

InternalCloningError

Table of contents

- [Serialization](#)
 - [Final variables](#)

Serialization

Wicket serializes pages to the session between requests. That's why entire component tree must be serializable - i.e., may not contain non-serializable objects. And, if such an object happens to be referenced at the moment of serialization by a field somewhere in a component tree, an `WicketRuntimeException` will be thrown. Unfortunately, in Tomcat stack trace displayed is not very helpful and the only useful information there is the class of `WebPage` that contains a non-serializable somewhere in the component tree. This faulty object may be found using the exclusion method - by removing components one by one and checking after each removal if the exception is no longer thrown.

Final variables

There is a pitfall concerning final variables and serialization. Consider the following fragment:

```
public MyPanel(String id)
{
    super(id);

    // ...

    final NonSerializableObject nonSerializableObj = getNSO();

    // ...

    Link link = new Link("myLink")
    {
        public void onClick()
        {
            if (nonSerializableObj.getSomeBooleanProperty())
            {
                // ...
            }

            // ...
        }
    }

    // ...
}
```

Everything looks fine at the first sight, but let's look closer at how it works:

1. `nonSerializableObj` is actually inaccessible from our `Link` anonymous subclass
2. To make it accessible, compiler:
 - a. Creates an anonymous field of the `NonSerializableObject` class and an accessor to it
 - b. Assigns value to it simultaneously with assignment to `nonSerializableObj`
 - c. Makes `nonSerializableObj` inside the `Link` anonymous subclass a local variable
 - d. Initializes this local variable through forementioned accessor
3. That's why after compilation we have `nonSerializableObj` stored in a `FIELD` instead of `LOCAL VARIABLE`
4. And because it is non-serializable, serialization of entire component tree fails

How this can be worked around? Extract this property as a local variable and reference it inside the inner class instead of the entire object. If this property is a `Serializable` then its conversion to field won't do any harm and therefore serializableness will be preserved.