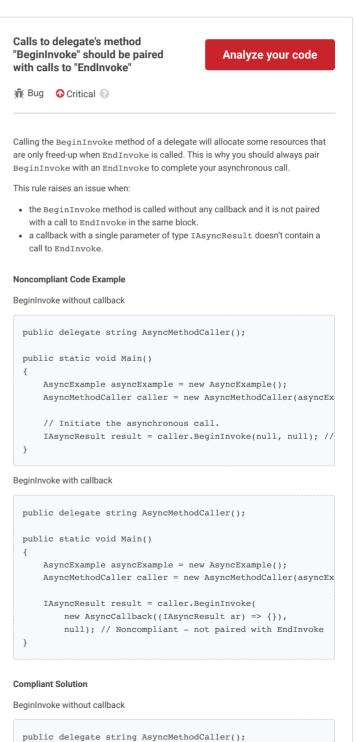


be marked with STAThread



public static void Main()

AsyncExample asyncExample = new AsyncExample();



Collection elements should not be replaced unconditionally



Exceptions should not be created without being thrown

```
🖟 Bug
```

Collection sizes and array length comparisons should make sense



Serialization event handlers should be implemented correctly

```
AsyncMethodCaller caller = new AsyncMethodCaller(asyncEx
    IAsyncResult result = caller.BeginInvoke(null, null);
    string returnValue = caller.EndInvoke(result);
}
```

BeginInvoke with callback

```
public delegate string AsyncMethodCaller();
public static void Main()
    AsyncExample asyncExample = new AsyncExample();
    AsyncMethodCaller caller = new AsyncMethodCaller(asyncEx
    IAsyncResult result = caller.BeginInvoke(
        new AsyncCallback((IAsyncResult ar) =>
                // Call EndInvoke to retrieve the results.
                string returnValue = caller.EndInvoke(ar);
            }), null);
}
```

Calling Synchronous Methods Asynchronously

Available In:

sonarlint ⊖ | sonarcloud ↔ | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy