

Re: XStream versus XMLEncoder

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.help/2007-12/msg00083.html>

- *From:* apm35@xxxxxxxxxxxxxxxxxxxxx
 - *Date:* Thu, 6 Dec 2007 00:52:02 -0800 (PST)
-

On 5 Dec, 20:00, Lew <1...@xxxxxxxxxxxxxxxx> wrote:

This seems to me to be a basic POJO vs JavaBeans pattern discussion.
Both have their place.

In the ground of being behind Mark's advice is the notion that the class you wrote will be used.

Many of the gripes or suggestions for change with Java have to do with consumers of classes feeling like they don't have to worry about no steenkeen checked exceptions, or setters, or whatever. From the point of view of the *designers* of those classes is the concern that some client will, in fact, ignore that checked exception or call that setter. The designer may wish to close that loophole so that consumers of the class need not concern themselves with it, nor risk the consequences.

Exactly.

Thus, declare a checked exception;
consumer classes at least will not be surprised by them.

Sounds like I need to do this.

Make the class
members immutable; the client can ignore many threading issues and has no risk of constructed values suddenly, mysteriously changing to code for.

Sounds like I can't do that coz XMLEncoder will not let me.

Client programmers may grumble that they know perfectly well how to construct

Re: XStream versus XMLEncoder

an object completely before use, but just let their colleague forget to and they will curse the API writer's name for letting such a thing happen and how hard it is to debug and why, oh, why didn't they have the sense to make that member final?

—

Lew