

Directory Structure Guide

Ape Framework - Directory Structure Guide

Introduction

This guide provides a comprehensive overview of the Ape Framework's directory structure, explaining the purpose of each folder and key files within the project.

Root Directory Structure

D:\Projects\Repos\Ape\	
├── Ape/	# Main application folder
├── docs/	# Documentation (markdown and PDF)
├── README.md	# Project readme
├── LICENSE	# MIT license
├── Ape.slnx	# Visual Studio solution file
└── .gitignore	# Git ignore rules

Main Application Structure

Ape/	
├── Areas/	# ASP.NET Areas (Identity)
├── Controllers/	# MVC Controllers
├── Data/	# Database context and migrations
├── Middleware/	# Custom middleware
├── Models/	# Entity models and view models
├── Services/	# Business logic services
├── Views/	# Razor views
├── wwwroot/	# Static files (CSS, JS, images)
├── ProtectedFiles/	# Uploaded PDF documents
├── Program.cs	# Application startup
├── appsettings.json	# Configuration
├── web.config	# IIS configuration
└── Ape.csproj	# Project file

Detailed Directory Breakdown

/Controllers

Contains MVC controllers that handle HTTP requests and return responses.

File	Purpose
InfoController.cs	Public pages (Home, Contact, Privacy, TOS) and contact form processing
DocumentController.cs	Document library management - upload, download, folder operations
GalleryController.cs	Image gallery management - upload, categorization, thumbnails
LinksController.cs	Link directory management - categories and links CRUD
SystemCredentialsController.cs	Encrypted credential management with audit logging
EmailLogController.cs	Email transmission history and log viewer
EmailTestController.cs	Email configuration testing tools
ContactFormSettingsController.cs	Configure contact form recipients
ActiveUsersController.cs	Dashboard showing currently active users

/Models

Contains entity classes that map to database tables and view models for the UI.

Core Entity Models

File	Purpose	Database Table
UserProfiles.cs	Extended user profile data (name, address, activity)	UserProfiles
SystemCredential.cs	Encrypted credential storage with audit support	SystemCredentials
EmailLog.cs	Email transmission records	EmailLogs
SystemSetting.cs	Key-value application settings	SystemSettings
PDFCategory.cs	Document library folders (hierarchical)	PDFCategories
CategoryFile.cs	Document files within folders	CategoryFiles
GalleryCategory.cs	Image gallery categories (hierarchical)	GalleryCategories
GalleryImage.cs	Images within categories	GalleryImages
LinkCategory.cs	Link directory categories	LinkCategories
CategoryLink.cs	Individual links	CategoryLinks

Supporting Classes (in SystemCredential.cs)

Class	Purpose
CredentialAuditLog	Audit trail for credential access/changes
CredentialCategory	Static class with category constants (Database, API, Email, etc.)
CredentialAction	Static class with action constants (Created, Updated, Viewed, etc.)

View Models (/Models/ViewModels/)

File	Purpose
DocumentExplorerViewModels.cs	Models for document browser (folders, files, breadcrumbs, tree)
GalleryViewModels.cs	Models for gallery browser with pagination support
CredentialViewModels.cs	Models for credential management UI
ActiveUsersViewModel.cs	Models for active users dashboard

/Services

Contains business logic services that implement core functionality.

File	Purpose
CredentialEncryptionService.cs	AES-256 encryption/decryption for credentials using master key
SecureConfigurationService.cs	Retrieves and decrypts credentials, manages audit logging
DocumentManagementService.cs	Document library operations (folders, files, access control)
GalleryManagementService.cs	Gallery operations (categories, images, thumbnails)
ImageOptimizationService.cs	Image processing, resizing, and thumbnail generation
EmailService.cs	SMTP-based email sending (implements IEmailSender)
EnhancedEmailService.cs	Advanced email with Azure Communication Services + SMTP fallback
SystemSettingsService.cs	Application settings management

Service Interfaces

Interface	Implementation
IDocumentManagementService	DocumentManagementService
IGalleryManagementService	GalleryManagementService
IImageOptimizationService	ImageOptimizationService
ISystemSettingsService	SystemSettingsService

/Data

Contains Entity Framework Core database context and migrations.

File/Folder	Purpose
ApplicationDbContext.cs	Main DbContext with DbSet for all entities and model configurations
/Migrations/	EF Core migration files for database schema evolution

DbSets in ApplicationDbContext

```

DbSet<UserProfiles> UserProfiles
DbSet<SystemCredential> SystemCredentials
DbSet<CredentialAuditLog> CredentialAuditLogs
DbSet<EmailLog> EmailLogs
DbSet<SystemSetting> SystemSettings
DbSet<PDFCategory> PDFCategories
DbSet<CategoryFile> CategoryFiles
DbSet<GalleryCategory> GalleryCategories
DbSet<GalleryImage> GalleryImages
DbSet<LinkCategory> LinkCategories
DbSet<CategoryLink> CategoryLinks

```

/Middleware

Contains custom ASP.NET Core middleware components.

File	Purpose
ActivityTrackingMiddleware.cs	Tracks user activity by updating LastActivity timestamp (throttled to every 2 minutes)

/Views

Contains Razor views organized by controller.

```
Views/
├── Shared/
│   ├── _Layout.cshtml           # Shared layouts and partials
│   ├── _PartialHeader.cshtml    # Master layout template
│   ├── _PartialFooter.cshtml    # Navigation header
│   ├── _Footer.cshtml           # Footer
│   ├── _LoginPartial.cshtml     # Login/logout UI
│   ├── _TopButton.cshtml        # Scroll-to-top button
│   ├── _ValidationScriptsPartial.cshtml
│   └── Error.cshtml
├── Info/
│   ├── Index.cshtml             # Public pages
│   ├── Contact.cshtml           # Home page
│   ├── Privacy.cshtml           # Contact form
│   ├── TOS.cshtml               # Privacy policy
│   └──
├── Document/
│   ├── Browse.cshtml            # Document library
│   ├── ViewPdf.cshtml           # Main document explorer
│   └── _FolderTree*.cshtml       # PDF viewer
├── Gallery/
│   ├── Browse.cshtml            # Folder tree partials
│   └── _CategoryTree*.cshtml     # Image gallery
├── Links/
│   ├── MoreLinks.cshtml         # Main gallery browser
│   ├── ManageCategories.cshtml  # Category tree partials
│   └── ManageLinks.cshtml       # Link directory
├── SystemCredentials/
│   ├── Index.cshtml             # Public link display
│   ├── Create.cshtml            # Admin category management
│   ├── Edit.cshtml              # Admin link management
│   ├── Delete.cshtml            # Credential management
│   └── AuditHistory.cshtml       # Credential list
├── EmailLog/
│   ├── Index.cshtml             # Create credential
│   └──
├── EmailTest/
│   ├── Index.cshtml             # Edit credential
│   └──
├── ContactFormSettings/
│   ├── Index.cshtml             # Delete confirmation
│   └──
├── ActiveUsers/
│   ├── Index.cshtml             # Audit log viewer
│   └──
└──
```

/Areas/Identity

Contains ASP.NET Core Identity Razor Pages for authentication.

```
Areas/Identity/Pages/
├── Account/
│   ├── Login.cshtml             # User login
│   ├── Register.cshtml          # User registration
│   ├── ForgotPassword.cshtml     # Password reset request
│   ├── ResetPassword.cshtml      # Password reset form
│   ├── ConfirmEmail.cshtml       # Email confirmation
│   └── Manage/
│       ├── Index.cshtml          # Account management pages
│       ├── ChangePassword.cshtml # Profile overview
│       ├── TwoFactorAuthentication.cshtml
│       └── EnableAuthenticator.cshtml
│   ...
├── Users.cshtml                 # User list (Admin)
├── EditUser.cshtml              # Edit user (Admin)
├── RoleManagement.cshtml        # Role management (Admin)
└── ManageRoles.cshtml           # Assign roles to user (Admin)
```

/wwwroot

Contains static files served directly to clients.

```
wwwroot/
├── css/
│   └── site.css           # Main application styles
├── js/
│   └── site.js           # Main JavaScript (includes top button,
etc.)
│   └── qr.js             # QR code functionality
├── lib/                  # Client-side libraries
│   ├── bootstrap/       # Bootstrap CSS/JS
│   ├── jquery/          # jQuery
│   └── jquery-validation/ # Form validation
│   └── ...
├── images/
│   └── site/             # Site images (logo, etc.)
├── Galleries/           # Uploaded gallery images (runtime)
└── favicon.ico          # Browser icon
```

/ProtectedFiles

Contains uploaded PDF documents for the Document Library.

- **Location:** Outside wwwroot for security
 - **Access:** Controlled through DocumentController
 - **Files:** PDF documents organized by database references
-

Configuration Files

Program.cs

Application entry point and configuration: - Database connection setup (environment variables or appsettings) - Identity configuration (roles, email confirmation) - Dependency injection registration - Middleware pipeline configuration - Auto-migration on startup - Default admin account creation

appsettings.json

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=...;Database=...;"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "AdminEmail": "",
  "AdminPassword": ""
}
```

Ape.csproj

Project configuration: - Target Framework: .NET 10.0 - NuGet packages: - Microsoft.AspNetCore.Identity.EntityFrameworkCore - Microsoft.AspNetCore.Identity.UI - Microsoft.EntityFrameworkCore.SqlServer - Azure.Communication.Email - SixLabors.ImageSharp

web.config

IIS hosting configuration for ASP.NET Core Module V2.

Environment Variables

Variable	Purpose
DB_SERVER_ILLUSTRATE	Database server address
DB_NAME_ILLUSTRATE	Database name
DB_USER_ILLUSTRATE	Database username
DB_PASSWORD_ILLUSTRATE	Database password
MASTER_CREDENTIAL_KEY_ILLUSTRATE	Master encryption key (32+ characters)

Key Architectural Decisions

Hierarchical Data

Document folders and gallery categories use self-referential relationships: - ParentCategoryID points to parent record (null = root level) - ChildCategories navigation property for children - Cascade delete restricted to prevent orphaned records

Access Control

Two-level access system: - **Member** (0) - Visible to all authenticated users - **Admin** (1) - Visible only to administrators

Encryption

System credentials encrypted with: - AES-256 algorithm (CBC mode, PKCS7 padding) - Master key from environment variable - Random IV for each encryption - Audit logging for all access

Email System

Dual-provider architecture: - Primary: Azure Communication Services - Fallback: SMTP - Automatic failover on Azure failures - All transmissions logged

File Naming Conventions

Type	Convention	Example
Controllers	{Name}Controller.cs	DocumentController.cs
Services	{Name}Service.cs	EmailService.cs
Interfaces	I{Name}Service.cs	IDocumentManagementService.cs
Models	{EntityName}.cs	SystemCredential.cs
ViewModels	{Feature}ViewModels.cs	GalleryViewModels.cs
Views	{Action}.cshtml	Browse.cshtml
Partials	_{Name}.cshtml	_TopButton.cshtml

Version: 1.0.0 **Framework:** Ape Framework **Site:** <https://Illustrate.net>