Atitit node.js按行读取同步功能实现

# **async-stream-reader**

**[IMG_256](https://npmjs.org/package/async-stream-reader)** **[IMG_257](https://travis-ci.org/viRingbells/async-stream-reader)** **[IMG_258](https://codecov.io/github/viRingbells/async-stream-reader?branch=master)**

An async way to read stream

## **Why**

Sometime you may want to read something from a stream, typically a file stream, in an async-await style. This tool may help you by providing an async read method for streams.

The async-stream-reader holds events and returns them one by one every time async StreamReader#next get called. If async-stream-reader has got some events in the holder, it will try to pause the input stream, and resume it if the holder get empty.

So as long as the input stream supports pause and resume, you can just do your work at each frame of data at ease, without worring missing something, or memory consumed in large quantities

    //     const fs            = require('fs');

    // const readline      = require('readline');

    const StreamReader = require('async-stream-reader');

    const rl = readline.createInterface({

        input: fs.createReadStream(file)

    });

    const reader = new StreamReader(rl, {

        events: { data: 'line', end: 'close' },

    });

    var line;

    while (line = await reader.next()) {

        try {

            logger.info(line);

            let rzt = await query(connection, line)

            logger.info(rzt);

        } catch (e) {

            logger.error(e);

        }

    }

async-stream-reader - npm