

inter-process-messaging

[Source code of released version](#) | [Source code of development version](#)

Support for sending messages between a parent process and a child process that was spawned (by the parent) with [an IPC channel](#).

Usage

After spawning a child process, calling `enable(childProcess)` enables `childProcess.sendMessage(topic, payload)`, which delivers `payload` to any listeners in the child process that were registered for the given `topic` string.

Child processes register themselves to receive messages by calling `onMessage(topic, callback)`. The `callback` function may return a `Promise`, in which case the parent will receive an array of callback results from all children that subscribed to the given `topic`, after the results have been resolved.

Note that both `payload` and the response values from the child process listeners will be serialized for delivery between processes, though [the serialization library](#) used by this code is more robust than JSON, so it can handle circular and repeated references, `Set` s and `Map` s that contain themselves, etc.

The key features that differentiate this API from the native Node `childProcess.send(message)` and `childProcess.on("message", callback)` APIs are (1) restricting messages to a particular topic string, (2) more robust serialization logic thanks to `arson`, and (3) reliable responses from all listener callbacks in the child process, which lets the parent process proceed with confidence that the child is done with its work.