
Question

[Ward De Backer](#) · Oct 26, 2022

IRIS Native api for Node.js: all calls synchronous?

Hi,

If I test the [Native api for Node.js](#) from the documentation, I noticed (if I'm correct) all methods and calls are synchronous. By default due to the nature of Node.js, there is only one thread of execution and normally all JavaScript methods and all calls should be asynchronous and use either a callback function (the "old way") or promises or the async/await construct to return their result, e.g.:

- `myFunction(params, callbackFunction(response))`
- `myFunction(params).then(resolveFunction(response))`
- `async someFunction() { response = await myFunction(params) }`

See also [this post explaining the concepts](#).

In the current Native api, I only found methods and functions returning their result synchronously. For a low-load or single user application, this is not a big issue. But for high-load or multi-user applications, as each synchronous Native api call blocks the Node.js execution thread, this can become a severe performance issue. E.g. if you call a class method which performs a longer running embedded SQL query, all users would be blocked until the method call returns.

Am I overlooking asynchronous method variants in the Native api?

Thanks for clarifying!

[#JavaScript](#) [#Node.js](#) [#Performance](#) [#InterSystems](#) [IRIS](#)

Product version: IRIS 2022.1

Source URL: <https://community.intersystems.com/post/iris-native-api-nodejs-all-calls-synchronous>