JavaScript Widget Integration Guide

Generated on October 31, 2026

Ticket System SSO Integration Guide

Overview

This guide explains how to integrate Single Sign-On (SSO) between your main website and the Central Tickets system, allowing users to seamlessly authenticate from the logout landing page.

How It Works

When a user clicks "Sign In with [Your Company]" on the logout landing page:

1. User is redirected to your website's SSO endpoint: `https://yourwebsite.com/ticket-sso`
2. Your website checks if the user is logged in
3. If logged in, your website calls the Ticket System API with user credentials
4. The API generates an encrypted authentication token
5. User is redirected back to the ticket system with the token
6. Ticket system validates the token and creates a session
7. User lands on their dashboard, fully authenticated

Required Parameters

The logout page passes these URL parameters to your SSO endpoint:

- `redirect_url`: Where to send the user after authentication (e.g., `https://ticket.yourcompany.com/yourcompany/dashboard`)

- `callback_api`: The API endpoint to call for authentication (e.g., `https://tickets.flare99.com/api/auth/redirect/yourcompany`)
- `tenant`: Your tenant slug (e.g., `yourcompany`)

Implementation Options

Option 1: Server-Side (PHP - Recommended)

Create a file at `https://yourwebsite.com/ticket-sso.php`:

```php
<?php
// Check if user is logged in
session_start();
if (!isset($_SESSION['user_id'])) {
    // Redirect to login with return URL
    $returnUrl = urlencode($_SERVER['REQUEST_URI']);
    header("Location: /login?return=" . $returnUrl);
    exit;
}

// Get user data from your session/database
$user = getUserById($_SESSION['user_id']); // Your user retrieval function

// Get parameters
$redirectUrl = $_GET['redirect_url'] ?? '';
$callbackApi = $_GET['callback_api'] ?? '';
$tenantSlug = $_GET['tenant'] ?? '';

// Validate parameters
if (empty($redirectUrl) || empty($callbackApi)) {
    die('Invalid SSO request');
}

// Call ticket system API
$userData = [
    'email' => $user['email'],
    'name' => $user['name'],
    'redirect_url' => $redirectUrl
];

$ch = curl_init($callbackApi);
curl_setopt_array($ch, [
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_POST => true,
    CURLOPT_HTTPHEADER => [
        'Content-Type: application/json',
        'Accept: application/json'
    ],
    CURLOPT_POSTFIELDS => json_encode($userData),
    CURLOPT_FOLLOWLOCATION => false,
    CURLOPT_HEADER => true
]);

$response = curl_exec($ch);
$httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);

// Extract Location header from response
if ($httpCode === 302) {
    preg_match('/Location: (.+)/', $response, $matches);
    if (isset($matches[1])) {
        $redirectTo = trim($matches[1]);
        curl_close($ch);
        header("Location: " . $redirectTo);
        exit;
    }
}

curl_close($ch);
die('Authentication failed');
```

```
?>
```

Option 2: Server-Side (Laravel)

Route (routes/web.php):

```
Route::get('/ticket-sso', [SSOController::class, 'handleTicketSSO'])
    ->middleware('auth')
    ->name('ticket.sso');
```

Controller (app/Http/Controllers/SSOController.php):

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Http;
use Illuminate\Support\Facades\Auth;

class SSOController extends Controller
{
    public function handleTicketSSO(Request $request)
    {
        // Validate user is logged in
        if (!Auth::check()) {
            return redirect('/login?return=' . urlencode($request->fullUrl()));
        }

        // Get parameters
        $redirectUrl = $request->query('redirect_url');
        $callbackApi = $request->query('callback_api');
        $tenantSlug = $request->query('tenant');

        // Validate parameters
        if (!$redirectUrl || !$callbackApi || !$tenantSlug) {
            abort(400, 'Invalid SSO request parameters');
        }

        // Get authenticated user
        $user = Auth::user();

        // Call ticket system API
        try {
            $response = Http::withHeaders([
                'Accept' => 'application/json',
                'Content-Type' => 'application/json',
            ])->post($callbackApi, [
                'email' => $user->email,
                'name' => $user->name,
                'redirect_url' => $redirectUrl,
            ]);

            // Check if response is a redirect
            if ($response->redirect()) {
                return redirect($response->header('Location'));
            }

            // If JSON response with redirect_url
            $data = $response->json();
            if (isset($data['redirect_url'])) {
                return redirect($data['redirect_url']);
            }

            abort(500, 'Invalid API response');

        } catch (\Exception $e) {
            \Log::error('Ticket SSO failed', [
                'error' => $e->getMessage(),
                'user_id' => $user->id,
            ]);
```

```
            return view('ticket-sso-error', [
                'message' => 'Unable to authenticate with ticket system. Please try again.'
            ]);
        }
    }
}
```

Option 3: Client-Side (JavaScript - Advanced)

Create a page at `https://yourwebsite.com/ticket-sso.html`:

```
<!DOCTYPE html>
<html>
<head>
    <title>Signing you in...</title>
    <style>
        body {
            font-family: system-ui;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
            background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
        }
        .container {
            background: white;
            padding: 2rem;
            border-radius: 1rem;
            text-align: center;
            box-shadow: 0 20px 60px rgba(0,0,0,0.3);
        }
        .spinner {
            border: 4px solid #f3f3f3;
            border-top: 4px solid #667eea;
            border-radius: 50%;
            width: 50px;
            height: 50px;
            animation: spin 1s linear infinite;
            margin: 0 auto 1rem;
        }
        @keyframes spin {
            0% { transform: rotate(0deg); }
            100% { transform: rotate(360deg); }
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="spinner"></div>
        <h1>Signing you in...</h1>
        <p id="status">Authenticating with ticket system...</p>
    </div>

    <script>
        // Get URL parameters
        const urlParams = new URLSearchParams(window.location.search);
        const redirectUrl = urlParams.get('redirect_url');
        const callbackApi = urlParams.get('callback_api');
        const tenantSlug = urlParams.get('tenant');

        // Get user data from your authentication system
        // This example assumes you have a function that returns logged-in user data
        async function getUserData() {
            // Replace with your actual user data retrieval
            const response = await fetch('/api/user', {
                credentials: 'include'
            });
            if (!response.ok) return null;
            return await response.json();
        }
```

```
        async function authenticate() {
            try {
                // Get current user
                const user = await getUserData();

                if (!user) {
                    // Not logged in - redirect to login
                    window.location.href = '/login?return=' + encodeURIComponent(window.locatio
                    return;
                }

                // Call ticket system API
                const response = await fetch(callbackApi, {
                    method: 'POST',
                    headers: {
                        'Content-Type': 'application/json',
                        'Accept': 'application/json'
                    },
                    body: JSON.stringify({
                        email: user.email,
                        name: user.name,
                        redirect_url: redirectUrl
                    })
                });

                if (response.redirected) {
                    window.location.href = response.url;
                } else {
                    const data = await response.json();
                    if (data.redirect_url) {
                        window.location.href = data.redirect_url;
                    } else {
                        throw new Error('No redirect URL received');
                    }
                }

            } catch (error) {
                console.error('SSO Error:', error);
                document.querySelector('.container').innerHTML = `
                    <h1 style="color: #e53e3e;">Authentication Failed</h1>
                    <p>Unable to sign you in. Please try again.</p>
                    <button onclick="history.back()" style="margin-top: 1rem; padding: 0.75rem
                        Go Back
                    </button>
                `;
            }
        }

        // Start authentication
        authenticate();
    </script>
</body>
</html>
```

Testing

1. **Test with logged-in user:**
    - Log in to your main website
    - Go to ticket system logout page
    - Click "Sign In with [Your Company]"
    - You should be redirected through your SSO page and land on the ticket dashboard

2. **Test with logged-out user:**
    - Log out from your main website
    - Go to ticket system logout page
    - Click "Sign In with [Your Company]"
    - You should be redirected to your website's login page

Troubleshooting

"Authentication Failed" Error

- Verify your SSO endpoint URL is correct (`https://yourwebsite.com/ticket-sso`)
- Check that the user is logged in on your main website
- Verify the API call to `/api/auth/redirect/{tenant}` is successful
- Check server logs for CURL errors or API failures

Redirect Loop

- Ensure your SSO page doesn't redirect back to itself
- Check that the API returns a proper Location header (302 redirect)
- Verify the encrypted token is being passed correctly

CORS Errors

- Use server-side implementation (Option 1 or 2) to avoid CORS issues
- Client-side fetch() cannot access cookies from different domains

Security Notes

- Always validate that the user is authenticated before making the API call
- Use HTTPS for all SSO endpoints
- The ticket system API uses encrypted tokens with 2-minute expiration
- Tokens are single-use and cannot be replayed

Support

For additional help, refer to:

- `TENANT_INTEGRATION_GUIDE.md` - Complete integration documentation
- `/integration` route - Integration hub with downloadable examples
- Example files in `tenant-examples/` directory