SSO Features

SSO Integration with Forma Cloud

1. What is Forma Cloud SSO?

The FormaLMS SSO system allows the generation of automatic login links that authenticate the user securely through a temporary cryptographic token. This is a common solution for:

- Automatic redirect from a web application
- SSO login links in HTML pages
- API endpoint returning an SSO URL

This guide explains how to correctly implement Single Sign-On (SSO) authentication with FormaLMS, allowing users to access the platform directly from external systems without entering their credentials.

---

2. How SSO Works

The SSO login process is based on three elements:

1. **login_user** – The username of the user who must be authenticated
2. **time** – A Unix timestamp representing when the link was generated
3. **token** – An MD5 hash that validates the authenticity of the request

The token is calculated using the username, timestamp, and a shared secret key (SSO Secret).

---

3. SSO URL Structure

A valid SSO URL follows this structure:

```
https://PLATFORM/index.php?r=adm/homepage/sso&login_user=USERNAME&time=TIMESTAMP&token=TOKEN
```

Parameters

- **r=adm/homepage/sso** – FormaLMS SSO endpoint
- **login_user** – User's username (URL-encoded)
- **time** – Current Unix timestamp
- **token** – Uppercase MD5 hash

Example

```
https://forma.example.com/index.php?r=adm/homepage/sso&login_user=mario.rossi&time=1729681425&token=A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6
```

---

4. Generating the SSO Token

Step 1: Generate the Unix Timestamp

The Unix timestamp represents the number of seconds elapsed since Jan 1st, 1970 (UTC).

Examples:

- **PHP:** `time()`
- **Node.js:** `Math.floor(Date.now() / 1000)`
- **Python:** `int(time.time())`
- **C# (.NET):** `DateTimeOffset.UtcNow.ToUnixTimeSeconds()`
- **Java**: System.currentTimeMillis() / 1000

> **Important:** The timestamp must be generated dynamically at the moment of the SSO request.

Step 2: Calculate the Token

The token is an **uppercase MD5 hash** of a specific concatenated string:

```
token = MD5_UPPERCASE(login_user + "," + time + "," + sso_secret)
```

Where:

- **login_user** = the username

- **time** = timestamp from Step 1
- **sso_secret** = secret key configured in FormaLMS

Example (PHP)

```
$loginUser = "mario.rossi";
$time = 1729681425;
$ssoSecret = "mia_chiave_segreta_123";
// Build the string
$stringToHash = $loginUser . ',' . $time . ',' . $ssoSecret;
// Result: "mario.rossi,1729681425,mia_chiave_segreta_123"
// Calculate MD5 and convert to UPPERCASE
$token = strtoupper(md5($stringToHash));
// result: "A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6"
```

> **Requirements**:
> - Commas must match exactly
> - No extra spaces
> - Final hash **must be uppercase**

Step 3: Build the Final URL

Once the timestamp and token are ready, construct the final SSO URL:

Example (PHP):

```
$platformUrl = "https://forma.example.com";

$ssoUrl = $platformUrl . "/index.php"
    . "?r=adm/homepage/sso"
    . "&login_user=" . urlencode($loginUser)
    . "&time=" . $time
    . "&token=" . $token;
```

---

5. Configuration in FormaLMS

Before using SSO, configure the secret key inside FormaLMS:

1. Log in as admin

2. Navigate to **Administration → Configuration → SSO**
3. Enter the **SSO Secret Key** (must match the one used in your code)
4. Save settings

---

6. Link Validity

SSO links are time-limited for security reasons.

Characteristics:

- Typical validity: **5–10 minutes**
- The link must be used immediately
- Past or future timestamps are not accepted
- Expired links require regeneration

Best practices:

- Generate the link just before redirecting
- Do not store SSO URLs for future use
- Implement automatic regeneration if needed

---

7. Implementation Checklist

Configuration

- Secret key configured in FormaLMS
- Secret matches the one used in external system
- HTTPS enabled

Token Generation

- Timestamp is a Unix timestamp
- Timestamp generated per request
- String format: `username,time,secret` (commas required)
- MD5 hash converted to uppercase
- No extra spaces

URL

- `r=adm/homepage/sso` is present
- Username is URL-encoded

- All required parameters included

---

8. Testing and Verification

Manual Token Test

Verify the token by hashing a test string in the exact format:

```
username,timestamp,secret
```

Use any MD5 generator to confirm the result.

Access Test

1. Generate a valid SSO URL
2. Open it in a browser within 5 minutes
3. Confirm the user is logged in
4. Confirm redirection to the homepage

---

9. Troubleshooting

"Invalid token" or "Access denied"

Check:

- Matching SSO secret
- Uppercase MD5 token
- Correct format (username,timestamp,secret)
- Existing username in FormaLMS

"Link expired"

Check:

- Link used within validity window
- Server clocks synchronized (NTP recommended)
- Timestamp generated at request time

Redirects to login page

Check:

- Correct endpoint (`adm/homepage/sso`)
- Username properly URL-encoded
- All parameters present

---

10. Security

Protecting the SSO Secret

- Keep the secret strictly confidential
- Never store it in logs, public repos, or client-side code
- Use environment variables or secret-management systems
- Rotate the key every 6–12 months

Protect SSO Links

- Always use HTTPS
- Do not send SSO URLs through insecure channels
- Avoid logging full SSO URLs
- Implement rate limiting

Monitoring

- Monitor failed SSO attempts
- Implement alerts for suspicious activity
- Periodically audit SSO access logs

---

11. SSO Variants

FormaLMS supports two SSO endpoints:

1. Standard SSO (username-based)

```
?r=adm/homepage/sso&login_user=USERNAME&time=TIME&token=TOKEN
```

Token: `MD5(username,time,secret)`

2. Custom SSO (email-based)

```
?r=adm/ssologin/show&email=EMAIL&time=TIME&token=TOKEN
```

Token: `MD5(email,time,secret)`

Check which one your installation uses.

12. Full Implementation examples

PHP

```
<?php

/**
 * Generates a valid SSO URL for FormaLMS
 *
 * @param string $loginUser   The username of the user
 * @param string $ssoSecret   Secret key configured in FormaLMS
 * @param string $platformUrl Base platform URL
 * @return array              Generated SSO data
 */
function generateSSOUrl($loginUser, $ssoSecret, $platformUrl)
{
    // STEP 1: Generate current Unix timestamp
    $time = time();

    // STEP 2: Calculate the token
    $stringToHash = $loginUser . ',' . $time . ',' . $ssoSecret;
    $token = strtoupper(md5($stringToHash));

    // STEP 3: Build the URL
    $url = rtrim($platformUrl, '/') . "/index.php"
         . "?r=adm/homepage/sso"
         . "&login_user=" . urlencode($loginUser)
         . "&time=" . $time
         . "&token=" . $token;

    return [
        'url' => $url,
        'login_user' => $loginUser,
        'time' => $time,
```

```
        'token' => $token,
        'valid_until' => date('Y-m-d H:i:s', $time + 300)
    ];
}

// Example usage
$ssoData = generateSSOUrl(
    'mario.rossi',                    // Username
    'my_secret_key_123',              // SSO Secret
    'https://forma.example.com'       // Platform URL
);

echo "SSO URL: " . $ssoData['url'] . "\n";
echo "Valid until: " . $ssoData['valid_until'] . "\n";
?>
```

JavaScript (Node.js)

```
const crypto = require('crypto');

/**
 * Generates a valid SSO URL for FormaLMS
 */
function generateSSOUrl(loginUser, ssoSecret, platformUrl) {

    // STEP 1: Unix timestamp
    const time = Math.floor(Date.now() / 1000);

    // STEP 2: Calculate token
    const stringToHash = `${loginUser},${time},${ssoSecret}`;
    const token = crypto
        .createHash('md5')
        .update(stringToHash)
        .digest('hex')
        .toUpperCase();

    // STEP 3: Build URL
    const baseUrl = platformUrl.replace(/\/$/, '');
```

```
    const url = `${baseUrl}/index.php?r=adm/homepage/sso`
              + `&login_user=${encodeURIComponent(loginUser)}`
              + `&time=${time}`
              + `&token=${token}`;

    return {
        url: url,
        login_user: loginUser,
        time: time,
        token: token
    };
}

// Example usage
const ssoData = generateSSOUrl(
    'mario.rossi',
    'my_secret_key_123',
    'https://forma.example.com'
);

console.log('SSO URL:', ssoData.url);
console.log('Token:', ssoData.token);
```

Python

```
import hashlib
import time
from urllib.parse import quote

def generate_sso_url(login_user, sso_secret, platform_url):
    """
    Generates a valid SSO URL for FormaLMS
    """

    # STEP 1: Unix timestamp
    current_time = int(time.time())

    # STEP 2: Calculate token
```

```
    string_to_hash = f"{login_user},{current_time},{sso_secret}"
    token = hashlib.md5(string_to_hash.encode()).hexdigest().upper()

    # STEP 3: Build URL
    base_url = platform_url.rstrip('/')
    url = (f"{base_url}/index.php?r=adm/homepage/sso"
           f"&login_user={quote(login_user)}"
           f"&time={current_time}"
           f"&token={token}")

    return {
        'url': url,
        'login_user': login_user,
        'time': current_time,
        'token': token
    }

# Example usage
sso_data = generate_sso_url(
    'mario.rossi',
    'my_secret_key_123',
    'https://forma.example.com'
)

print('SSO URL:', sso_data['url'])
print('Token:', sso_data['token'])
```

C# (.NET)

```
using System;
using System.Security.Cryptography;
using System.Text;
using System.Web;

public class FormaSSO
{
    public static string GenerateSSOUrl(string loginUser, string ssoSecret, string
platformUrl)
```

```
    {
        // STEP 1: Unix timestamp
        long time = DateTimeOffset.UtcNow.ToUnixTimeSeconds();

        // STEP 2: Calculate token
        string stringToHash = $"{loginUser},{time},{ssoSecret}";
        using (MD5 md5 = MD5.Create())
        {
            byte[] hashBytes = md5.ComputeHash(Encoding.UTF8.GetBytes(stringToHash));
            string token = BitConverter.ToString(hashBytes).Replace("-", "").ToUpper();

            // STEP 3: Build URL
            string baseUrl = platformUrl.TrimEnd('/');
            string url = $"{baseUrl}/index.php?r=adm/homepage/sso"
                        + $"&login_user={HttpUtility.UrlEncode(loginUser)}"
                        + $"&time={time}"
                        + $"&token={token}";

            return url;
        }
    }
}

// Example usage
string ssoUrl = FormaSSO.GenerateSSOUrl(
    "mario.rossi",
    "my_secret_key_123",
    "https://forma.example.com"
);

Console.WriteLine($"SSO URL: {ssoUrl}");
```

Revision #6
Created 2025-07-01 08:46:05 UTC by Alberto Pastorelli
Updated 2025-11-17 16:02:09 UTC by Alberto Pastorelli