) often return 404 - use GraphQL endpoints instead

GraphQL endpoints are more reliable (/i/api/graphql/)

You need the CSRF token (ct0 cookie) for POST requests

The bearer token (AAAAAAAAAAAAAAAAAAAAAAAAANRILg...) is public - you can find it in the Network tab

This is for YOUR ACCOUNT ONLY - using on others' accounts is illegal

Keep your tokens PRIVATE - never share them with anyone

?

Found by: SleepTheGod / ClumsyLulz

Use responsibly • For educational purposes only