

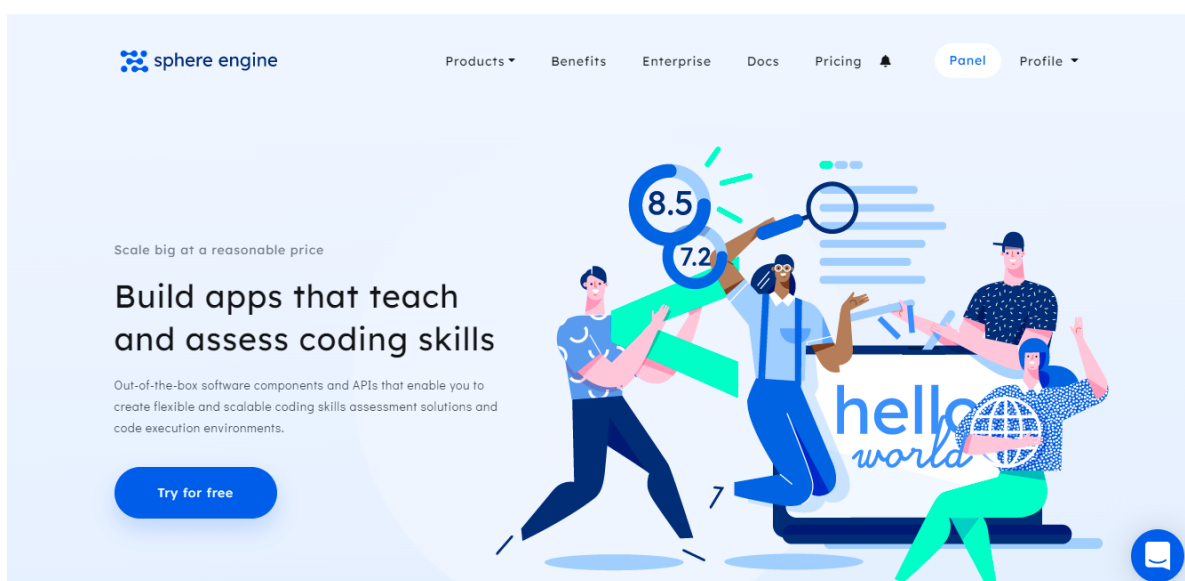
# Online Compiler API

学习Online Compiler主要是因为在做4001软工项目CUHKSZ-Stackoverflow的时候，想着如果用户输入在我们的平台上的代码能够运行就好了，听上去是挺fancy的一件事。后来在逛Github偶然看到了一些平台是专门有提供这样的online compiler的API的，只要我们把要编译的代码打包起来后给这些平台发送编译/运行请求，它们就会把运行的结果返回回来。

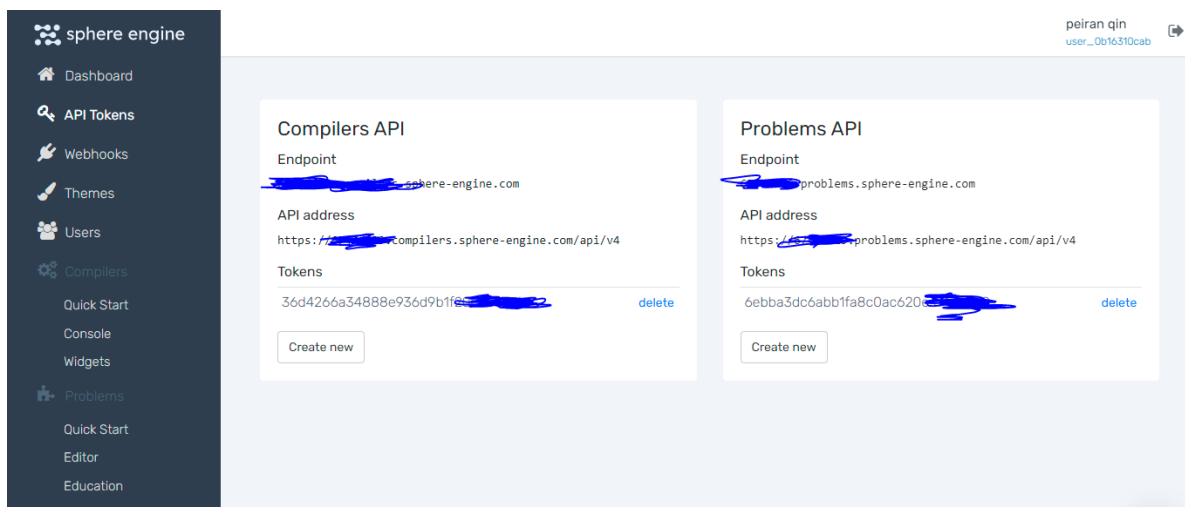
这篇笔记主要记录了我关于Sphere Engine (<https://sphere-engine.com/>) 的compiler API的学习笔记。

## 申请用户

- 第一步是在这个网站上注册账户：



- 点Try for free并申请账户后，就会获得一个Token和一个EndPoint, 可以理解为这是你使用这个平台的账号密码：



- 接着在官方文档中有很多的教程，讲得挺明白的：<https://docs.sphere-engine.com/compilers/api/quickstart>

## 上代码

- 这里主要是在springboot项目上用python来做演示：
- 相关的包：

```
from sphere_engine import CompilersClientV4
from sphere_engine.exceptions import SphereEngineException
import requests
import urllib.request
```

- 发送一个编译请求给网站：

```
source = request.POST['source_code']
# online compiling and running:

accessToken = '36d4266a34888e936d9b1f20c9e634b8'
endpoint = '6747df29.compilers.sphere-engine.com'

# initialization
client = CompilersClientV4(accessToken, endpoint)

compiler = 116 # default python
input = '111'

response = client.submissions.create(source, compiler, input)
```

- 首先 accessToken 和 endpoint 是刚刚申请完账户给的，用 client = CompilersClientV4(accessToken, endpoint) 是组装成一个申请的用户。
- compiler=116 是选择用什么语言的编译器，116代表运行python，13是代表用C。更多的语言详见：<https://sphere-engine.com/supported-languages>
- input 是指你想输入到程序里的变量
- 最后，通过 response = client.submissions.create(source, compiler, input) 发送运行的请求，并且得到返回的结果为response

返回的结果会是这个任务的Id(数据类型是一个字典)：

```
{
  "id": 42
}
```

- 接着我们需要不断检查这个任务的运行状态，直到运行完成：

```
task_id = response.get("id")
response = client.submissions.get(task_id)

executing_status = response.get("executing")
while (executing_status):
    response = client.submissions.get(task_id)
    executing_status = response.get("executing")
```

- 首先从刚刚返回的response中取出任务的id
  - 接着用 response = client.submissions.get(task\_id) 查看这个任务运行的状态（是否正在运行，时间，内存），用 response.get("executing") 把“是否正在运行”取出来。用 while 循环直到任务结束运行。
- 取出运行的结果：

```
uri=response.get("result").get("streams").get("output").get('uri')
```

- 这个平台返回的运行结果是个uri, 所以接下来还要去uri里取出具体的函数的output
- 取出结果, utf-8编码, 返回:

```
contents = urllib.request.urlopen(uri).read().decode('utf-8')  
  
return render(request, 'compile_result.html', context={'result':contents})
```

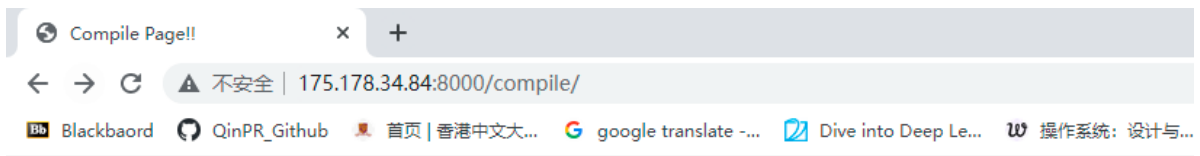
- python使用 `urllib.request.urlopen(uri).read()` 取出uri里的值
- 完整代码:

```
def compile(request):  
    source = request.POST['source_code']  
    # online compiling and running:  
  
    accessToken = '36d4266a34888e936d9b1f20c9e634b8'  
    endpoint = '6747df29.compilers.sphere-engine.com'  
  
    # initialization  
    client = CompilersClientV4(accessToken, endpoint)  
  
    compiler = 116 # default python  
    input = '111'  
  
    response = client.submissions.create(source, compiler, input)  
  
    task_id = response.get("id")  
    response = client.submissions.get(task_id)  
  
    executing_status = response.get("executing")  
    while (executing_status):  
        response = client.submissions.get(task_id)  
        executing_status = response.get("executing")  
  
    uri = response.get("result").get("streams").get("output").get('uri')  
  
    contents = urllib.request.urlopen(uri).read().decode('utf-8')  
  
    return render(request, 'compile_result.html', context={'result':contents})
```

- (要运行的源码source\_code是用post从前端发过来的)

---

## 效果



# This is a online compiler

Please write a python code in text-box below:

```
print("hello csc4001!!")

i = 0;
while (i < 11):
    print(i)
    i += 1;
```

提交

- 现在输入框写一个简单的python代码，提交运行



**hello csc4001!! 0 1 2 3 4 5 6 7 8 9 10**

- 运行结果成功返回了！