

Pull Request Checklist

General Items

1. Code compiles
2. Code has been developer-tested
3. Code is tidy (indentation, line length, no commented-out code, no spelling mistakes, etc)
4. Code adheres to the Coding Styles and Standards Guidelines (Found in the documentation folder)
5. Exceptions have been used properly
6. New functionality is appropriately logged as needed
7. Unused using's have been removed
8. Eliminated warnings
9. Ensured NullReference Exceptions caught or handled appropriately
10. Leftover stubs and test routines have been removed
11. Hard-coded or development only things have been removed
12. Considered performance and scalability
13. Considered security, and potential exploits
14. All resources (HTTP connections, DB connection, files, etc) properly released in all code exits (i.e. normal and exception)
15. Corner cases and workarounds for known limitations of dependencies are well documented
16. No new code is a repeat of existing functionality that can be safely reused
17. Thread safety and possible deadlocks are considered and handled accordingly
18. Method bodies are kept as close to 4 lines or less if possible
19. Method names must be descriptive (reveal their intention aka what they do)
20. Methods do one thing only
21. Asynchronous Methods have the post-fix of "Async" appended to the end of their names.

Utopia Specific Checklist Items

1. All code added to UtopiaSharedClasses is safe to expose to customers (No sensitive information)
2. Libraries or projects that could become Libraries should be stored at the top level of the Utopia Repository
3. Public API endpoints have documentation comments if needed

General

1. .NET standard Coding Guidelines: <https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/>
2. Clean Code Book with corresponding guidelines are found in the Utopia Repository under Documentation.

Code Reviewer's Checklist

1. Types have been generalized where possible
2. Parameterized types have been used appropriately
3. Exceptions have been used appropriately
4. Repetitive code has been factored out
5. Frameworks have been used appropriately – methods have all been defined appropriately
6. Classes have been designed to only have one purpose
7. Views do not contain business logic
8. Common errors have been checked for
9. Potential threading issues have been eliminated where possible
10. Any security concerns have been addressed
11. Performance and Scalability was considered
12. The functionality fits the current design/architecture
13. The code does not use unjustifiable static methods/blocks
14. The code complies to coding standards
15. Logging used appropriately (proper logging level and details)
16. Exception use is in compliance with proper methodology
17. Code quality is excellent

Utopia Specific Checklist Items

1. All code added to UtopiaSharedClasses is safe to expose to customers (No sensitive information)
2. Libraries or projects that could become Libraries should be stored at the top level of the Utopia Repository
3. Public API endpoints have documentation comments if needed
4. Functionality should be exposed at the highest layer of the architecture as possible and only moved down when reuse becomes necessary. (i.e. CSV generation at the presentation layer, but could be moved into business logic if reuse becomes necessary)

Revision #2

Created 18 July 2022 23:13:59 by Bryce Holloway

Updated 9 February 2023 20:20:50 by Bryce Holloway