

Distribution Management System – WSO2 IS + .NET 9 Guide

Table of Contents

- [Introduction](#)
- [Architecture Overview](#)
- [OAuth2 + JWT Integration with WSO2 IS](#)
- [Project Structure](#)
- [API Examples](#)
- [Roles & Permissions](#)
- [Role-based API Access Table](#)
- [.env Configuration](#)
- [Local Development Tips \(Windows\)](#)
- [Troubleshooting](#)
- [Support & References](#)
- [Author](#)

Introduction

This document provides a complete technical reference for implementing a Distribution Management System with .NET 9 and WSO2 Identity Server 7.1. It covers the architecture, key concepts (OAuth2, JWT, role-based access), project file structure, APIs, sample request/response, and troubleshooting.

Architecture Overview

- **Web UI** (HTML/CSS/JavaScript) served from wwwroot.
- **ASP.NET Core 9 API**: Manages authentication (OAuth2 Password grant), products & users, serves SPA, and provides Swagger docs.
- **Application Layer**: DTOs, interfaces, business logic.
- **Infrastructure Layer**: EF Core database, repository patterns, integration with WSO2 IS.
- **Domain Layer**: Core Entities, Enums, Business Rules.
- **Identity Management**: WSO2 IS provides OAuth2 authentication, user info, and roles.

OAuth2 + JWT Integration with WSO2 IS

- **Login flow**: Frontend POSTs username and password → .NET backend → WSO2 IS `/token` endpoint → receives JWT bearer token.
- **User Info**: Backend calls WSO2 IS `/userinfo` endpoint (with Bearer token) to get user roles and info.
- **Role Extraction**: Roles (`yks_admin`, `yks_user`, `yks_test`) added from the JWT claims, used in `[Authorize]` directives.

Project Structure

```
WS02_IS_Dotnet_SS0/
├── src/
│   ├── DistributionManagement.Domain/
│   │   ├── Entities/{Product,User}.cs
│   │   └── Enums/UserRole.cs
│   ├── DistributionManagement.Application/
│   │   ├── DTOs/
│   │   ├── Interfaces/
│   │   └── Services/
│   ├── DistributionManagement.Infrastructure/
│   │   ├── Data/ (ApplicationDbContext, DbInitializer)
│   │   ├── Repositories/
│   │   └── ExternalServices/WS02AuthenticationService.cs
│   └── DistributionManagement.API/
│       ├── Controllers/{AuthController,ProductController,UserController}.cs
│       ├── Middleware/
│       ├── Configuration/
│       ├── wwwroot/{index.html,styles.css,app.js}
│       ├── Program.cs
│       └── appsettings.json
├── setup.sh
├── scripts/setup.ps1
├── README.md
├── QUICKSTART.md
└── WS02_IS_Dotnet_SS0.sln
```

API Examples

Login

```
curl -X POST http://localhost:5299/api/auth/login \
-H "Content-Type: application/json" \
-d '{"username": "yks", "password": "YOUR_PASSWORD"}'
```

Sample Response:

```
{
  "accessToken": "eyJ...",
  "tokenType": "Bearer",
  "expiresIn": 3600,
  "userInfo": {
    "username": "yks",
    "role": "yks_admin",
    "roles": ["yks_admin"]
  }
}
```

Get Products

```
curl -X GET http://localhost:5299/api/product \
-H "Authorization: Bearer YOUR_TOKEN"
```

Sample Response:

```
[
  {
    "id": 1,
    "name": "Product A",
    "description": "Description for Product A",
    "price": 99.99,
    "stockQuantity": 100,
    "category": "Category 1",
    "sku": "SKU001",
    "isActive": true
  },
  ...
]
```

Create Product

```
curl -X POST http://localhost:5299/api/product \
-H "Authorization: Bearer YOUR_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "name": "Test Product",
  "description": "Something",
  "price": 123.45,
  "stockQuantity": 99,
  "category": "General",
  "sku": "SKU999",
  "isActive": true
}'
```

Sample Response:

```
{
  "id": 12,
  "name": "Test Product",
  "description": "Something",
  "price": 123.45,
  "stockQuantity": 99,
  "category": "General",
  "sku": "SKU999",
  "isActive": true
}
```

Roles & Permissions

- **yks_admin** – Full CRUD on products, View/Create/Manage Users
- **yks_user** – View/Create/Edit products
- **yks_test** – View products only

Role-based API Access Table

Method	Endpoint	Roles
GET	/api/product	yks_admin, yks_user, yks_test
POST	/api/product	yks_admin, yks_user

Method	Endpoint	Roles
PUT	/api/product/{id}	yks_admin, yks_user
DELETE	/api/product/{id}	yks_admin
GET	/api/user	yks_admin

.env Configuration

```
WSO2__TokenEndpoint=https://iam.bimats.com/oauth2/token
WSO2__UserInfoEndpoint=https://iam.bimats.com/oauth2/userinfo
WSO2__ClientId=YOUR_CLIENT_ID
WSO2__ClientSecret=YOUR_CLIENT_SECRET
JWT__Issuer=https://iam.bimats.com/oauth2/token
JWT__Audience=YOUR_CLIENT_ID
Database__Provider=SQLite
ConnectionStrings__DefaultConnection=Data Source=distribution.db
ASPNETCORE_ENVIRONMENT=Development
```

Local Development Tips (Windows)

- Trust dev HTTPS certificate if you use HTTPS: `dotnet dev-certs https --trust`
- Reset database (development): delete or move `src/DistributionManagement.API/distribution.db`
- If scripts are blocked: launch PowerShell as Administrator and run: `Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass`

Troubleshooting

- **403 Unauthorized:** verify JWT, user roles, Authorization header.
- **Port 5299 in use:** `taskkill /PID <pid> /F` after finding with `netstat -ano | findstr :5299`
- **Database errors:** remove `distribution.db` and rerun.
- **WSO2 connection:** verify credentials, endpoint URL, OAuth2 client ID/secret, and login with `swagger/curl`.

Support & References

- [WSO2 IS 7.1 Docs](#)
- [.NET 9 Docs](#)
- [OAuth2 Spec](#)
- [JWT.io Debugger](#)

Author

Thomas Hein Thura – https://github.com/ThomasHeinThura/WSO2_IS_Dotnet_SSO