

11.6 通过XML-RPC实现简单的远程调用¶

问题¶

你想找到一个简单的方式去执行运行在远程机器上面的Python程序中的函数或方法。

解决方案¶

实现一个远程方法调用的最简单方式是使用XML-RPC。下面我们演示一下一个实现了键-值存储功能的简单服务器：

```
from xmlrpc.server import SimpleXMLRPCServer

class KeyValueServer:
    _rpc_methods_ = ['get', 'set', 'delete', 'exists', 'keys']
    def __init__(self, address):
        self._data = {}
        self._serv = SimpleXMLRPCServer(address, allow_none=True)
        for name in self._rpc_methods_:
            self._serv.register_function(getattr(self, name))

    def get(self, name):
        return self._data[name]

    def set(self, name, value):
        self._data[name] = value

    def delete(self, name):
        del self._data[name]

    def exists(self, name):
        return name in self._data

    def keys(self):
        return list(self._data)

    def serve_forever(self):
        self._serv.serve_forever()

# Example
if __name__ == '__main__':
    kvserv = KeyValueServer('', 15000)
    kvserv.serve_forever()
```

下面我们从一个客户端机器上面来访问服务器：

```
>>> from xmlrpc.client import ServerProxy
>>> s = ServerProxy('http://localhost:15000', allow_none=True)
>>> s.set('foo', 'bar')
>>> s.set('spam', [1, 2, 3])
>>> s.keys()
['spam', 'foo']
>>> s.get('foo')
'bar'
>>> s.get('spam')
[1, 2, 3]
>>> s.delete('spam')
>>> s.exists('spam')
False
>>>
```

讨论¶

XML-RPC 可以让我们很容易的构造一个简单的远程调用服务。你所需要做的仅仅是创建一个服务器实例，通过它的方法 `register_function()` 来注册函数，然后使用 `serve_forever()` 启动它。在上面我们将这些步骤放在一起写到一个类中，不够这并不是必须的。比如你还可以像下面这样创建一个服务器：

```
from xmlrpc.server import SimpleXMLRPCServer
def add(x,y):
    return x+y

serv = SimpleXMLRPCServer(('', 15000))
serv.register_function(add)
serv.serve_forever()
```

XML-RPC暴露出来的函数只能适用于部分数据类型，比如字符串、整形、列表和字典。对于其他类型就得需要做些额外的功课了。例如，如果你想通过 XML-RPC 传递一个对象实例，实际上只有他的实例字典被处理：

```
>>> class Point:
...     def __init__(self, x, y):
...         self.x = x
...         self.y = y
...
>>> p = Point(2, 3)
>>> s.set('foo', p)
>>> s.get('foo')
{'x': 2, 'y': 3}
>>>
```

类似的，对于二进制数据的处理也跟你想象的不太一样：

```
>>> s.set('foo', b'Hello World')
>>> s.get('foo')
<xmlrpc.client.Binary object at 0x10131d410>

>>> _.__dict__
{'data': b'Hello World'}
>>>
```

一般来讲，你不应该将 XML-RPC 服务以公共API的方式暴露出来。对于这种情况，通常分布式应用程序会是一个更好的选择。

XML-RPC的一个缺点是它的性能。SimpleXMLRPCServer 的实现是单线程的，所以它不适合于大型程序，尽管我们在 11.2小节中演示过它是可以通过多线程来执行的。另外，由于 XML-RPC 将所有数据都序列化为XML格式，所以它会比其他的方式运行的慢一些。但是它也有优点，这种方式的编码可以被绝大部分其他编程语言支持。通过使用这种方式，其他语言的客户端程序都能访问你的服务。

虽然XML-RPC有很多缺点，但是如果你需要快速构建一个简单远程过程调用系统的话，它仍然值得去学习的。有时候，简单的方案就已经足够了。