Re: Column numbers in stack trace − enhancement request

Source: http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2007−01/msg01603.html

- *From*: "Sasi" <kvsasi@xxxxxxxxxxx>
- *Date*: 15 Jan 2007 06:02:24 −0800

When I encounter these kinds of issues during development, I wouldn't even need a line number to debug the problem. Unfortunately, not all bugs are so nice. Some decide to show up only in production deployment and are not reproducible in the dev environment. In those cases, it will be very helpful to know which exact method faltered so that you can work the logic backwards and figure out what could have gone wrong.

Robert Klemme wrote:

> On 15.01.2007 13:16, Sasi wrote:
>
> > I filed the following enhancement request to Sun. Would like to hear opinion about how useful implementing this feature would be.
> >
> > Synopsis: Need column numbers in stack traces
> > Description:
> > A DESCRIPTION OF THE REQUEST :
> > Stack traces contain only line numbers and in certain cases line number alone is not sufficient for figuring out where exactly an exception occurred. Consider the following line of code.
> >
> > value = getItem().getRelatedItem().getName().getValue();
> >
> > If the above line throws a NullPointerException, we have no clue whether it is the getItem, getRelatedItem or the getName that is returning a null value. So providing just the line number is not sufficiently helpful in narrowing down the problem. If the stack trace also contains the column number where the null was encountered, it will be really helpful.
> >
> > Though the above code could be rewritten to several lines so that we can clearly identify which method returned null, there are tons of such existing code and changing them all will be an unreasonably complex task.
>
> You can as well debug your app.

robert

.