

Error reporting and logging

Errors

Profound.js provides built-in error reporting for application errors. By default, application errors, including call stack information, are reported both to the user and written to an error log.

The user will see errors in a dialog that looks like this:



The error log will show similar information. It will include the relevant session id and IBM i job information if the program is called from an IBM i Genie session

```
2017-05-04 17:05:27 | Session 7147825B34B29A13CACDE127FB33F26005DF114F93A593A1E6D8678AC4BC02C7 |
User ALEXR | Job QPADEV0006 | Job# 053842
TypeError: console.logg is not a function
  at switchTurn (C:\Users\Alex\pjsdemo17\modules\PJSSAMPLES\connect4.js:30:9)
  at drop (C:\Users\Alex\pjsdemo17\modules\PJSSAMPLES\connect4.js:68:7)
  at Object.connect4 (C:\Users\Alex\pjsdemo17\modules\PJSSAMPLES\connect4.js:12:37)
  at Object.profound.call (C:\Users\Alex\pjsdemo17\node_modules\profoundjs\api\call.jse:155:25)
  at C:\Users\Alex\pjsdemo17\node_modules\profoundjs\server\controller.jse:111:23
```

Logs

Profound.js uses STDERR and STDOUT for logging. If you are running the Profound.js server on IBM i, both the error logs and the standard output logs can be found in IFS directory `/profoundjs-base/instances/instance_name/logs`, where *instance_name* is the name of the installation instance.

To write to standard output, use `console.log()` in your JavaScript code.

You can also enable detailed runtime logging (rlog) for debugging purposes in an isolated development environment. This is done by passing the `-rlog` command line parameter to `start.js` when starting a Profound.js instance. The runtime log will log to standard output every line of code executed and the field values affected by it.