ValueMomentum

Cloud Native Application templates

Contents

[1 Problem Statement 2](#_Toc168480998)

[2 Solution Approach 2](#_Toc168480999)

[3 Resources 3](#_Toc168481000)

[4 Tech Stack Requirements 3](#_Toc168481001)

[5 Tentative Timelines 3](#_Toc168481002)

# Problem Statement

In typical project implementations, teams spend a considerable amount of time setting up the project structure. This includes:

1. **Project Structuring**: Determining the placement of various implementations within the project.
2. **Dependency Injection (DI)**: Effectively performing dependency injection.
3. **Infrastructure Services**: Implementing the base around infrastructure services.
4. **Startup Code Setup**:
   * Loading configurations from external configuration providers.
   * Loading logging-related configurations.
   * Loading commonly used dependencies/frameworks such as AutoMapper, MediatR, etc.

This overhead requires developers to devise an approach to address these concerns and ensure compliance throughout the project implementation timeline along with the feature/functionality implementation.

# Solution Approach

To address these concerns, we propose leveraging pre-built project templates. These templates come pre-equipped with:

1. **Standard Project Structure**: Aligning with CLEAN architecture principles.
2. **Comprehensive Startup Code**: Covering all startup code concerns.
3. **Dapr Integration**: Leveraging Dapr for infrastructure-related implementations and base implementations.
4. **AutoFac DI Capabilities**: Utilizing AutoFac for dependency injection.
5. **Azure App Configuration**: Using Azure App configuration as an external configuration provider.
6. **Assembly Scanning**: Enabling assembly scanning to load AutoMapper and MediatR framework-related files.
7. **AutoMapper**: Handling mapping concerns wherever applicable.
8. **MediatR Framework**: Implementing the CQRS pattern for functionality.
9. **12 Factor Application Methodology:** implement templates aligning to this methodology.

These templates cover the following use cases:

1. **Web API Template**: Integrated with Entity Framework.
2. **System API Template**: Integrated with the Dapr framework.
3. **Domain API Template**: Integrated with the Dapr framework.
4. **Experience API Template**: Integrated with the Dapr framework.
5. **.NET Full Stack Application**: Supporting Blazor/Razor Pages, MVVM, Entity framework, and other infrastructures.
6. **Batch Application:** leveraging .NET core background services
7. **Message driven Application:**  supporting Azure Service Bus (ASB), Azure EventHub.
8. **Azure Durable Functions:** .NET application template

To ensure that developers are fully aligned with these templates, a playbook will be provided. This will detail the standards and principles followed for project template implementation and how to align with them during development activities. Which also includes, a checklist that should be followed by reviewers for enforcing required governance.

These templates will be developed aligning to the Microsoft coding standards and best practices. If there is a need, these templates can be tweaked to align with an enterprise level coding standard.

# Resources

|  |  |
| --- | --- |
| Skill Set | No Of Resources |
| Tech lead – c#, .Net Core & Azure | 1 |
| Developer – c#, .Net Core | 1 |

# Tech Stack Requirements

* C#
* .NET Core framework v8
* MediatR
* AutoFac
* AutoMapper
* Dapr + Dapr dotnet SDK
* Entity Framework
* Azure Messaging Platforms - ASB & Eventhubs
* Application Insights
* Azure App Configuration
* Blazor (optional)
* Razor Pages

# Tentative Timelines

|  |  |
| --- | --- |
| Activity | Duration (In Weeks) |
| Building Technical Competency & Planning | 2-4 |
| Understanding Domain API Template | 2 |
| Domain API Template – Documentation | 1 |
| Web API (System) Template - Design | 1 |
| Web API (System) Template – Development & Unit Testing | 3 |
| Web API (System) Template – Documentation & Training Material | 1 |
| Full Stack App Template - Design | 2 |
| Full Stack App Template – Development & Unit Testing | 6 |
| Full Stack App Template – Documentation & Training Material | 2 |
| Experience API Template - Design | 1 |
| Experience API Template – Development & Unit Testing | 3 |
| Experience API Template – Documentation & Training Materials | 1 |
| Batch App Template - Design | 1 |
| Batch App Template – Development & Unit Testing | 3 |
| Batch App Template – Documentation & Training Material | 1 |
| Message Driven App Template - Design | 1 |
| Message Driven App Template – Development & Unit Testing | 2 |
| Message Driven App Template – Documentation & Training Material | 1 |
| Azure Durable Function App Template - Design | 1 |
| Azure Durable Function App Template – Development & Unit Testing  Durable Function Types:  Orchestrator, Activity, Entity & Client Functions | 6 |
| Azure Durable Function App Template – Documentation & Training Material | 2 |