

Getting Started Guide

Ape Framework - Getting Started Guide

Introduction

Welcome to the Ape Framework! This guide will walk you through the initial setup and configuration of your new ASP.NET Core MVC application.

Live Demo: <https://Illustrate.net> **Repository:** <https://github.com/MikishVaughn/Ape>

Prerequisites

Before you begin, ensure you have the following installed:

- **.NET 10.0 SDK or later**
 - Download from: <https://dotnet.microsoft.com/download>
 - Verify installation: `dotnet --version`
 - **SQL Server**
 - LocalDB (included with Visual Studio) for development
 - SQL Server Express or full SQL Server for production
 - **Visual Studio 2022 or VS Code (recommended)**
 - Visual Studio: Include “ASP.NET and web development” workload
 - VS Code: Install C# extension
-

Step 1: Clone the Repository

Open a terminal or command prompt and run:

```
git clone https://github.com/MikishVaughn/Ape.git  
cd Ape
```

Step 2: Set Up the Master Encryption Key

The framework uses AES-256 encryption for storing sensitive credentials. You must set a master key before running the application.

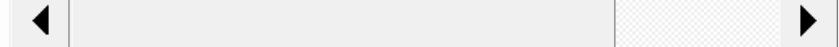
For Development (Windows)

Option A: Environment Variable

1. Open System Properties → Advanced → Environment Variables
2. Add a new User variable:
 - Name: `MASTER_CREDENTIAL_KEY_ILLUSTRATE`
 - Value: A secure string of 32+ characters (e.g., `YourSecure32CharacterKeyHere123!`)

Option B: User Secrets (Recommended for Development)

```
cd Ape  
dotnet user-secrets init  
dotnet user-secrets set "MASTER_CREDENTIAL_KEY_ILLUSTRATE" "YourSecure32Ch
```



For Production

Set environment variables at the server/hosting level:

Variable	Description
MASTER_CREDENTIAL_KEY_ILLUSTRATE	Encryption key (32+ characters)
DB_SERVER_ILLUSTRATE	Database server address
DB_NAME_ILLUSTRATE	Database name
DB_USER_ILLUSTRATE	Database username
DB_PASSWORD_ILLUSTRATE	Database password

Step 3: Configure the Database

Development (LocalDB)

The default configuration in `appsettings.json` uses LocalDB:

```
{  
  "ConnectionStrings": {  
    "DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=aspnet-A  
  }  
}
```



No changes needed for local development.

Production

The application automatically constructs the connection string from environment variables when they are present. Set the `DB_*` variables listed above.

Step 4: Run Database Migrations

The application auto-migrates on startup, but you can also run migrations manually:

```
cd Ape  
dotnet ef database update
```

This creates all required tables: - Identity tables (users, roles, claims) - UserProfiles (extended user data) - SystemCredentials (encrypted credential storage) - EmailLogs (email audit trail) - PDFCategories & CategoryFiles (document library) - GalleryCategories & GalleryImages (photo gallery) - LinkCategories & CategoryLinks (links directory)

Step 5: Build and Run

```
dotnet build  
dotnet run
```

The application will start and display the URL (typically <https://localhost:5001>).

Step 6: First Login

Default Admin Account

On first run, if no admin users exist, the system creates a default administrator:

- **Email:** admin@admin.com
- **Password:** Admin123!

IMPORTANT: Change this password immediately after first login!

Custom Admin Credentials

You can specify custom admin credentials in `appsettings.json` before first run:

```
{  
  "AdminEmail": "your-email@example.com",  
  "AdminPassword": "YourSecurePassword123!"  
}
```

Step 7: Configure Email (Required for Full Functionality)

Email is required for: - User registration (email confirmation) - Password reset - Contact form - Two-factor authentication setup

See the [Email Configuration Guide](#) for detailed setup instructions.

Quick Setup: 1. Log in as Admin 2. Navigate to [System Credentials](#) 3. Add your SMTP credentials (at minimum): - `SMTP_SERVER` - e.g., `smtp.gmail.com` - `SMTP_PORT` - e.g., `587` - `SMTP_USERNAME` - your email - `SMTP_PASSWORD` - your password or app password - `SMTP_SSL` - true

Step 8: Verify Your Setup

After completing the steps above, verify everything works:

1. **Test Email:** Go to Admin menu → Email Test → Send a test email
 2. **Register a User:** Log out, register a new account, confirm email
 3. **Document Library:** Create a folder, upload a PDF
 4. **Photo Gallery:** Create a category, upload an image
 5. **Contact Form:** Submit a test message from the Contact page
-

Project Structure Overview

```
Ape/  
  |-- Areas/Identity/          # Authentication pages (login, register, 2FA)  
  |-- Controllers/            # MVC controllers  
  |-- Data/                   # Database context and migrations  
  |-- 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
```

Next Steps

- [Email Configuration Guide](#) - Set up Azure and SMTP email
 - [Administrator Guide](#) - Learn all admin features
 - [User Management Guide](#) - Managing users and roles
 - [Document & Gallery Guide](#) - Managing content
-

Troubleshooting

“Master key not configured”

Ensure the `MASTER_CREDENTIAL_KEY_ILLUSTRATE` environment variable is set and restart the application.

Database connection errors

1. Verify SQL Server is running
2. Check connection string in appsettings.json
3. For production, verify environment variables are set

Email not sending

1. Check System Credentials for correct email settings
2. Use the Email Test page to diagnose issues
3. Review Email Logs for error messages

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