

Nood.js C++ 模块 addon插件（三） - 程序园

 voidcn.com /blog/zhuzg1991/article/p-4356838.html

前面一篇介绍了一些基础，现在我们来看看官方的V8接口文档。

我们先来看看简单的hello world例子

```
#include <node.h>
#include <v8.h>

using namespace v8;

Handle<Value> Method(const Arguments& args) {
    HandleScope scope;
    return scope.Close(String::New("world"));
}

void init(Handle<Object> target) {
    target->Set(String::NewSymbol("hello"),
        FunctionTemplate::New(Method)->GetFunction());
}

NODE_MODULE(hello, init)
```

这个一个模板形式：

一个初始化函数一个node_module

```
void Initialize (Handle<Object> target);
NODE_MODULE(module_name, Initialize)
```

接下来是第二段例子 add相加的方法：

```

#define BUILDING_NODE_EXTENSION
#include <node.h>

using namespace v8;

Handle<Value> Add(const Arguments& args) {
    HandleScope scope;

    if (args.Length() < 2) {
        ThrowException(Exception::TypeError(String::New("Wrong number of
arguments")));
        return scope.Close(Undefined());
    }

    if (!args[0]->IsNumber() || !args[1]->IsNumber()) {
        ThrowException(Exception::TypeError(String::New("Wrong arguments")));
        return scope.Close(Undefined());
    }

    Local<Number> num = Number::New(args[0]->NumberValue() + args[1]-
>NumberValue());
    return scope.Close(num);
}

void Init(Handle<Object> target) {
    target->Set(String::NewSymbol("add"),
        FunctionTemplate::New(Add)->GetFunction());
}

NODE_MODULE(addon, Init)

```

是不是看到了一些新的内容比如字符串转换String::New("Wrong arguments");把c++字符串转成js字符串
args[0]->NumberValue()把js的Number转成C++可用的double

V8接口Local这种类型都是用来将c++的数据传输给js，

V8命名空间下的一些类，比如String，Date都是用来将C++的数据转换成输出给js用的，当然这些类都是继承Value类

感觉模块的工作流程就是这样的，接受node端的参数，如果要运算则变成C++的数据类型进行运算，然后再将计算完成的数据转化成js的数据类型返回给js端

上面的js调用：

```

var addon = require('./build/Release/addon');

console.log( 'This should be eight:', addon.add(3,5)
);

```

下面继续看例子：

实现一个js的回调函数参数为hello world

```

#define BUILDING_NODE_EXTENSION
#include <node.h>

using namespace v8;

Handle<Value> RunCallback(const Arguments& args) {
    HandleScope scope;

    Local<Function> cb = Local<Function>::Cast(args[0]);
    const unsigned argc = 1;
    Local<Value> argv[argc] = { Local<Value>::New(String::New("hello world"))
};
    cb->Call(Context::GetCurrent()->Global(), argc, argv);

    return scope.Close(Undefined());
}

void Init(Handle<Object> target) {
    target->Set(String::NewSymbol("runCallback"),
        FunctionTemplate::New(RunCallback)->GetFunction());
}

NODE_MODULE(addon, Init)

```

js端调用代码：

```

var addon =
require('./build/Release/addon');
addon.runCallback(function(msg) {
    console.log(msg);
});

```