


```
function jsonp({url, params, cb}) {
    return new Promise((resolve, reject) => {
        let script = document.createElement("script");

        params = {...params, cb};
        let arr = [];
        for(let key in params){
            // 属性名和属性值以等号方式连接在一起; 并且放入到数组中;
            arr.push(`${key}=${params[key]}`);
        }

        //console.log(arr); // [[
        script.src = `${url}?${arr.join("&")}`;
        document.body.appendChild(script);
    })
}
```

- 使用

```
jsonp({
    url: "",
    params: {},
    cb: "show"
}).then(function(data) {
    // data 就是跨域请求的数据;
    console.log(data);
})
```

- node 部分

```

let express = require("express");
let app = express();
// app.get : 前端是get请求; 请求的路由是/getData;
app.get("/getData",function (req,res) {
    // req: 请求体 res : 响应体;
    let {cb} = req.query;
    //console.log(cb);
    // res.end :
    res.end(`show("我喜欢你们")`)
})
app.listen(3000);

```

cros 方法

- node部分

```

let express = require('express');
let app = express();
//
/*app.use(express.static(__dirname))*/
// app.use: express 的一个中间件;
// 会在所有的返回数据之前对当前的请求进行处理;
app.use(function (req,res,next) {
    // 设置允许的访问的url的源;
    res.setHeader("Access-Control-Allow-Origin",
    "http://localhost:63342");
    // put 需要重新设置;
    // 允许put的请求方式
    res.setHeader("Access-Control-Allow-Methods","PUT");
    // 允许设置请求头;
    res.setHeader("Access-Control-Allow-Headers","name");
    // 下一步;
    next();
});

```

- 客户端

```

function getHello() {
    var xhr = new XMLHttpRequest();
    xhr.open("post", "http://b.example.com/Test.ashx", true);

```

```
xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
// 声明请求源
xhr.setRequestHeader("Origin", "http://a.example.com");
xhr.onreadystatechange = function () {
  if (xhr.readyState == 4 && xhr.status == 200) {
    var responseText = xhr.responseText;
    console.info(responseText);
  }
}
xhr.send();
}
```