csharp.instructions

C#/.NET Coding Standards and Best Practices

This document defines the C# and .NET conventions, patterns, and best practices for this repository. All contributors should follow these guidelines for consistency and maintainability.

---

1. File & Project Structure

- One type per file; file name matches type name.
- File-scoped namespaces.
- Organize by feature (not technical layer) where possible.
- Use Clean Architecture: Domain, Application, Infrastructure, Api.

2. Naming Conventions

- `PascalCase` for types, methods, properties, events.
- `camelCase` for local variables and parameters.
- Constants: `ALL_CAPS`.
- Interfaces: prefix with `I` (e.g., `INotificationService`).

3. Coding Style

- 4-space indentation.
- Use `async/await` for all async operations.
- Use `nameof` in exceptions and guard clauses.
- Prefer `sealed` classes unless inheritance is required.
- Use guard clauses (fail fast) instead of nested ifs.
- Prefer immutable value objects and strongly-typed IDs.

4. Exception Handling

- Throw specific exceptions (e.g., `ArgumentNullException`).
- Use guard clauses for parameter validation.
- Avoid catching general `Exception` unless necessary.

5. Testing

- Use xUnit for unit/integration tests.
- Use FakeItEasy for mocking.
- Organize tests by feature and method.
- Name tests descriptively (method_underTest_expectedBehavior).

6. Documentation

- Use XML comments for public APIs.
- Keep method/class summaries clear and concise.

7. Commits & PRs

- Use Conventional Commits.
- One logical change per commit.

For more details, see the main Copilot instructions and workshop README.