

4. Using from SSJS

To log a caught error, the following options are available:

- **`openLogBean.addError(Exception, Component)`**: pass the error object from your try/catch and the component. To avoid hard-coding, use *this* from a property or **`this.getParent()`** from an event (so

```
try{
  ...
} catch(e) {
  openLogBean.addError(e,this);
}
```

If no component is available, pass *null*).
- **`openLogBean.addError(Exception, Component, int)`**: the extra parameter here is an integer, 1 to 7, corresponding to the Java Level logged. 1 is the most severe, 7 the least.
- **`openLogBean.addError(Exception, Component, int, string)`**: the extra parameter here is the UNID of a NotesDocument to pass. It is not good practice to store Notes objects in a bean, so only pass the UNID. The code will attempt to retrieve the document from the current database. If that cannot be done the UNID will be added to the error logging message instead.
- **`openLogBean.addError(Exception, string, Component)`**: the new second parameter is an extra message as a string. This provides the equivalent to `logErrorEx`. If there is no error, pass *null*.
- **`openLogBean.addError(Exception, string, Component, int)`**: passing error object, additional message, component and error level.
- **`openLogBean.addError(Exception, string, Component, int, string)`**: passing error object, additional message, component, error level and NotesDocument UNID.

To log an event, the following options are available:

- **`openLogBean.addEvent(string, Component)`**: the first parameter is a message, the second parameter is the component to log against.
- **`openLogBean.addEvent(string, Component, int)`**: the extra parameter is a warning level, 1 to 7, as in `addError()`.
- **`openLogBean.addEvent(string, Component, int, string)`**: the extra parameter is the UNID of a NotesDocuemnt to pass, as in `addError()`.