Permission Implementation Guide: Codes & Logs

Date: January 25, 2026
Focus: Order Codes/Serials, SMS Logs, Email Logs

Overview

Three sections need permission-based show/hide implementation:

1. **Order Details - Serial Codes Section** (inside order dashboard)
2. **SMS Logs Page** (entire page)
3. **Email Logs Page** (entire page)

1. Order Serial Codes Section

Location

`/frontend/src/Admin/Components/OrderDetailsPage/OrderDetailsPage.tsx`

Current Status

✅ Backend protected: `permission:view codes` (line 307 in admin routes)
❌ Frontend: Shows codes to everyone (lines 691-783)

Problem

The codes/serials section is visible to all admins regardless of permissions:

```
// Current code (lines 691-783)
{item?.codes && item.codes.length > 0 && (
  <div className="mt-4 space-y-3">
    <h4>{t("codes")}:</h4>
    <div className="grid grid-cols-1 gap-3">
      {item.codes.map((code, codeIndex) => (
        <div key={codeIndex}>
          {/* Serial number and code displayed */}
        </div>
      ))}
    </div>
    </div>
  )}
```

Required Permission

- Backend uses: `view codes`
- Frontend should check: `view codes`

Implementation

Step 1: Import hasPermission utility

```
// At top of file (around line 10)
import { hasPermission } from '@/Admin/Utils/HasPermission';
import { useAppSelector } from '@/Hooks/ReduxHooks/ReduxHooks';
```

Step 2: Get user permissions

```
// Inside OrderDetailsPage component (around line 20)
const userPermissions = useAppSelector(
  (state) => state.admin.admin?.permissions || []
);
```

Step 3: Check permission before rendering

```
// Replace lines 691-783 with:
{item?.codes && item.codes.length > 0 && hasPermission(userPermissions, 'view codes')
  <div className="mt-4 space-y-3">
    <h4 className={`text-sm font-semibold transition-colors duration-300 ${
      theme === "dark" ? "text-gray-300" : "text-slate-700"
    }`}>
      {t("codes")}:
    </h4>
    <div className="grid grid-cols-1 gap-3">
      {item.codes.map((code, codeIndex) => (
        // ... rest of code display logic
      ))}
    </div>
  </div>
)}
```

Complete Code Block:

```
// Line 691-783
{/* Codes Section - Only show if user has permission */}
```

```
{item?.codes &&
 item.codes.length > 0 &&
 hasPermission(userPermissions, 'view codes') && (
  <div className="mt-4 space-y-3">
    <h4
      className={`text-sm font-semibold transition-colors duration-300 ${
        theme === "dark" ? "text-gray-300" : "text-slate-700"
      }`}
    >
      {t("codes")}:
    </h4>
    <div className="grid grid-cols-1 gap-3">
      {item.codes.map((code, codeIndex) => (
        <div
          key={codeIndex}
          className={`rounded-xl p-4 border transition-colors duration-300 ${
            theme === "dark"
              ? "bg-gray-700/50 border-gray-600/50"
              : "bg-slate-50 border-slate-200"
          }`}
        >
          <div className="space-y-2">
            {code.serial_number && (
              <div className="flex items-center justify-between">
                <span
                  className={`text-sm font-medium transition-colors duration-300 ${
                    theme === "dark" ? "text-gray-400" : "text-slate-600"
                  }`}
                >
                  {t("serialNumber")}:
                </span>
                <span
                  className={`font-mono text-sm px-2 py-1 rounded transition-colors du
                    theme === "dark"
                      ? "bg-gray-600 text-gray-200"
                      : "bg-slate-100 text-slate-900"
                  }`}
                >
                  {code.serial_number}
                </span>
              </div>
            )}
            <div className="flex items-center justify-between">
              <span
                className={`text-sm font-medium transition-colors duration-300 ${
                  theme === "dark" ? "text-gray-400" : "text-slate-600"
                }`}
              >
                {t("code")}:
              </span>
              <span
                className={`font-mono text-sm px-2 py-1 rounded transition-colors dura
                  theme === "dark"
```

```
                      ? "bg-gray-600 text-gray-200"
                      : "bg-slate-100 text-slate-900"
                  }`}
                >
                  {code.serial_code}
                </span>
              </div>
              <div className="flex items-center justify-between">
                <span
                  className={`text-sm font-medium transition-colors duration-300 ${
                    theme === "dark" ? "text-gray-400" : "text-slate-600"
                  }`}
                >
                  {t("validTo")}:
                </span>
              </div>
            </div>
          </div>
        ))}
      </div>
    </div>
  )}
```

Result

- ✅ Users WITH `view codes` permission: See full serial codes section
- ❌ Users WITHOUT permission: Section completely hidden (as if it doesn't exist)

2. SMS Logs Page

Location

`/frontend/src/app/[locale]/(admin)/secureacwadash/sms-logs/page.tsx`

Current Status

✅ Backend protected: `permission:viewAny smsLog`
✅ Sidebar hidden: Uses `export data` permission
❌ Frontend page: No permission check - anyone can access via URL

Required Permission

- Backend uses: `viewAny smsLog`
- Sidebar shows if: `export data`
- **Frontend should use: `viewAny smsLog`** (match backend)

Implementation

Wrap entire component with AdminPermissionWrapper:

```
// File: /frontend/src/app/[locale]/(admin)/secureacwadash/sms-logs/page.tsx

'use client';

import { useState, useMemo } from 'react';
import { useLocale, useTranslations } from 'next-intl';
import { useAppSelector } from '@/Hooks/ReduxHooks/ReduxHooks';
import PageTitle from '@/Admin/common/PageTitle';
import SearchAndFilters from '@/Admin/common/SearchAndFilters';
import TableLoading from '@/Admin/common/TableLoading';
import Pagination from '@/Admin/common/Pagination';
import WebsiteLoading from '@/Components/UI/Loading/WebsiteLoading';
import { useSMSLogs } from '@/Admin/Hooks/SMSLogs/useSMSLogs';
import {
  SMSLogsStatisticsCards,
  SMSLogsTable,
} from '@/Admin/Components/SMSLogs';
import SmsTestForm from '@/Admin/Components/SMS/SmsTestForm';
import { FaTimes } from 'react-icons/fa';
import { RiRefreshLine } from 'react-icons/ri';
import AdminPermissionWrapper from '@/Admin/Components/HOC/AdminPermissionWrapper'; //

type FilterValue = string | number | undefined;
type FiltersState = Record<string, FilterValue>;

export default function SMSLogsPage() {
  // ... all existing code ...

  return (
    <AdminPermissionWrapper requiredPermission="viewAny smsLog">
      <main className={/* ... existing classes ... */}>
        {/* ... all existing page content ... */}
      </main>
    </AdminPermissionWrapper>
  );
}
```

Alternative: Use Multiple Permissions

If page should be visible to users with EITHER `viewAny smsLog` OR `export data` :

```
<AdminPermissionWrapper
  requiredPermissions={["viewAny smsLog", "export data"]}
  requireAllPermissions={false}  // ANY permission is enough
>
```

```
    {/* page content */}
  </AdminPermissionWrapper>
```

Result

- ✅ Users WITH `viewAny smsLog` : Full page access
- ❌ Users WITHOUT: Redirected to `/unauthorized`

3. Email Logs Page

Location

`/frontend/src/app/[locale]/(admin)/secureacwadash/emails/page.tsx`

Current Status

✅ Backend protected: `permission:viewAny emailLog`
✅ Sidebar hidden: Uses `viewAny emailLog` permission
❌ Frontend page: No permission check - anyone can access via URL

Required Permission

- Backend uses: `viewAny emailLog`
- Sidebar shows if: `viewAny emailLog`
- **Frontend should use: `viewAny emailLog`** (match backend)

Implementation

Wrap entire component with AdminPermissionWrapper:

```
// File: /frontend/src/app/[locale]/(admin)/secureacwadash/emails/page.tsx

"use client";

import { useEmailsLogs } from "@/Admin/Hooks/Emails/useEmailsLogs";
import Pagination from "@/Admin/common/Pagination";
import { useState, useMemo, useEffect } from "react";
import WebsiteLoading from "@/Components/UI/Loading/WebsiteLoading";
import EmailTable from "@/Admin/Components/EmailPage/EmailTable";
import { useLocale, useTranslations } from "next-intl";
import SearchAndFilters from "@/Admin/common/SearchAndFilters";
import { useAppSelector } from "@/Hooks/ReduxHooks/ReduxHooks";
import { useTheme } from "next-themes";
import AdminPermissionWrapper from '@/Admin/Components/HOC/AdminPermissionWrapper'; //
```

```
interface EmailFilters {
  search?: string;
  status?: string;
  sort_by?: string;
  per_page?: string;
}

export default function EmailsPage() {
  // ... all existing code ...

  return (
    <AdminPermissionWrapper requiredPermission="viewAny emailLog">
      <main className={`${
        locale === "ar"
          ? IsSideBarOpen ? "lg:pr-72" : "lg:pr-28"
          : IsSideBarOpen ? "lg:pl-72" : "lg:pl-28"
      } pt-16 transition-colors duration-300 ${
        theme === 'dark' ? 'bg-gray-900 text-white' : 'bg-gray-50 text-gray-900'
      }`}>
        {/* ... all existing page content ... */}
      </main>
    </AdminPermissionWrapper>
  );
}
```

Result

- ✅ Users WITH `viewAny emailLog` : Full page access
- ❌ Users WITHOUT: Redirected to `/unauthorized`

Permission Mapping Summary

Feature	Location	Backend Permission	Sidebar Permission	Frontend Should Use
Order Codes/Serials	Component section	`view codes`	N/A (not in sidebar)	`view codes`
SMS Logs Page	Full page	`viewAny smsLog`	`export data`	`viewAny smsLog`
Email Logs Page	Full page	`viewAny emailLog`	`viewAny emailLog`	`viewAny emailLog`

Sidebar Permission Issue (⚠️ Inconsistency)

SMS Logs Sidebar:

```
// Current (line 196 in Resoruce.ts)
{
  name: "sms_logs",
  path: "/secureacwadash/sms-logs",
  permission: "export data",  // ⚠️ Wrong!
}
```

Issue: Sidebar shows for `export data` but page needs `viewAny smsLog`.

Recommended Fix: Change sidebar permission to match backend:

```
{
  name: "sms_logs",
  path: "/secureacwadash/sms-logs",
  permission: "viewAny smsLog",  // ✅ Correct
}
```

Same for Order Logs:

```
{
  name: "order-logs",
  path: "/secureacwadash/order-logs",
  permission: "viewAny order",  // Instead of "export data"
}
```

Same for Activity:

```
{
  name: "activity",
  path: "/secureacwadash/activity",
  permission: "view statistics",  // Instead of "export data"
}
```

Implementation Checklist

Order Codes/Serials:

- [ ] Import `hasPermission` utility

- [ ] Import `useAppSelector` hook
- [ ] Get user permissions from Redux
- [ ] Add permission check to codes section (line 692)
- [ ] Test: User WITH `view codes` sees codes
- [ ] Test: User WITHOUT `view codes` doesn't see section

SMS Logs Page:

- [ ] Import `AdminPermissionWrapper`
- [ ] Wrap entire page component
- [ ] Set `requiredPermission="viewAny smsLog"`
- [ ] Test: User WITH permission sees page
- [ ] Test: User WITHOUT permission → redirected
- [ ] Fix sidebar permission from `export data` to `viewAny smsLog`

Email Logs Page:

- [ ] Import `AdminPermissionWrapper`
- [ ] Wrap entire page component
- [ ] Set `requiredPermission="viewAny emailLog"`
- [ ] Test: User WITH permission sees page
- [ ] Test: User WITHOUT permission → redirected

Sidebar Consistency:

- [ ] Change SMS logs sidebar: `export data` → `viewAny smsLog`
- [ ] Change order logs sidebar: `export data` → `viewAny order`
- [ ] Change activity sidebar: `export data` → `view statistics`

Testing Scenarios

Test 1: Order Codes Visibility

1. Create user with `viewAny order` only (NO `view codes`)
2. Navigate to order details page
3. ✅ Expected: Order details visible, codes section hidden
4. Grant `view codes` permission
5. ✅ Expected: Codes section now visible

Test 2: SMS Logs Access

1. Create user with `access dashboard` only
2. Try navigating to `/secureacwadash/sms-logs`
3. ✅ Expected: Redirected to `/unauthorized`
4. Grant `viewAny smsLog` permission
5. ✅ Expected: Full page access

Test 3: Email Logs Access

1. Create user with `access dashboard` only
2. Try navigating to `/secureacwadash/emails`
3. ✅ Expected: Redirected to `/unauthorized`
4. Grant `viewAny emailLog` permission
5. ✅ Expected: Full page access

Test 4: Sidebar Consistency

1. User has `viewAny smsLog` but NOT `export data`
2. ✅ Expected: SMS Logs visible in sidebar (after fix)
3. Click SMS Logs → page loads successfully

Backend Permissions (Already Implemented ✅)

From `acwady/routes/versions/admin/v1.php`:

```
// Order codes endpoint
Route::post('products/codes', [CodeController::class, 'getOrderProductCodes'])
    ->middleware('permission:view codes');

// SMS logs endpoints
Route::get('sms-logs', [SmsLogController::class, 'index'])
    ->middleware('permission:viewAny smsLog');

Route::get('sms-logs/{id}', [SmsLogController::class, 'show'])
    ->middleware('permission:view smsLog');

// Email logs endpoints
Route::get('email-logs', [EmailLogController::class, 'index'])
    ->middleware('permission:viewAny emailLog');

Route::get('email-logs/{id}', [EmailLogController::class, 'show'])
    ->middleware('permission:view emailLog');
```

All backend routes are properly protected ✅

Files to Modify

File	Action
`/frontend/src/Admin/Components/OrderDetailsPage/OrderDetailsPage.tsx`	Add permission check
`/frontend/src/app/[locale]/(admin)/secureacwadash/sms-logs/page.tsx`	Wrap with permission
`/frontend/src/app/[locale]/(admin)/secureacwadash/emails/page.tsx`	Wrap with permission
`/frontend/src/Admin/Resoruce/Resoruce.ts`	Fix sidebar permissions

Total: 4 files

Security Impact

Before Fix:

- 🔴 Anyone can see order serial codes (high-value data leak)
- 🔴 Anyone can access SMS logs (contains customer phone numbers)
- 🔴 Anyone can access email logs (contains customer emails)

After Fix:

- ✅ Only authorized users see serial codes
- ✅ Only users with `viewAny` `smsLog` access SMS logs
- ✅ Only users with `viewAny` `emailLog` access email logs
- ✅ Consistent permissions across backend/frontend/sidebar

Report Date: January 25, 2026
Priority: 🔴 **HIGH** (Customer data exposure risk)
Estimated Implementation Time: 30 minutes