New issue

Jump to bottom

Polymorphic serialization hangs #179

Closed ukarlsson opened this issue on Sep 2, 2021 · 5 comments

Labels state:waiting for response

ukarlsson commented on Sep 2, 2021

Hello. Many thanks for this great library. We ran into an issue that is quite strange. The deserialization hangs in this specific case, meaning that the process does not terminate, but uses 100% CPU.

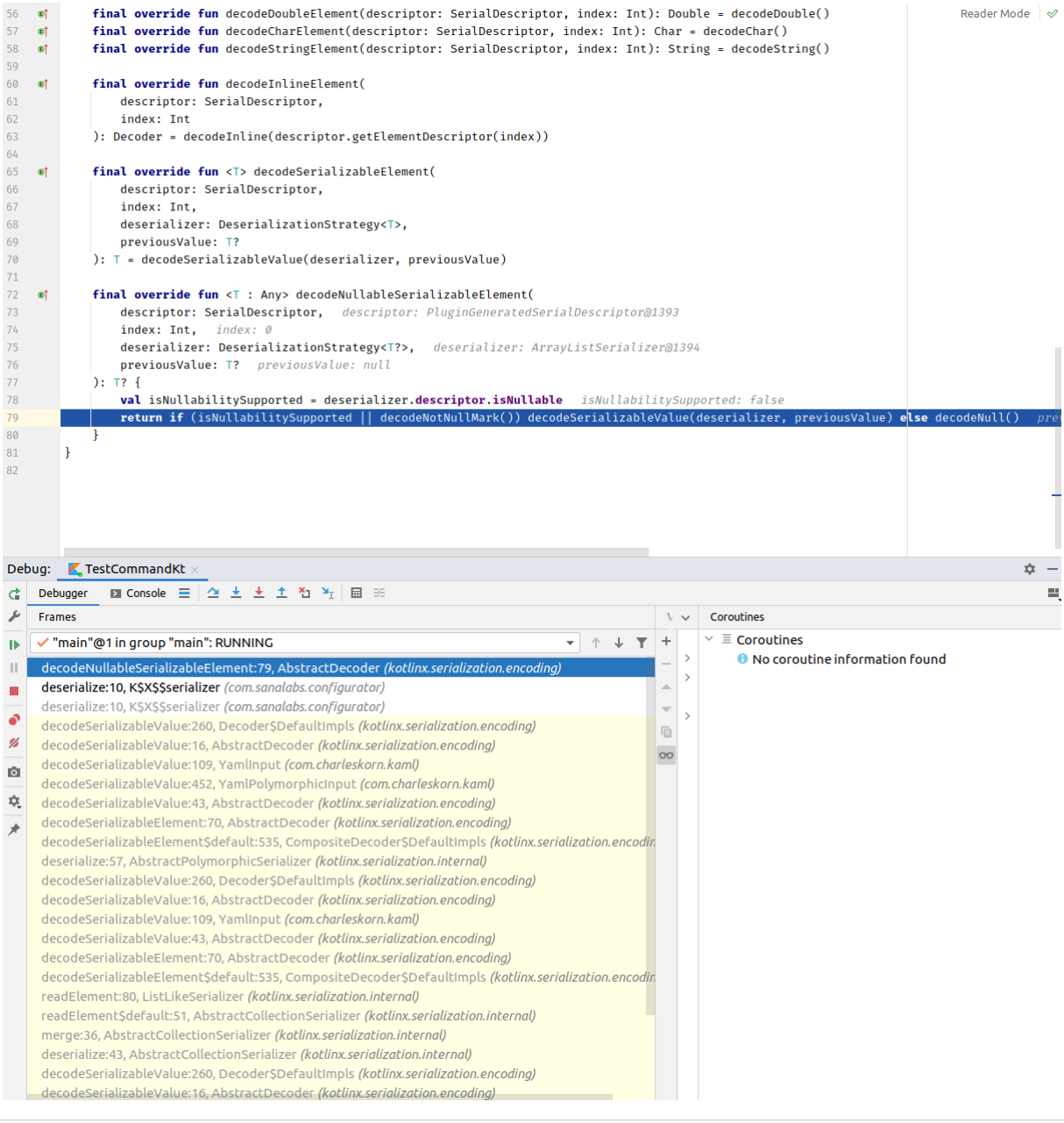
```
import com.charleskorn.kaml.Yaml
import kotlinx.serialization.SerialName
import kotlinx.serialization.Serializable
import kotlinx.serialization.builtins.ListSerializer

@Serializable
private sealed class K {
    @Serializable
    @SerialName("x")
    data class X(
        val property: List<String>? = null,
    ) : K()
}

const val s = """
- !<x>
"""

fun main() {
    Yaml.default.decodeFromString(ListSerializer(K.serializer()), s)
}
```

This is what I see when I pause and enter debugger



charleskorn commented on Sep 4, 2021 — Owner

Thanks for the bug report **@ukarlsson**. Are you able to share a sample project that triggers this issue?

If not, could you please share what version of kaml, Kotlin and kotlinx.serialization you're using?

charleskorn added the **state:waiting for response** label on Sep 4, 2021

ukarlsson commented on Sep 4, 2021 — Author

Latest versions that I could find. I attached project

kaml-bug.tar.gz

charleskorn commented on Sep 5, 2021 — Owner

Thanks for the sample project.

I've been able to reproduce the issue, and slightly simplify the scenario that triggers it:

```
import com.charleskorn.kaml.Yaml
import kotlinx.serialization.SerialName
import kotlinx.serialization.Serializable

@Serializable
private sealed class K {
    @Serializable
    @SerialName("x")
    data class X(
```

```
        val property: String? = null,
    ) : K()
}

const val s = """
!<x>
"""

fun main() {
    println("Started.")
    val result = Yaml.default.decodeFromString(K.serializer(), s)
    println("Finished, result is $result")
}
```

The hang doesn't happen when `PolymorphismStyle.Property` is used - it only affects tagged polymorphism.

charleskorn closed this as completed in `e18785d` on Sep 5, 2021

---

charleskorn commented on Sep 5, 2021 — Owner

Thanks again for reporting this issue **@ukarlsson**. I've fixed it and 0.35.3 includes the fix.

Note that this may be a vulnerability depending on how your application uses kaml - please see the security advisory for more information.

ukarlsson commented on Sep 6, 2021 — Author

Many thanks for fixing quickly!

Assignees

No one assigned

Labels

state:waiting for response

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants