

三步搭建你的手机/Web 应用后端服务



交大黑客松IBM问答群



该二维码7天内(5月12日前)有效, 重新进入将更新

在开发手机/Web 应用的时候, 前台客户端和后台服务器端一定需要通信, 而目前最流行的方式是采用 RESTful API 的方式进行。Restful API 是一种简单轻量的基于 HTTP 协议的前后台之间的通信标准, 采用 JSON 数据格式。它分离了前后端开发, 前后端之间不必一定使用同一种的语言, 便于更好的协同。关于 Restful 的更多信息, 可以参考:

<http://baike.baidu.com/item/RESTful>

<http://www.cnblogs.com/springyangwc/archive/2012/01/18/2325784.html>

IBM Bluemix 上的 Apache OpenWhisk 平台, 提供了一种便捷的搭建后端服务的方式, 不需要你配置任何虚拟机, 安装任何 Web 服务器和应用服务器, 就可以将你要实现的逻辑, 暴露为可以访问的 Restful API 的形式。

前提条件

本文的前提条件是你已经拥有了 Bluemix 的账号, 并且在本机上安装好了 OpenWhisk 客户端。如果您需要 Bluemix 账号, 请到 IBM 展台前寻求帮助。

下载 OpenWhisk 客户端可执行文件, 具体请参考

<https://console.ng.bluemix.net/openwhisk/learn/cli>。

好了, 下面就可以开发部署你的后台服务了。

第一步: 实现你的逻辑代码

将程序实现你的逻辑代码, 并在本地调试通过。可选的编程语言有: Java, JavaScript, Python, 和 Swift。

例如, 我这里有段代码, 实现了 HelloWorld 的功能, 由 JavaScript 实现。

```
function main(params) {  
    return {payload:'Hello,' + params.payload+'!'};  
}
```

将它存储为 hello.js 文件。

参考文档：

参考：

<https://github.com/openwhisk/openwhisk/blob/master/docs/actions.md>

更多 Action 实现的参考代码：

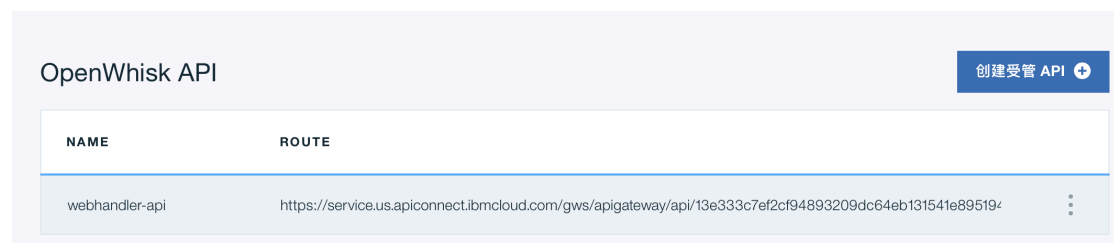
<https://github.com/openwhisk/openwhisk-catalog/tree/master/packages/samples>

第二步：把你的逻辑代码创建为 OpenWhisk Action

```
$wsk action create myhelloworld <file_path>/hello.js
```

其中，myhelloworld 是 Action 的名字。

第三步：将 OpenWhisk Action 暴露为 Restful API

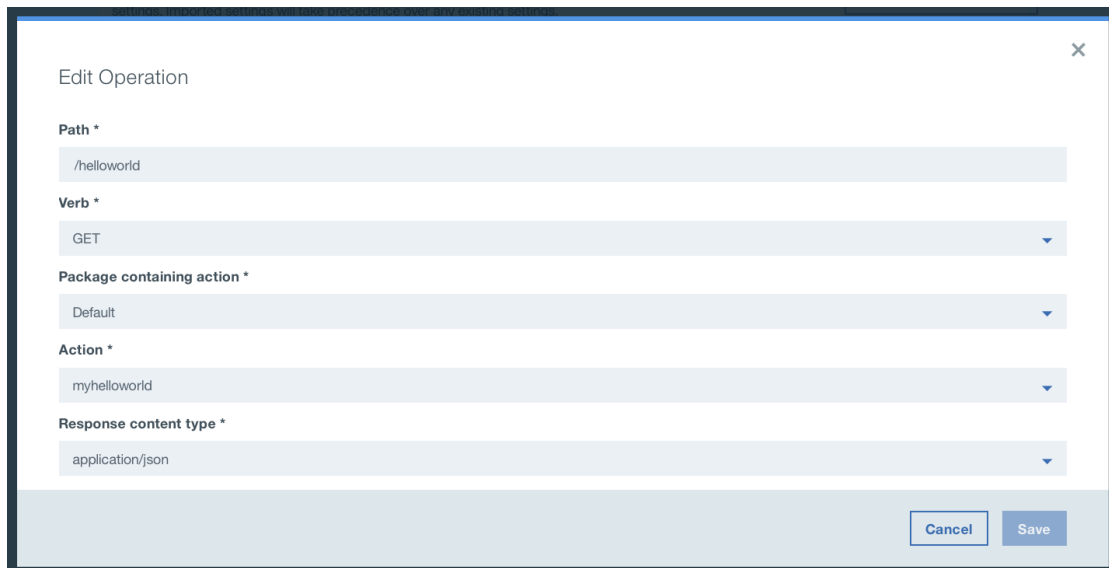


在 OpenWhisk 的 API 管理界面（见上图）中

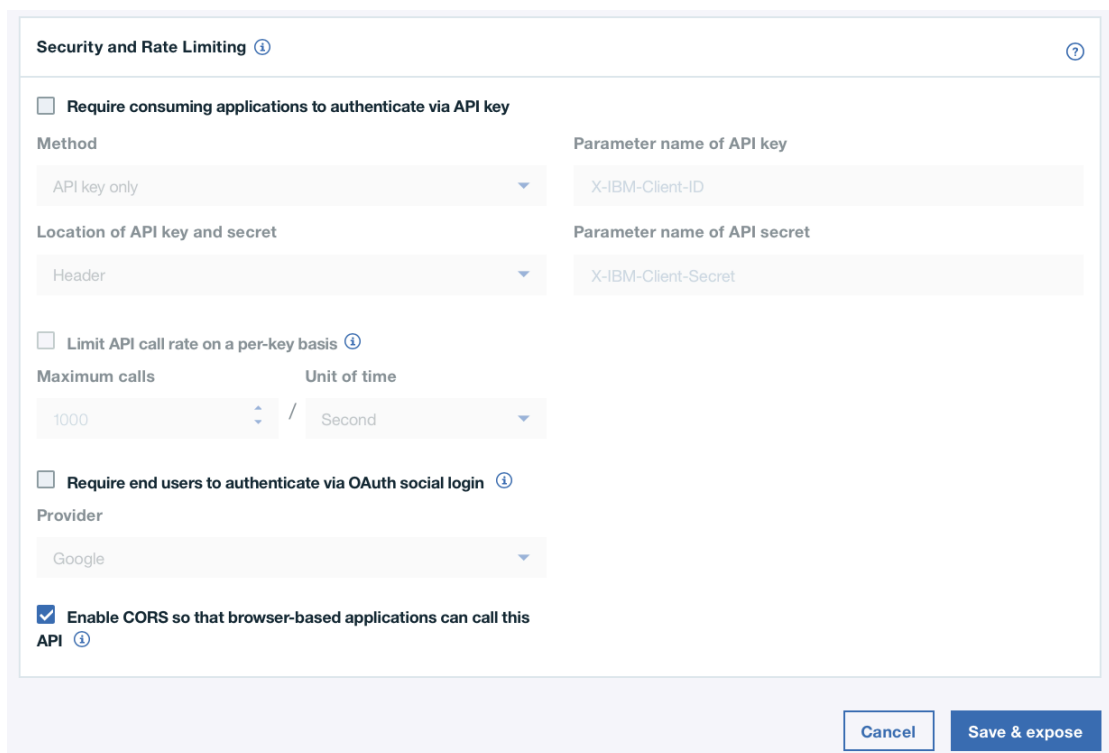
（<https://console.ng.bluemix.net/openwhisk/apimanagement>），点击“创建受管的 API”，就会进入“为 OpenWhisk 创建 API”的界面（见下图）。



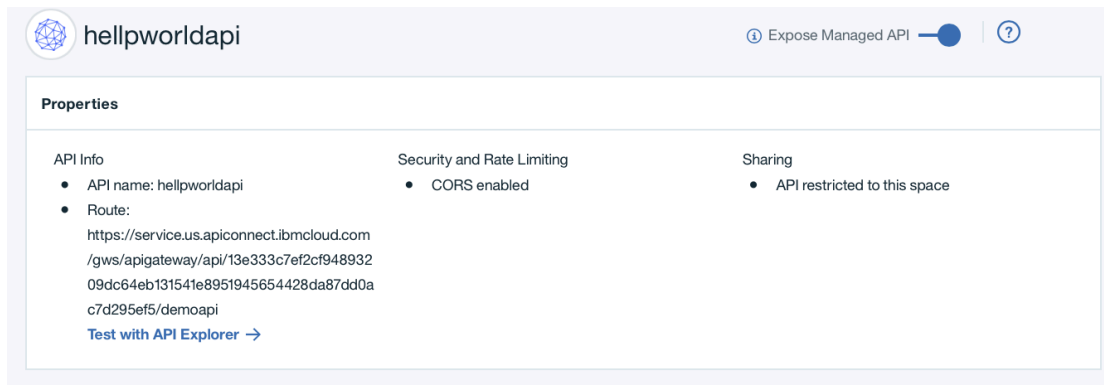
填入 API name，Base path for API（见上图）。点击 Create operation，会弹出 Edit Operation 界面（见下图）。



填写 Path，选择你刚才创建的 Action，点击 Save。又回到“为 OpenWhisk 创建 API”的页面。“Security and Rate Limiting”这里可以使用 缺省设置，也可以按需配置（见下图）。点击 Save & expose。



那么你的 API 就创建好了。在下图中，可以看到访问该 API 的 URL。



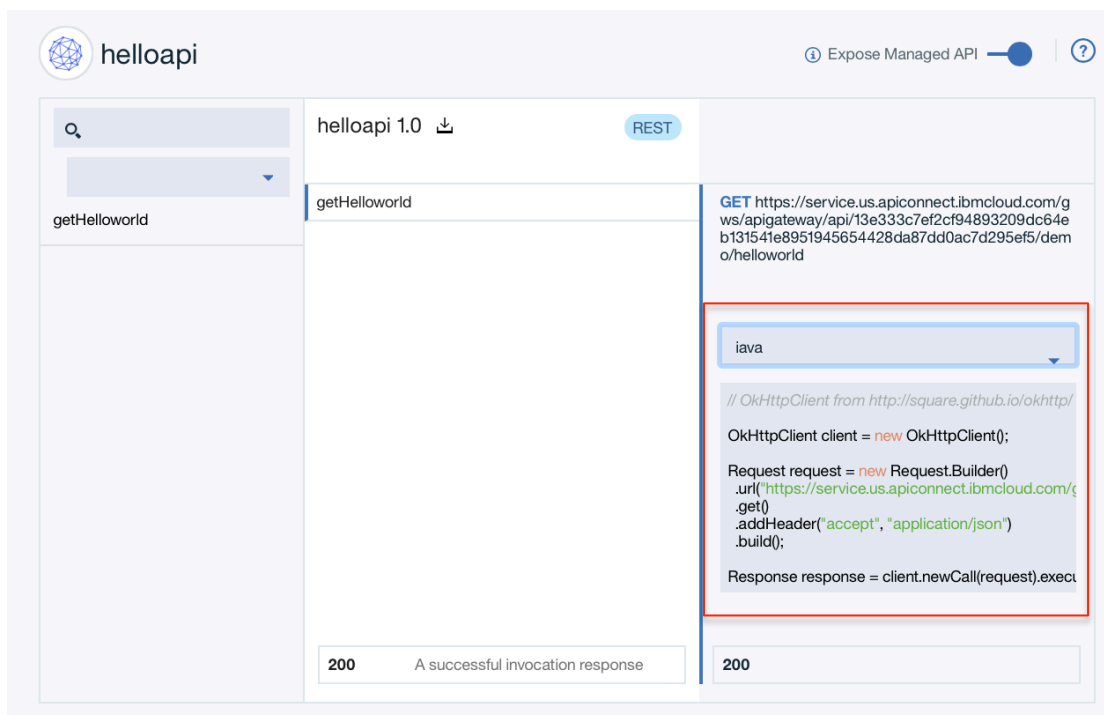
API 创建好之后，可以测试一下。

`curl --request GET --url`

<https://service.usapiconnect.ibmcloud.com/gws/apigateway/api/13e333c7ef2cf94893209dc64eb131541e8951945654428da87dd0ac7d295ef5/demoapi/hello-world?payload=daisy>

利用 API Explorer

对 RESTful API 的调用不熟悉？没关系，Test with API Explorer 会帮助你书写客户端代码。



在 API 摘要界面，点击"Test with API Explorer"，可以打开 API Explorer。在 API Explorer 的右侧，有一个下拉框，选择语言后，会给出该语言的调用例子，你参考这个例子，可以很容易的写出客户端调用代码。

高级功能：如何处理包依赖？

如果你的代码中有对第三方的库的依赖，你可以把你的 **action** 打包成一个 **zip** 文件然后上传到 **openwhisk** 当中，针对 **node.js**，你可以在你的 **action** 文件所在目录添加 **package.json**，里面添加你的包依赖，比如：

```
{
  "name": "my-action",
  "main": "index.js",
  "dependencies": {
    "left-pad": "1.1.3"
  }
}
```

然后运行 **npm install** 来安装所有依赖包。

之后，运行 **zip -r action.zip *** 来压缩所有文件。

然后通过 **wsk cli** 来上传你的 **action zip** 文件。

```
wsk action create packageAction --kind nodejs:6 action.zip
```

更详细的帮助文档，请参考：

<https://github.com/openwhisk/openwhisk/blob/master/docs/actions.md>