

```

    } else {
        address = restClass.getSimpleName();
    }
    // 确保最前面只有一个 /
    if (!address.startsWith("/")) {
        address = "/" + address;
    }
    address = address.replaceAll("\\\\+", "/");
    return address;
}
}

```

现在一个与 SOAP 插件类似的 REST 插件就开发完毕了,实现过程与 SOAP 插件完全一致。这里只是开发了一个框架原型,我们可以根据实际项目的需要,对该框架进行必要的扩展。

也许您会思考一个问题:能否将 SOAP 插件与 REST 插件合并为一个插件?这个就仁者见仁智者见智了,我认为还是有必要分开的,因为这是两种类型的 Web 服务,最好能够提供不同的插件,让开发者自行选择所需的插件,并不将一些用不到的特性包含到项目中。

## 5.8 总结

在本章中,我们对框架做出了一定的扩展,对 Action 方法参数进行了优化,若不存在任何请求参数,则可在 Action 方法中省略 Param 参数。提供了文件上传特性,并支持文件批量上传。使用 ServletHelper 来封装 Servlet API,在 Controller 或 Service 中可直接调用 ServletHelper 封装的 Servlet 常用 API。此外,我们也封装了 Shiro 框架,开发了一个简单的安全控制模块,并通过插件的方式集成到 Smart 框架中。同样也通过插件的方式封装了 CXF 框架,针对 SOAP 与 REST 服务开发了对应的 Smart 插件。

对于 Smart 框架现在只是一个开始,我们的目标是打造一款轻量级的 Web 开发框架,并提供大量的插件扩展机制,未来还有很长的路要走。