2.3SSL证书验证verify

Requests 可以为 HTTPS 请求验证 SSL 证书，就像 web 浏览器一样。要想检查某个主机的 SSL 证书，你可以使用 `verify` 参数：

```
>>> requests.get('https://kennethreitz.com', verify=True)
requests.exceptions.SSLError: hostname 'kennethreitz.com' doesn't match either of '*.herokuapp.com',
'herokuapp.com'
```

在该域名上我没有设置 SSL，所以失败了。但 Github 设置了 SSL：

```
>>> requests.get('https://github.com', verify=True)
<Response [200]>
```

对于私有证书，你也可以传递一个 CA_BUNDLE 文件的路径给 `verify`。你也可以设置`REQUEST_CA_BUNDLE` 环境变量。

如果你将 `verify` 设置为 False，Requests 也能忽略对 SSL 证书的验证。

```
>>> requests.get('https://kennethreitz.com', verify=False)
<Response [200]>
```

默认情况下， `verify` 是设置为 True 的。选项 `verify` 仅应用于主机证书。

你也可以指定一个本地证书用作客户端证书，可以是单个文件（包含密钥和证书）或一个包含两个文件路径的元组：

```
>>> requests.get('https://kennethreitz.com', cert=('/path/server.crt', '/path/key'))
<Response [200]>
```

如果你指定了一个错误路径或一个无效的证书：

```
>>> requests.get('https://kennethreitz.com', cert='/wrong_path/server.pem')
SSLError: [Errno 336265225] _ssl.c:347: error:140B0009:SSL routines:SSL_CTX_use_PrivateKey_file:PEM lib
```

警告

本地证书的私有 key 必须是解密状态。目前，Requests 不支持使用加密的 key。