

3.3 从资源类对象获取资源配置接口

测试资源的提供本质上是为了提供测试环境的配置信息，其中包括测试资源的配置接口的连接信息，比如串行口的波特率、SSH接口的IP地址、用户名和密码等，测试代码根据这些信息对测试资源进行连接。所以在测试资源配置实现之后，需要考虑如何通过这些资源配置来获取相应的配置接口。

3.3.1 资源类对象和配置接口分离

从面向对象的角度来说，当我们用类描述资源的时候，很容易会想到将一些对象的操作方法也封装到其中，然后直接在测试用例中调用。虽然看上去是合理的，但是从模块化的设计思想出发，我们仅仅希望一个模块负责单一的功能，而不是将很多逻辑操作都一股脑地封装在一起。

当一个团队设计了资源类的属性之后，如果将所有的逻辑操作都封装在ResourceDevice中，那么资源包就会需要很多外部依赖。比如一个设备，支持Telnet管理，就需要引用Telnet包，如果支持SSH管理，则需要引用SSH包，如果支持Web管理，就需要引用Selenium包。过多的依赖不仅让平台的管理变得麻烦，也会使部署变得不灵活，甚至显得太重。试想，如果另一个团队也想复用这个资源包，但是他们并不使用Web，没有安装Selenium，那么为了使用这个包，就需要安装Selenium——虽然意义不大，仅仅是为了解决一个组件的依赖。

配置接口是用来管理测试资源的接口，是代码用来向测试资源发送和接收信息的重要途径。对于一些简单的接口，比如基于文本协议的命令行Telnet、SSH、串行口等，可以使用相应的库进行封装，在调用的时候进行实例化。此外，还有很多复杂的管理接口，诸如需要测试资源设备供应商提供的连接控件、图形接口（GUI）等，需要实现更复杂的库。

但是无论这些接口的实现如何复杂，其实例化过程中所需要的参数都可以通过资源模块来提供，所以我们可以通过设计一个标准的配置接口来获取方法，将资源模块本身和真正实现接口的模块完全解耦合，如图3-10所示。

