

BitWarden - MobCAT Code Review

/src/App/Pages/Settings/ExportVaultPageViewModel.cs

Position: 106:20-106:73

| | |
|--------------------|--|
| Priority | high |
| Title | [Threading] Avoid Task.Result usage |
| Description | Avoid Task.Result usage in application in apps with the UI thread, this can cause deadlocks, await a task instead. |
| SHA | cd3585be58672aa646fd3293ccfc051c6fe29a53 |

/src/Core/Services/ApiService.cs

Position: 416:0-417:88

| | |
|--------------------|---|
| Priority | medium |
| Title | [Performance] Data deserialization/decryption is happening on the UI thread |
| Description | Result of http requests is processed on UI thread (awaited), parsed (json) or even decrypted. This pattern affects performance because the actual data processing is happening on the calling thread and in most cases it's the UI thread. The issue found in a few places: ApiService.cs#L407, AuditService.cs#L36, CipherService#L610,L792. |
| SHA | cd3585be58672aa646fd3293ccfc051c6fe29a53 |

Position: 138:0-138:0

| | |
|--------------------|--|
| Priority | low |
| Title | [Maintainability] Error handling issues (empty catch, rethrow) |
| Description | In some places we found empty catches, which is a significant maintainability issue, there is no visibility if any error happens, empty catch found in a few places: Attachment.cs#L55, AuthService.cs#L312, MobilePlatformUtilsService#L236. Also in some places re-throw logic doesn't supply an original error details which leads to a missing stack trace ApiService.cs#L138. In case of re-throwing error as is, no need to specify a variable name, this will help to keep the original error stack trace AuthService.cs#L236 |
| SHA | cd3585be58672aa646fd3293ccfc051c6fe29a53 |

/src/Core/Services/BroadcasterService.cs

Position: 22:24-22:73

| | |
|--------------------|--|
| Priority | low |
| Title | [Architecture] Messenger pub/sub pattern increases complexity |
| Description | BroadcasterService generates a thread for every subscriber on every message send, this could generate lots of threads. Moreover there is no error handling in case if subscriber failed. This will also can lead to a race conditions. I would recommend to use sequential subscribers calls with proper error handling and use just a single thread to send a message. I would also avoid using Pub/Sub pattern to decrease complexity and make it easier to debug/maintain. As instead, introduce services and inject them where needed via interfaces to be called. |

SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

/src/Core/Services/LockService.cs

Position: 152:12-152:70

Priority low
Title [Threading] Return task without async/await
Description Return a Task directly without async/await when no Task result processing is required. It will reduce overhead related to restoring the execution context.

SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

/src/Core/Services/PclCryptoFunctionService.cs

Position: 215:0-218:69

Priority low
Title [Performance] Manual string processing affects memory and performance
Description Work with strings using Regex (preferred) or StringBuilder instead of sequenced string. Replace to optimize memory management and performance.

SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

/src/App/Pages/Vault/GroupingsPage/GroupingsPage.xaml.cs

Position: 100:28-100:55

Priority low
Title [Threading] Not awaited task execution
Description An asynchronous method which returns a task is called without await and without subsequent task processing which can lead to silent errors and unpredictable app behavior.

SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

Position: 108:15-127:17

Priority low
Title [Maintainability] Custom retry logic with magic numbers in code
Description Retry data loading in catch make it more complex to support and edit retry logic. It's better to use a retry framework to do this kind of tasks (e.g. Polly).

Additional Info <https://alastaircrabtree.com/implementing-the-retry-pattern-using-polly>

SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

/src/App/Pages/BaseViewModel.cs

Position: 16:0-16:45

Priority low
Title [XF] BaseViewModel depends on XF PageContent
Description ViewModel has direct reference to Page (PageContent). ViewModels shouldn't know anything

about XF or page. All the required XF functionality should be injected with IoC and services. A good practice is to move View Models to a project which doesn't depends on XF package to make sure no dependencies will be added in future, it helps to avoid coupling, improves testability and maintainability of the ViewModels code

SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

/src/App/Pages/Vault/GroupingsPage/GroupingsPage.xaml

Position: 92:16-132:44

Priority high
Title [XF] Switch from ListView to CollectionView
Description XF is going to improve CollectionView going forward. This is now the main control to render collection of elements, highly optimized for the best performance. Also the ListView control affects the memory performance because of recently discovered memory leak confirmed in this app. The ListView control will be supported but it's recommended to switch to the CollectionView as better alternative.

SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

Position: 1:0-10:28

Priority medium
Title [XF] Apply Xamarin.Forms best practices for high performance layouts
Description Most of the items are already addressed in the code, want to outline the list here to make sure every item is taken into account: xaml compiler, compiled bindings, less bindings, fast renderers, startup tracing, layout compression, optimized layout, optimized images, shallow visual tree

Additional Info <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/deploy-test/performance>

SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

/src/App/Controls/MiButton.cs

Position: 12:0-17:26

Priority low
Title [XF] Controls customization with styles is more flexible than with custom classes
Description Material, Mono, FontAwesome buttons and labels controls are introduced into the code base to support custom fonts. It's better (easier and more flexible, maintainable) to use styles, changed dynamically with a theme.

SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

/src/App/Controls/RepeaterView.cs

Position: 7:4-7:43

Priority low
Title [XF] Replace RepeaterView with ItemsSource/ItemTemplate
Description RepeaterView is a custom StackLayout with ItemsSource/ItemTemplate properties. It's supported by XF with attached properties ItemsSource/ItemsTemplate. Please also consider

using CollectionView whenever you have unpredictable number of items to ensure the best possible experience.

SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

/src/iOS.Extension/MainInterface.storyboard

Position:

Priority medium
Title [XF] Use Xamarin.Forms in iOS extension projects
Description Lots of existing XF pages where re-implemented in iOS extensions natively. There is a way to reuse existing logic in the iOS extensions as well.
Additional Info <https://docs.microsoft.com/en-us/xamarin/ios/platform/extensions-with-xamarinforms>
SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

/src/App/App.xaml.cs

Position: 44:0-154:0

Priority low
Title [Performance] Heavy app startup
Description This constructor will directly affect app startup performance before the app is interactive. It's better to keep this method as simple as possible and offload all the logic for later.
SHA cd3585be58672aa646fd3293ccfc051c6fe29a53

/src/App/Pages/Vault/AttachmentsPage.xaml

Position: 30:1-96:17

Priority high
Title [XF] Multiple nested layouts decrease performance
Description Avoid creating multiple nested layouts, use shallow layout instead: AddEditPage.xaml#L56 (nesting level 4), AttachmentsPage.xaml#L30 (nesting level 6), ScanPage.xaml#L16. Items layout especially affects the performance - AutofillCiphersPage.xaml#L66, especially in repeater as long as items are not recycled - CollectionsPage.xaml#L38, SharePage.xaml#L65
Additional Info <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/deploy-test/performance#optimize-layout-performance>
SHA cd3585be58672aa646fd3293ccfc051c6fe29a53