

aardio 范例: 调用 curl 命令

```
//调用 curl 命令
import console;

//默认调用 Win10 1803 及之后系统自带的 curl 。
//改为 import process.curl.7.81 可兼容到 XP 系统
//XP, Win7 在市场上已经接近消失, 现在开发软件再处处考虑这些已经不重要了。
import process.curl;

/*
curl 文档:
https://quickref.me/zh-CN/docs/curl.html
https://curl.se/docs/manual.html
https://curl.se/docs/manpage.html
*/

//下载网页, 成功返回网页, 失败返回 null, 错误代码, 错误代码参考: https://everything.curl.dev/usingcurl/returns
var data = process.curl("https://www.aardio.com")
console.log( data );

//支持所有 curl 参数
var data = process.curl(`-X POST
-d "{\"username\": \"jacen\", \"password\": \"123456\"}"
-H "Content-Type: application/json"
http://httpbin.org/anything/test `);

//返回 JSON 对象或数组时, 会自动解析为 aardio 对象或数组
console.dumpJson(data)

/*
可以用逗号分隔为多个参数, 这样的好处是可以直接写原始字符串,
aardio 会自动处理参数转义, 在必要时会自动在参数首尾添加双引号。
多个参数会自动以空格分隔, 并合并为单个命令行参数
*/
var data = process.curl("-X", "POST",
    "-d", '{"username": "jacen", "password": "123456"}',
    "-H", "Content-Type: application/json",
    "http://httpbin.org/anything/test",
)
console.dumpJson(data)

/*
也可以用一个数组或表指定任意个数参数, 支持命名参数。
注意数组参数总是会被移到命名参数后面。
所有基于 process 或 process.popen 的对象都支持上述这几种命令行参数写法
*/
var data = process.curl({
    "-X"="POST",
    "-d"='{"username": "jacen", "password": "123456"}',
    "-H"="Content-Type: application/json",
    "http://httpbin.org/anything/test"
})
console.dumpJson(data)

//--json 如果直接指定表对象, aardio 会自动转换为 json 文本。
var data = process.curl({
    "--json" = {
        username = "jacen";
        password = "123456";
    }
    "http://httpbin.org/anything/test"
})
console.dumpJson(data)

//-d 如果直接指定表对象, aardio 会自动转换为字符串, 并自动处理 URL 编码。
var data = process.curl({
    "-d" = {
        username = "jacen";
        password = "123456";
    }
    "http://httpbin.org/anything/test"
})
console.dumpJson(data)

/*
更推荐使用 web.rest (体积小, 基于系统自带组件, 用法更简单)
```

```
https://mp.weixin.qq.com/s/4mYRDnO49alwpQoBD_cILg
```

或使用 `inet.http` （体积小，基于系统自带组件，用法更简单）

```
https://mp.weixin.qq.com/s/3Xp4c1LxsOQJsux5o8bhvA
*/
```

```
console.pause(true);
```

[Markdown 格式](#)