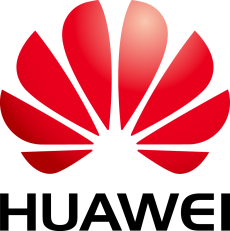
|  |  |  |
| --- | --- | --- |
| 华为技术有限公司 | 产品名称 | 密级 |
| NetworkOS | 内部公开 |
| 产品版本 | 共46页 |

OPS2.0 开放性增强Eclipse编程插件操作手册

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 拟制: | 张鹏宇 |  | 日期： | 2013-7-27 |
| 审核: |  |  | 日期： |  |
| 审核: |  |  | 日期： |  |
| 批准: |  |  | 日期： |  |



**华为技术有限公司**

版权所有 侵权必究

目 录

[OPS2.0 开放性增强Eclipse编程插件操作手册 1](#_Toc397181794)

[1 总体说明 4](#_Toc397181795)

[1.1 软件要求 4](#_Toc397181796)

[1.2 范围 4](#_Toc397181797)

[1.3 发布目录结构 5](#_Toc397181798)

[2 安装向导 6](#_Toc397181799)

[2.1 Eclipse安装要求 6](#_Toc397181800)

[2.2 安装OPSDEV for Java 6](#_Toc397181801)

[2.3 安装OPSDEV-Java 12](#_Toc397181802)

[3 使用指南 14](#_Toc397181803)

[3.1 创建OpsDev-java工程 14](#_Toc397181804)

[3.2 修改OpsDev-Java工程属性 20](#_Toc397181805)

[3.3 操作OpsDev-Java Restful Api 22](#_Toc397181806)

[3.4 在代码中进行动态生成body（add post body）操作 26](#_Toc397181807)

[3.5 网络监视器 33](#_Toc397181808)

[3.6 数据源管理 38](#_Toc397181809)

# 总体说明

## 软件要求

|  |  |
| --- | --- |
| 软件名称 | 软件版本 |
| Microsoft Windows | XP SP2，SER2003及其他兼容 |
| Eclipse | Eclipse-sdk-3.7.1版本 |
| JRE | 1.6.0\_11-b03 |
| OpsDev | OpsDev for Eclipse 1.0.0.2014.52314574710 版本 |

重要说明:

插件提供OPSDEV\_Java版本用户使用。

## 范围

VRP OPS2.0 开放性增强Eclipse编程插件, 提供给华为VRP网络业务编程者的编辑器环境。主要功能:

可视化的IDE编程环境，支持java编程环境 。

进行OPS数据源管理

主要完成对数据源的添加，修改，删除，配置相关数据源。

进行数据接口加载。

对数据接口进行配置加载。

提供网络监视器。

提供对OPS中涉及网络访问的监视功能。

Java IDE辅助编程提示。

主要完成对java编辑，可视，调试，编译的功能的相关提示。

由ops规格schema文件，自动生成restfulApi文档。

主要完成接口代码的生成功能。

提供对api路径编程提示功能。

## 发布目录结构

OpsDev插件发布目录结构：

--

|---- com.huawei.networkos.ops\_1.0.0.2014060416140410.jar

|---- com.huawei.networkos.ops.core\_1.0.0.2014060416140410.jar

(OpsDev)

|---- com.huawei.networkos.ops.java\_1.0.0.2014060416140410.jar (OpsDev\_java)

# 安装向导

## Eclipse安装要求

下载安装Eclipse开发环境 ，推荐版本3.7.1，也可以选择版本号在3.7.1以上的版本。

## 安装OPSDEV for Java

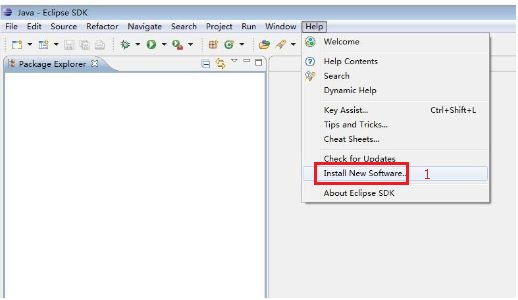
将PyDev 2.7.5.zip解压到eclipse的plugins目录中。

安装OPS2.0，打开OPS2.0.exe文件根据提示点击下一步即可，直至安装完成。

安装完成后生成目录plugins文件夹中找到com.huawei.networkos.ops.feature-1.0.0.2014042218392610.zip。

打开eclipse开发环境，开始安装插件。

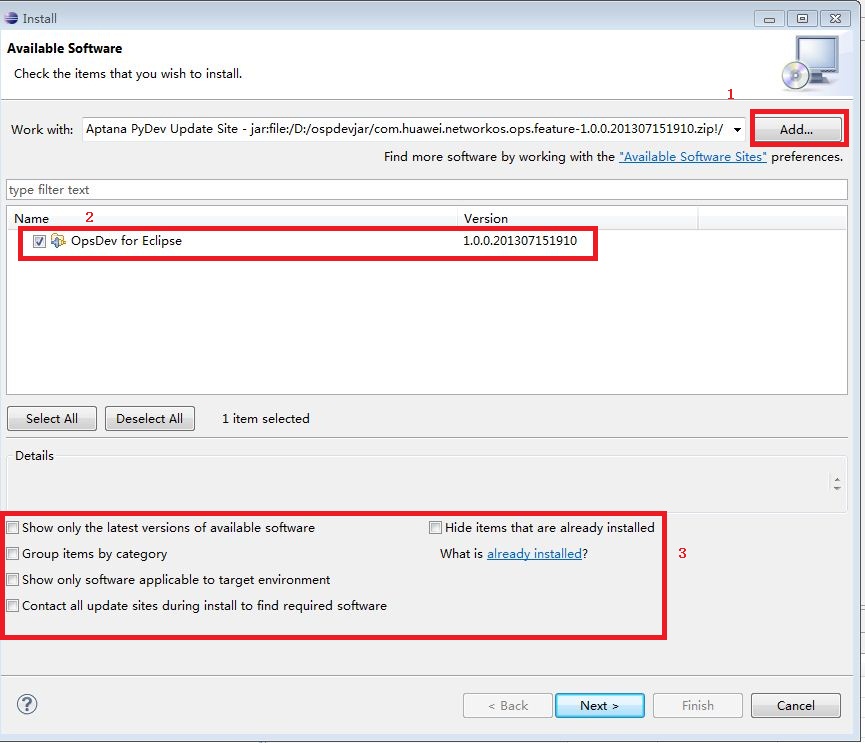
eclipse开发环境示意图



在菜单栏选择Help-->Install New Software，出现如图2-1所示界面。

点击Install New Software，出现如图2-2所示界面。

Install New Software界面示意图



增加OpsDev插件具体步骤如下：

Add按钮 ：com.huawei.networkos.ops.feature-1.0.0.2014042218392610.zip包加入到Work with路径框里，如图2-1所示红色框1。

钩选OpsDev for Eclipse，如图2-1所示红色框2。

如图2-1所示红色框3，五个选项不钩选。点击Next进入下一步。

图2-1所示红色框3中的解释说明：

Show only the latest versions of available software:

显示最终版本。

Group items by category:

插件项目类别总和。

Show only software applicable to target environment:

显示可用的软件。

Contact all update sites during install to find required software:

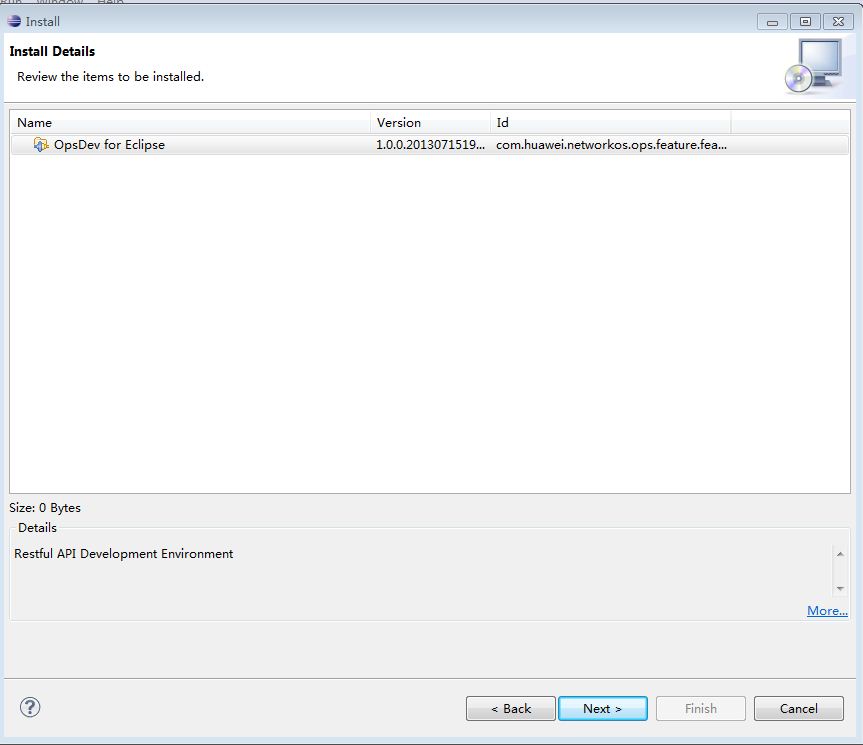
安装更新软件。

Hide items that are already installed:

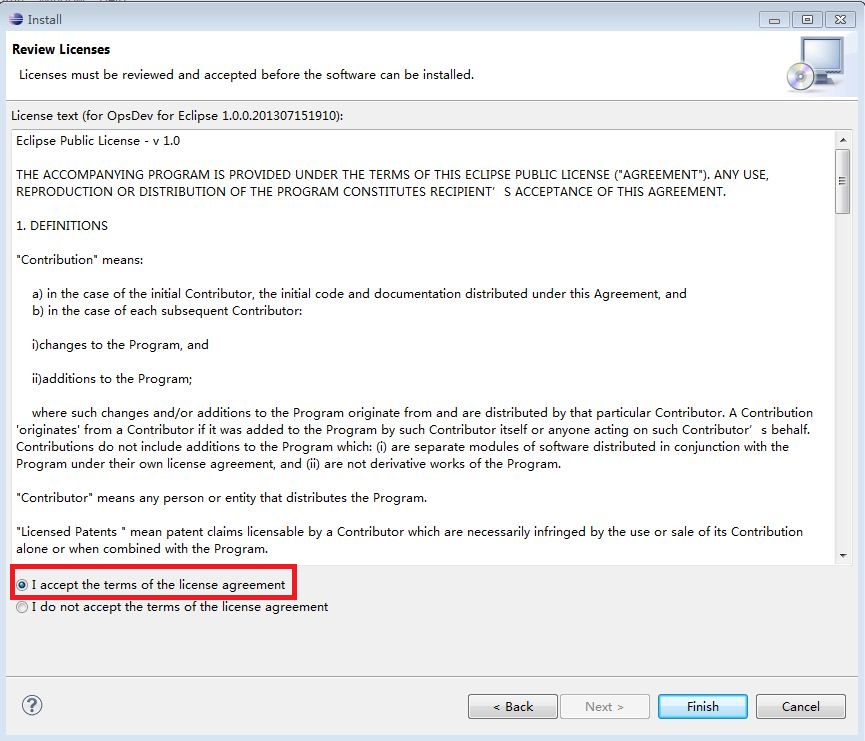
隐藏项目插件已安装。

进入安装信息界面，点击Next进入下一步，如图2-3所示。

安装界面示意图1



安装界面示意图2

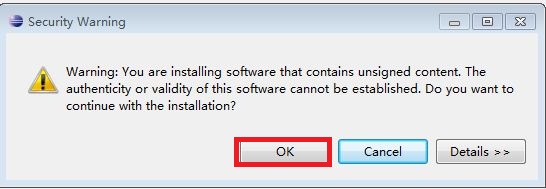


图表 4

如图2-4所示选择红色框，同意许可，点击Finish完成。

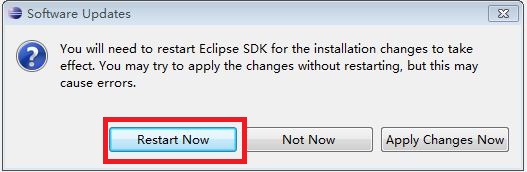
点击完成后，出现警告框，如图2-5所示，点击OK键，继续。

安装告警界面示意图



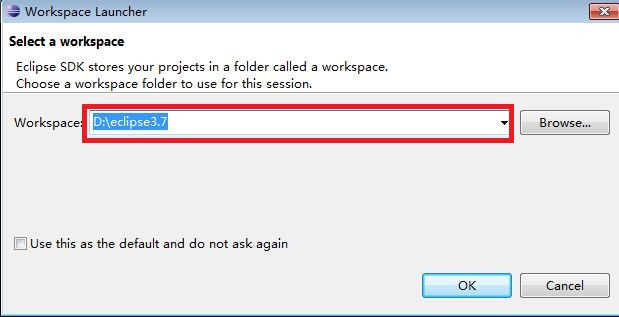
安装完成后，提示Eclipse需要重启。如图2-6所示，选择立即重启。

安装完成示意图



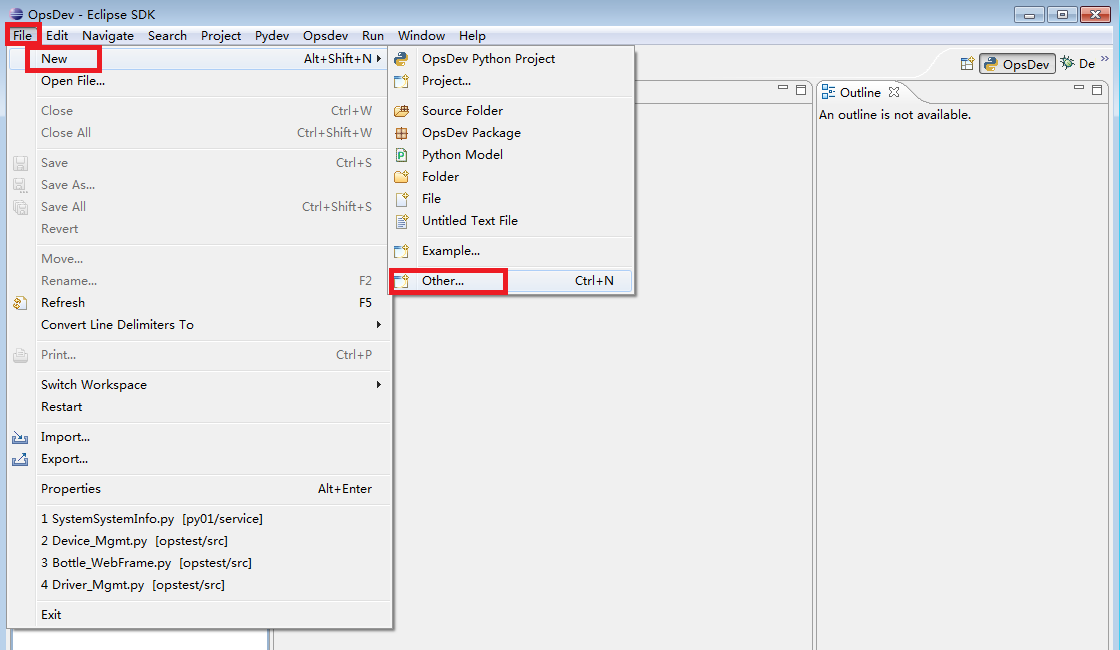
重新启动Eclipse后，选择工作路径，如图2-7所示。

选择工作路径示意图



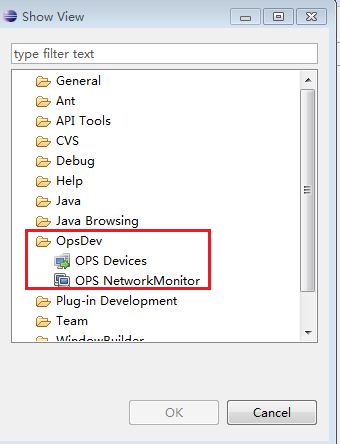
打开Eclipse，在菜单栏中选择File-->New-->Other，如图2-8所示。

OpsDev插件安装页面示意图



如果出现了OpsDev选项，说明安装成功，如图2-9所示。

OpsDev界面示意图



## 安装OPSDEV-Java

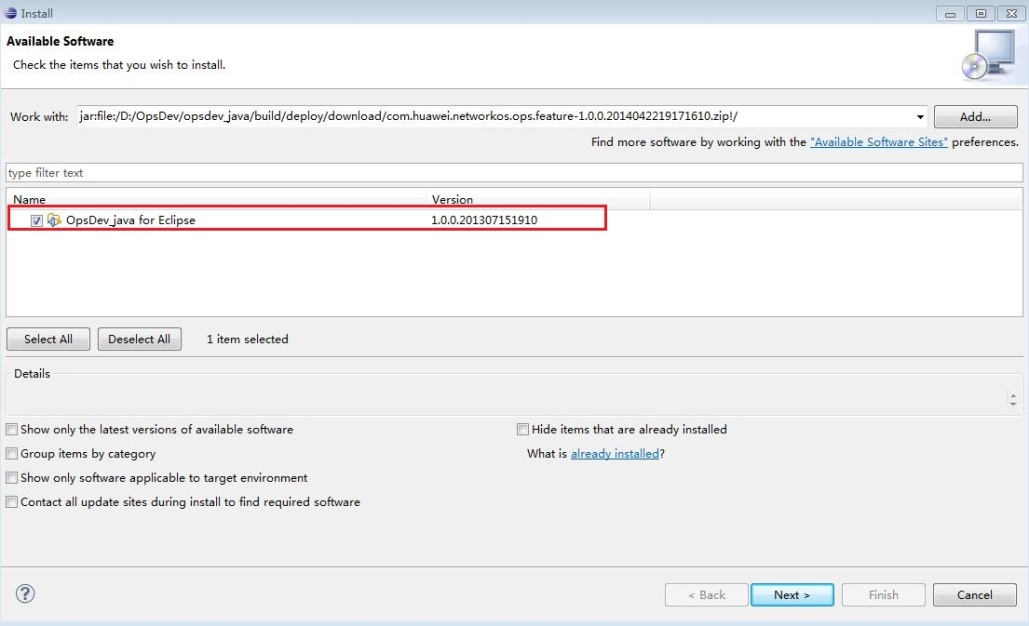
说明：

OpsDev-java的使用依赖于OPSDEV的安装。

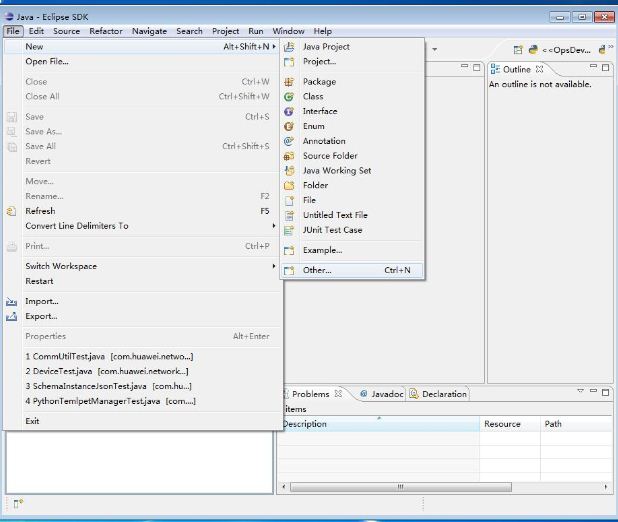
OpsDev-java安装步骤和OPSDEV的安装步骤一致，在这不做重复说明。

OpsDev-java安装，如图2-10图2-11图2-12所示步骤进行，出现图2-12所示界面时，说明安装成功。

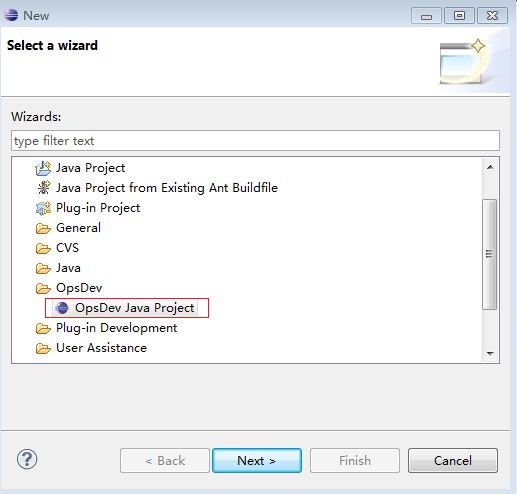
OpsDev\_Java安装示意图



OpsDev-java安装页面示意图



OpsDev-java 插件安装成功示意图

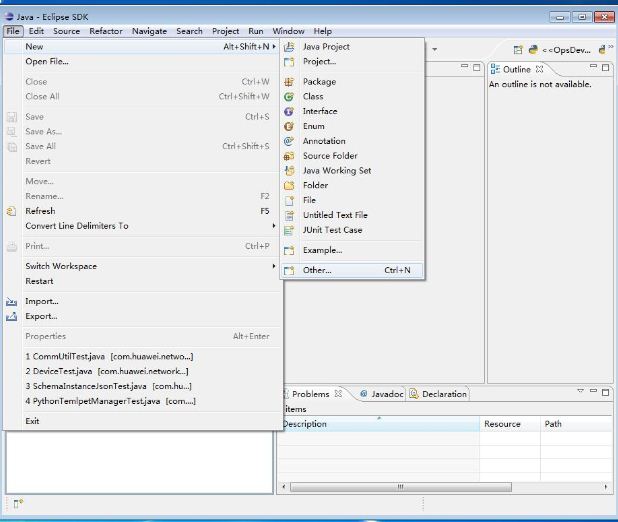


# 使用指南

## 创建OpsDev-java工程

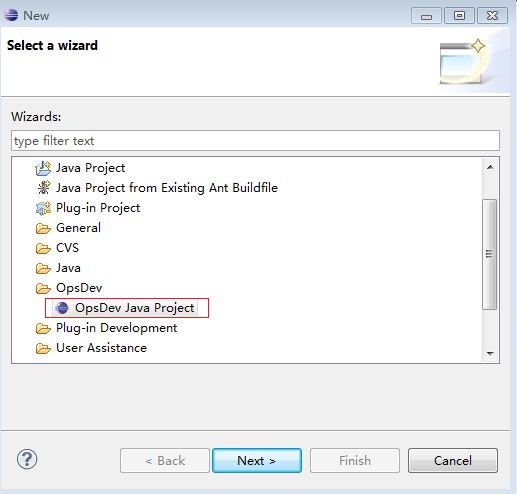
在Eclipse新增一个工程，选择OpsDev\_java Project。具体步骤为打开Eclipse，选择File-->New-->Other，如图3-1所示。

新增OpsDev-Java Project工程示意图



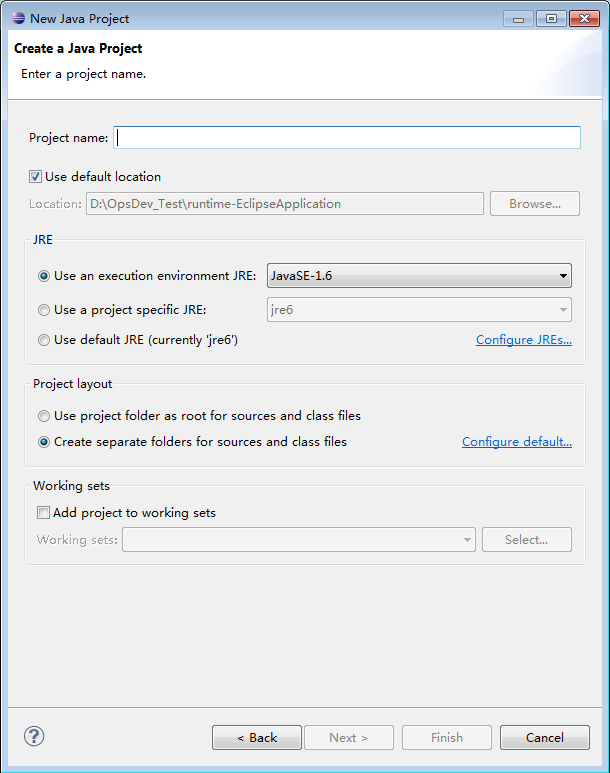
点击Other后，选择OpsDev java Project，点击Next进入下一步，如图3-2所示。

创建OpsDev-java Project示意图



设置OpsDev-Java Project项目，按需要进行以下设置后点击Next进入下一步，如图3-3所示。

OpsDev-java Project 工程示意图

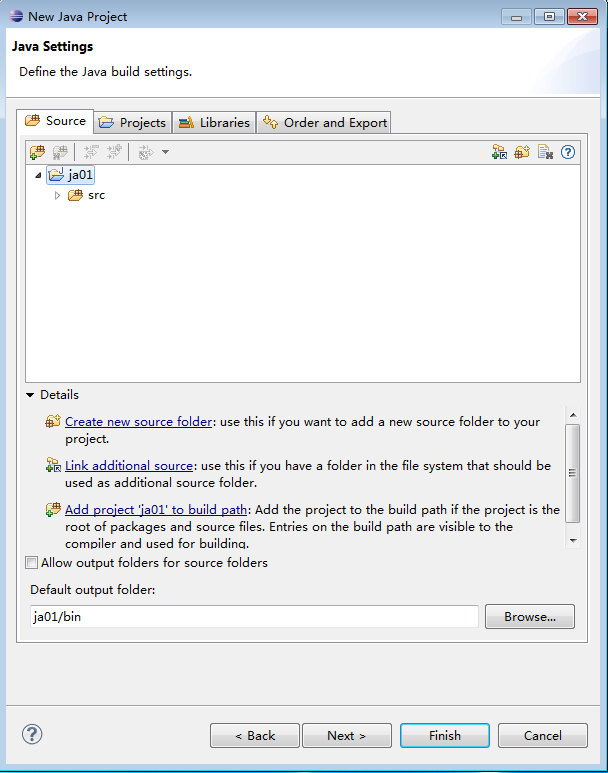


Project Name：工程名

Use Default location：默认工程路径。

JRE：选择JRE1.6

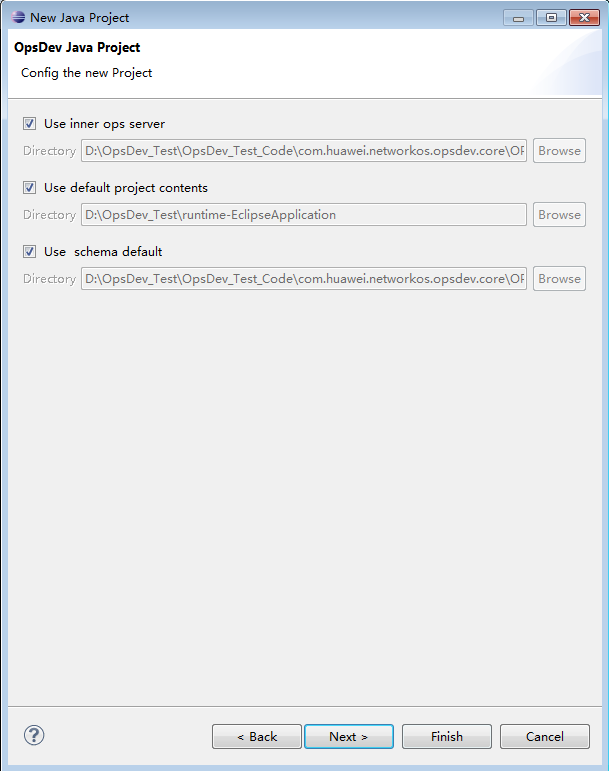
输入工程名，点击Next，进入如下图所示页面，定义java构建设置



图表 16

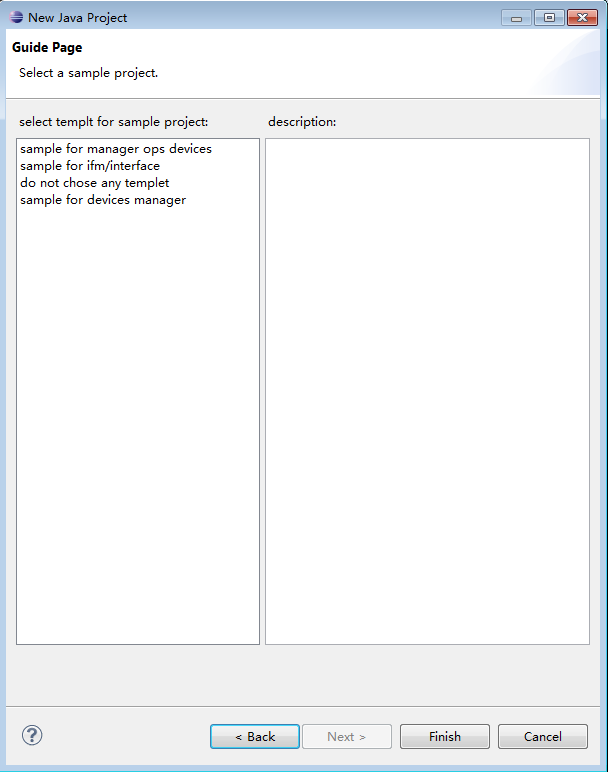
设置默认的OpsDev-java工程路径配置，如图3-4所示，设置完成后点击Next进入下一步。

默认的OpsDev-java工程路径配置示意图



选择OpsDev-java工程模板，如图3-5所示，完成后点Finish完成。

OpsDev-java工程模板示意图



在OpsDev-java视图中创建工程，具有相应的工程结构，如图3-6所示。

OpsDev-java视图中创建工程

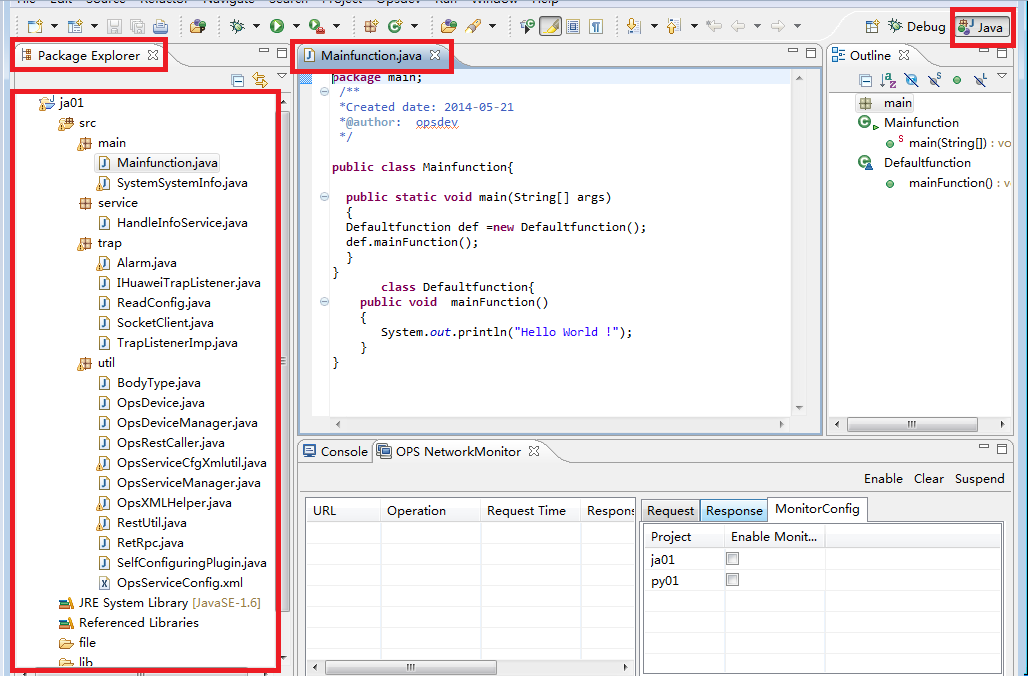


图3-6所示新建的工程。目录结构由工程主目录ja01，src、file和lib文件夹组成。

HandleInfoService.java: 配置OPS数据源相关信息。

Mainfunction.java：Mainfunction.java文件。

OpsRestCaller.java: 访问OPS客户端类。

OpsServiceConfig.java: 配置OPS连接信息。

OpsServiceManager.java: OPS访问相关服务类。

OpsServiceConfig.xml: OPS服务器配置信息。配置相关URL，端口号等。

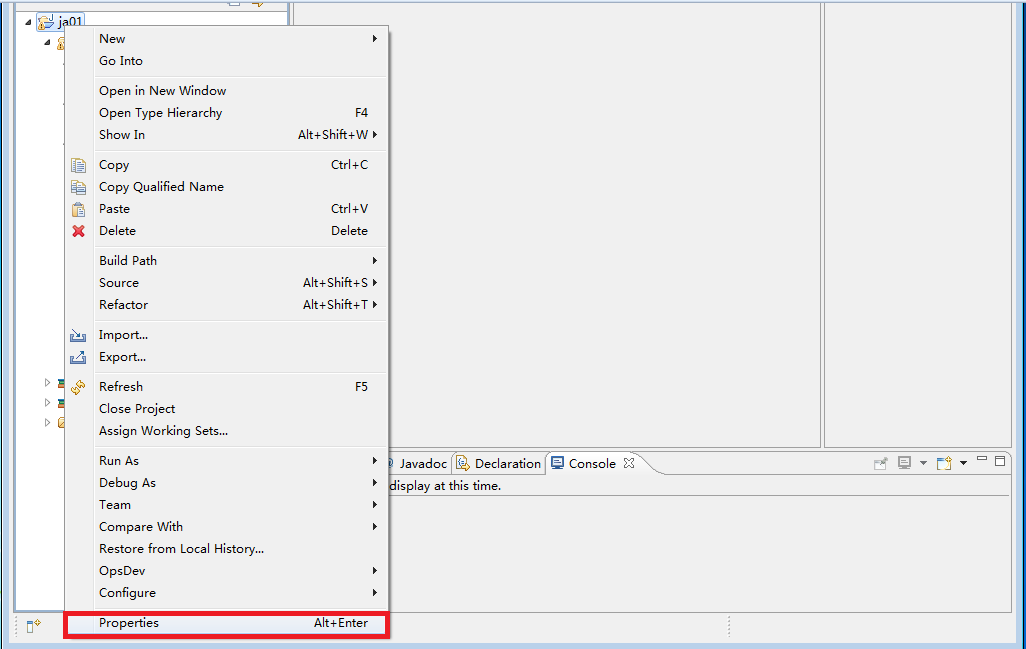
File：工程所需文件

lib：OpsDev-java编辑环境相关jar包。

## 修改OpsDev-Java工程属性

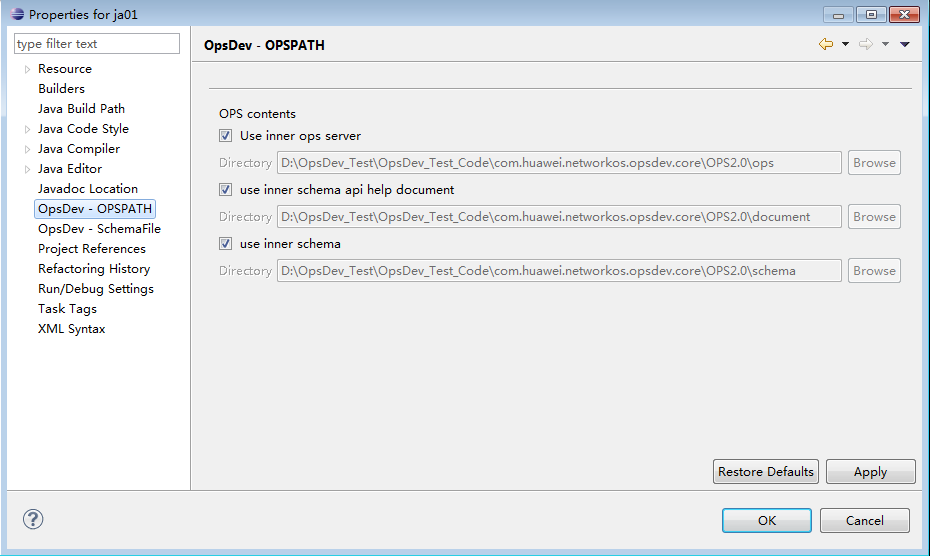
在新建的OpsDev-java工程上右键点击选择Properties，如图3-7所示。

进入OpsDev-Java工程属性界面示意图



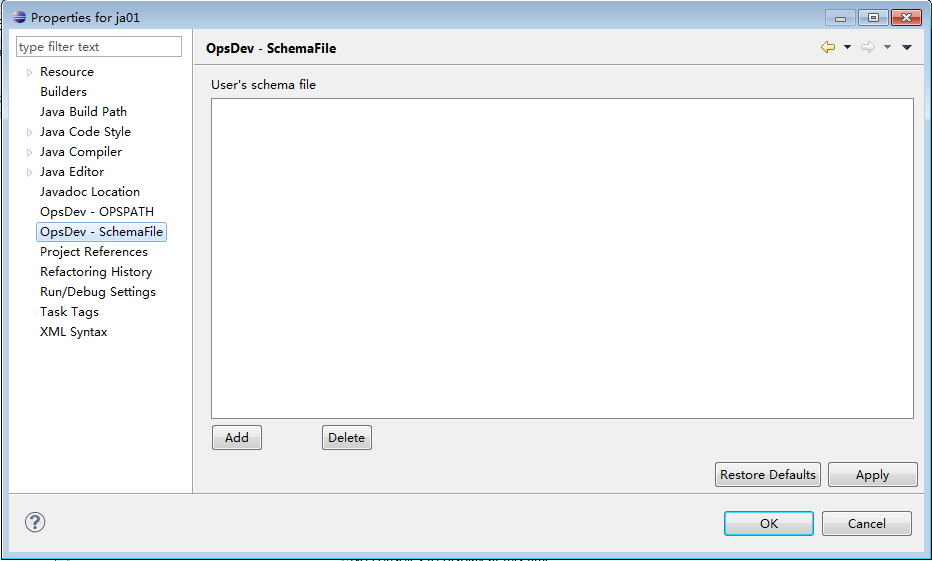
在OpsDev-OPSPATH中可以修改ops server、inner schema、user schema文件的路径，如图3-8所示。

OpsDev-OPSPATH修改属性示意图



可以在OpsDev-SchemaFile中添加、删除user schema文件，并且可以通过Restful Api接口调用，如图3-9所示，设置完成后点击OK完成。

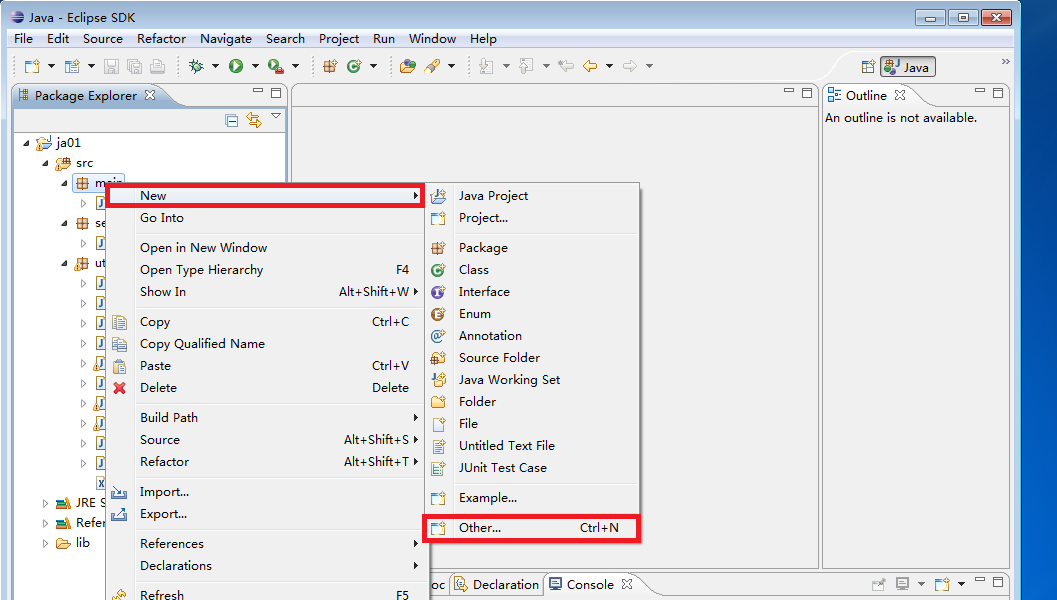
OpsDev-SchemaFile修改属性示意图



## 操作OpsDev-Java Restful Api

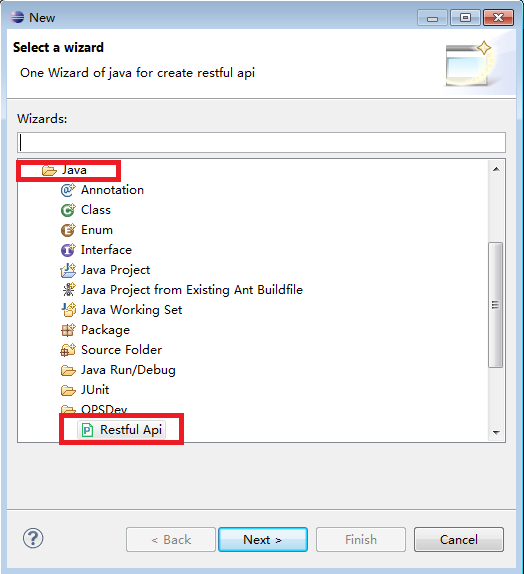
在工程的resource包中点击右键，选择New-->Other，如图3-10所示。

进入操作OpsDev-Java Restful Api界面示意图



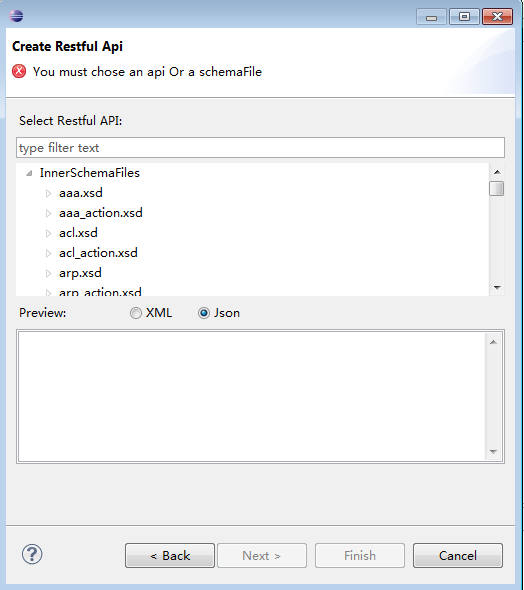
在Java中的OpsDev中选择Restful Api后点击Next进入下一步，如图3-11所示。

操作OpsDev-Java Restful Api界面视图



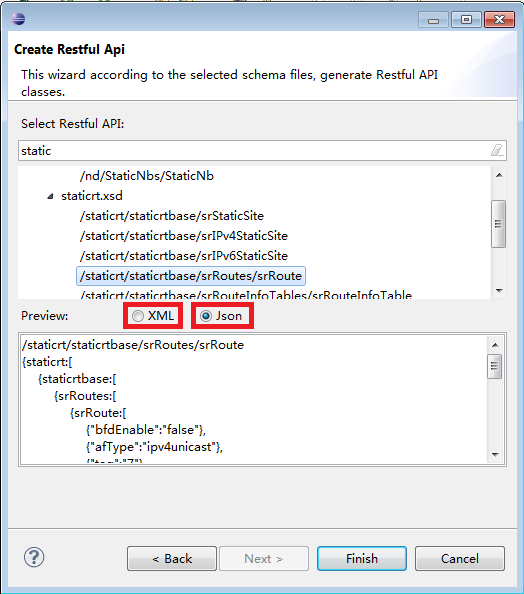
显示inner和user的schema文件树，如图3-12所示。

选择inner和user的schema文件树示意图



选择需要生成的报文格式，即可生成JSON或者XML格式的报文，如图3-13所示，设置完成后点击Finish完成。

选择OpsDev-Java Restful Api生成的报文格式界面示意图



在相应的包中生成.java文件，如图3-14所示。

生成文件示意图

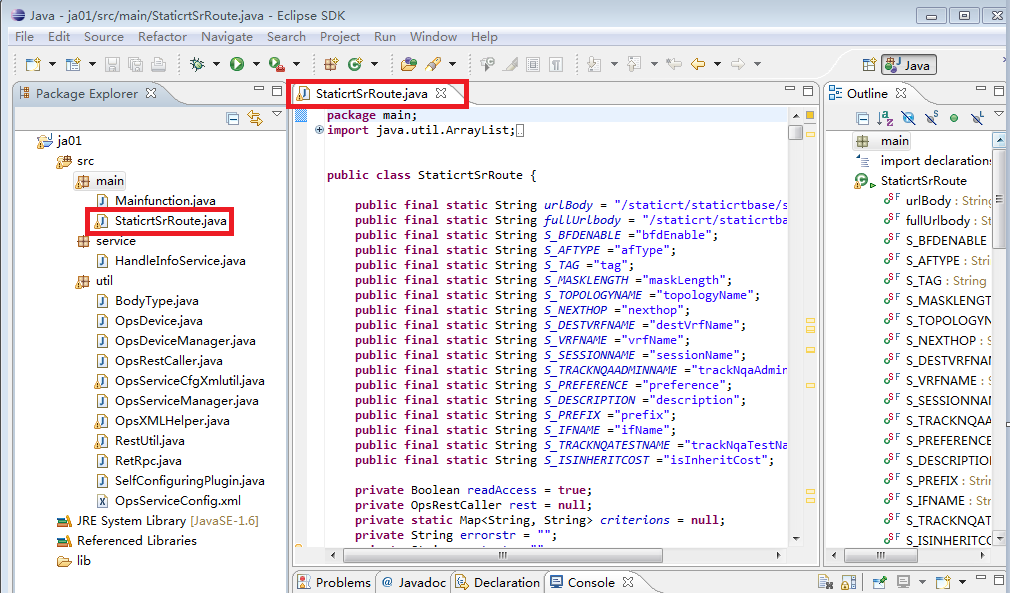


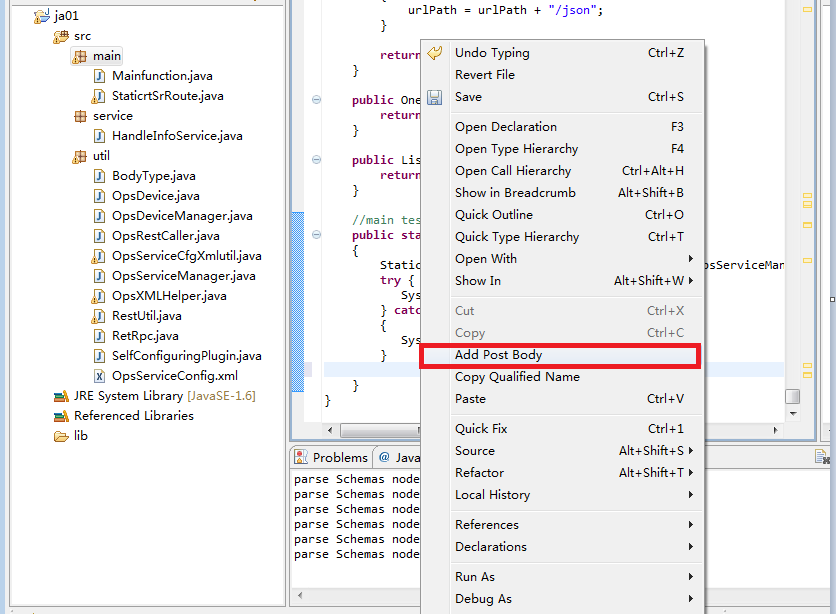
图3-14所示为新建的工程。目录结构由工程主目录ja01，src中由service,main和util三个包组成。

StaticrtSroute.java：工程自动生成的示例模板。

## 在代码中进行动态生成body（add post body）操作

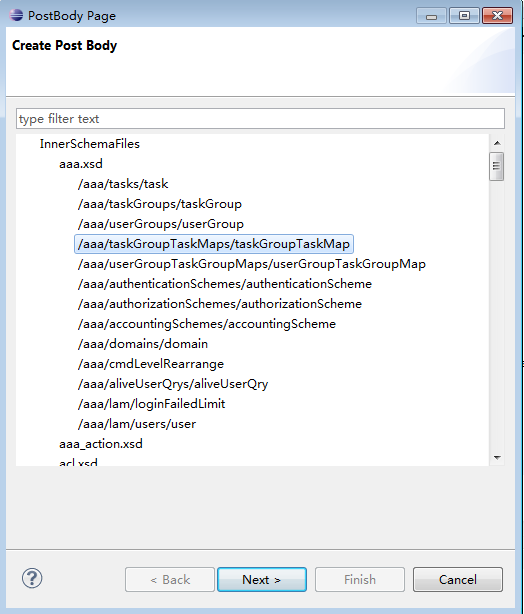
在代码的适当位置点击右键，选择Add Post Body菜单，如图3-15所示。

选择Add Post Body菜单界面示意图



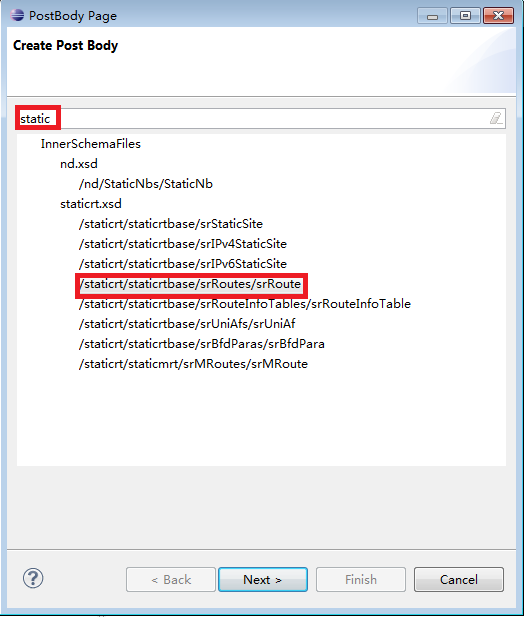
显示inner和user的schema文件树，如图3-16所示。

显示inner和user的schema文件树界面示意图



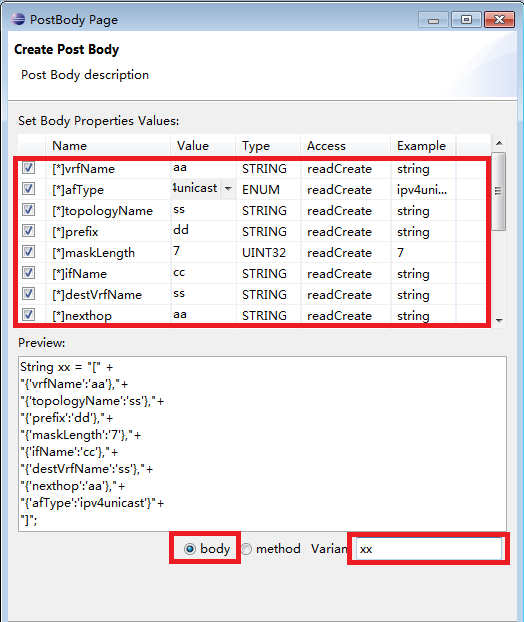
在文件树中选择schema文件，如图3-17所示。

选择Schema文件示意图



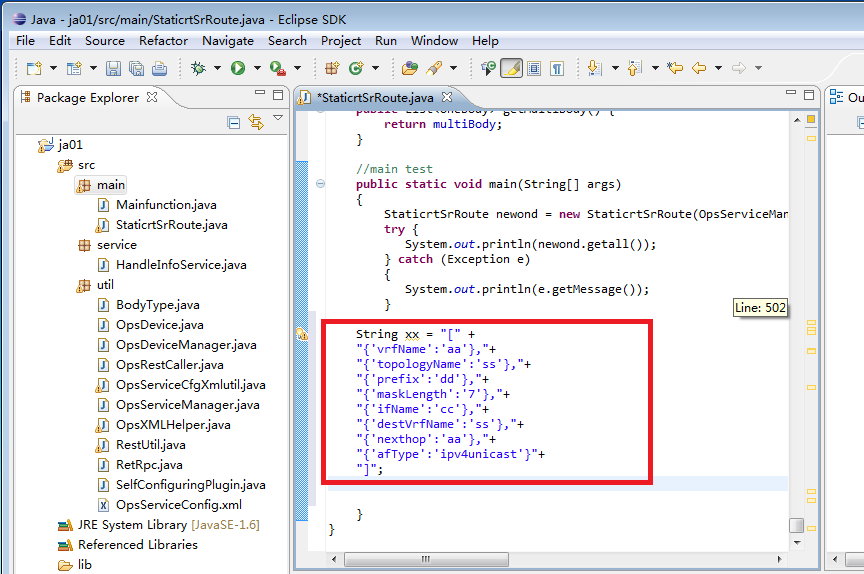
填写必填的参数\*号为必填参数，如图3-18所示。

填写参数示意图



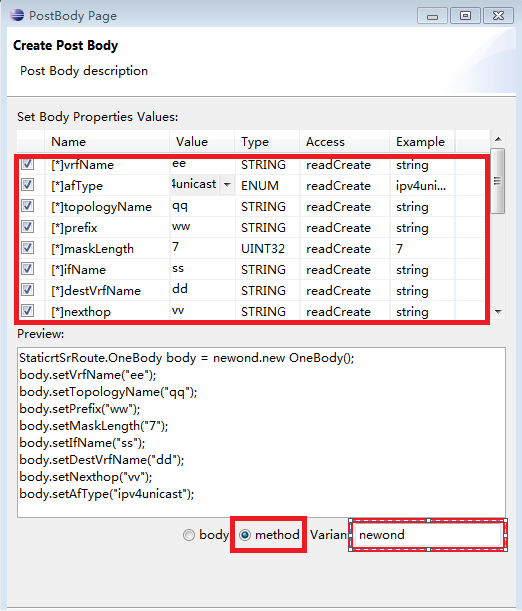
选择body则生成String格式的报文，String中显示的是json形式，如图3-19所示。

生成文件示意图

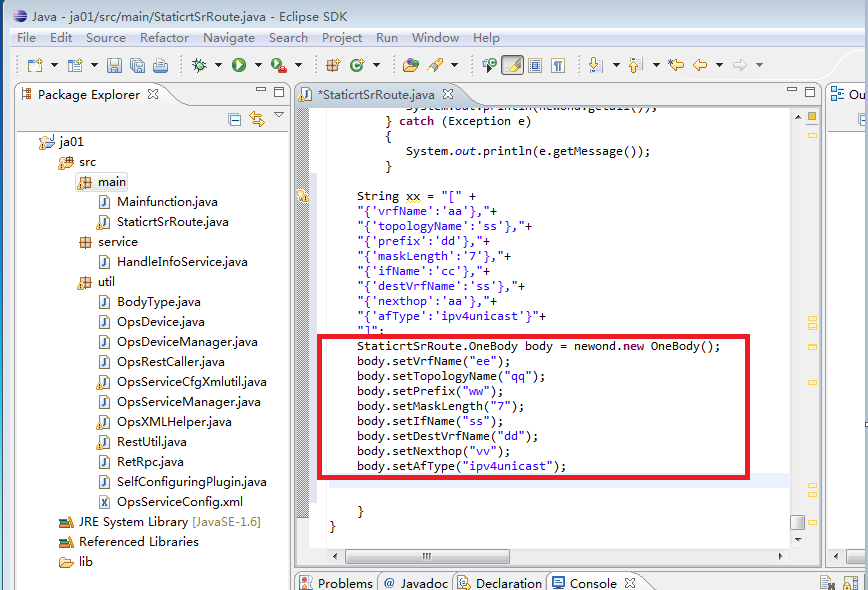


选择method则生成body格式的报文，如图3-20所示，生成文件如图3-21所示，红框部分为生成的Body体。

填写参数示意图



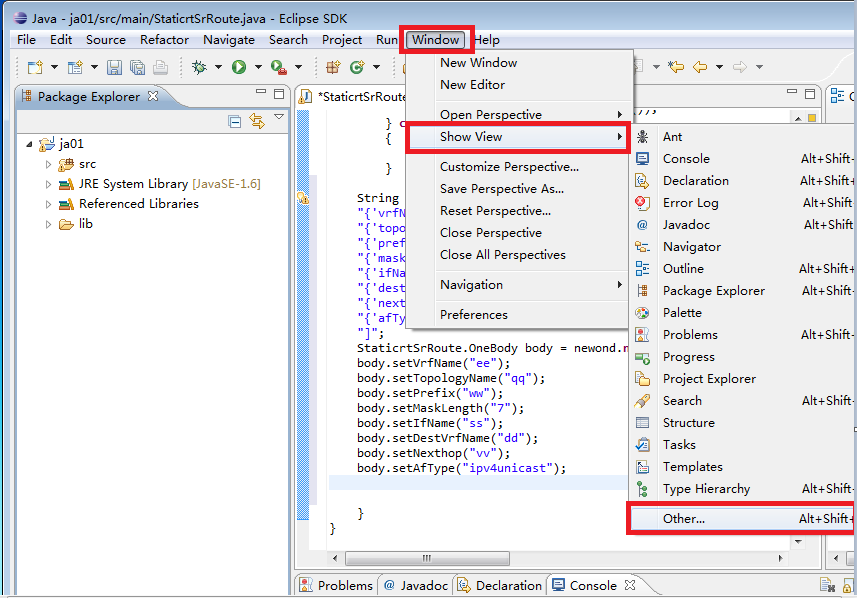
生成文件示意图



## 网络监视器

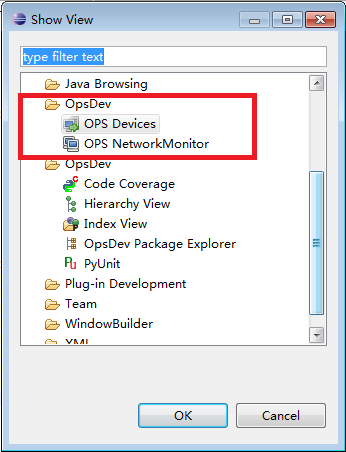
在Window中选择Show View，如图3-22所示，和OpsDev-Python类似，此处不重复描述。

选择Show View界面示意图

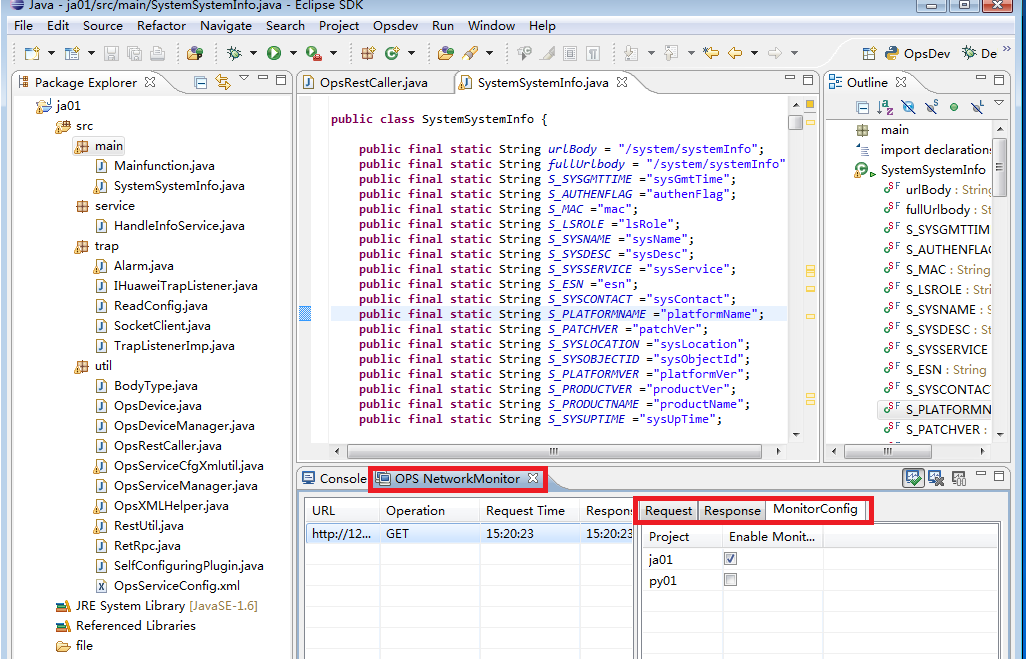


在OpsDev中打开OPS NetworkMonitor，如图3-23所示。

打开OPS NetworkMonitor界面示意图

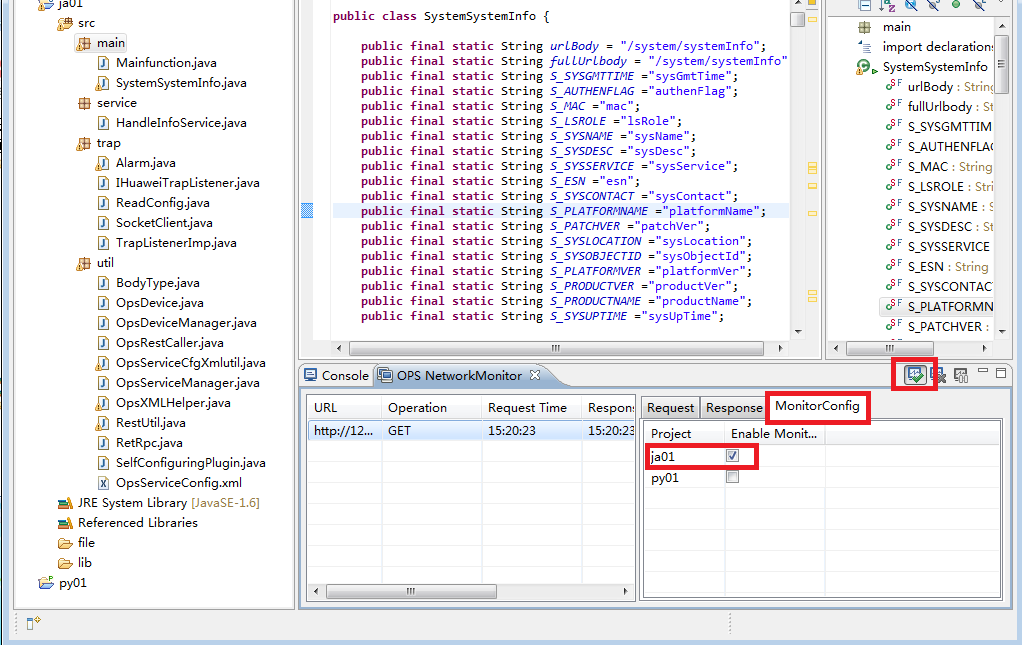


在上图的OpsDev中选择OPS NetworkMonitor，打开监视器界面，如下图所示



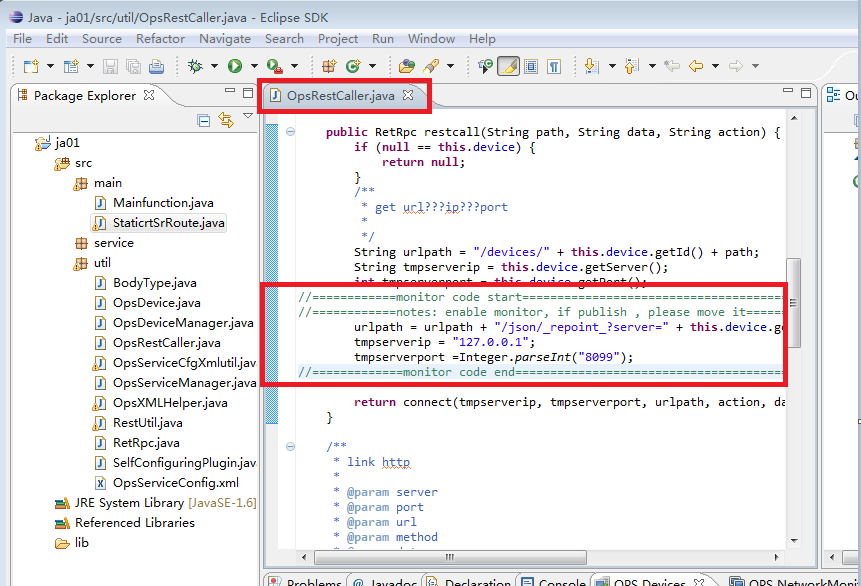
打开网络监视的开关，在MonitorConfig中选中对应工程的Enable Monitor勾选框，用来选择该工程是否需要进行监听，打钩则进行监听，不打钩则取消监听，如图3-24所示。

打开网络监视的开关示意图



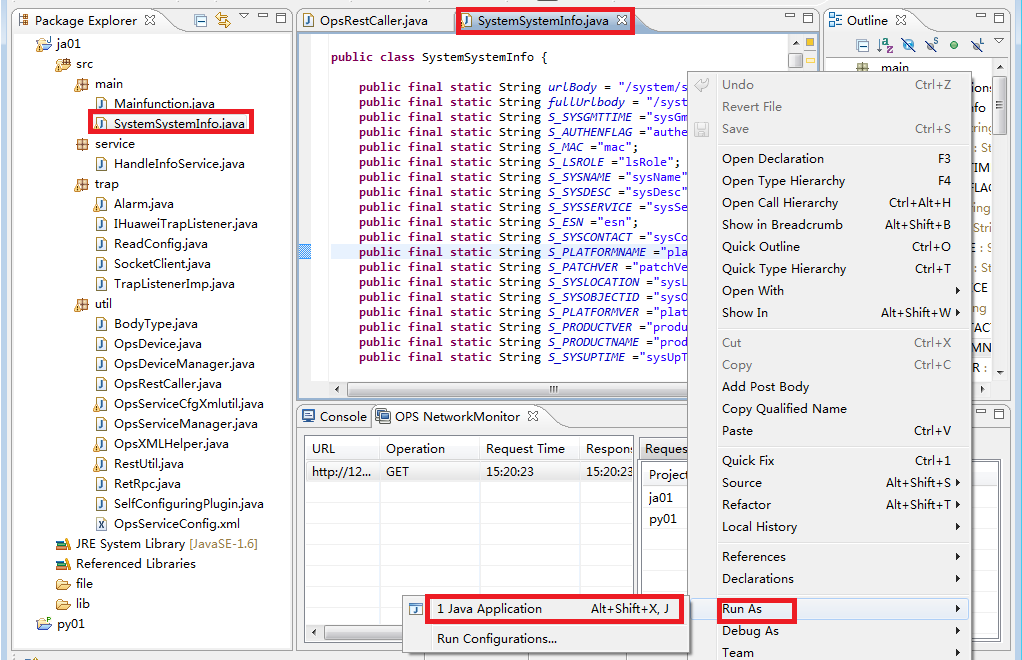
打开监视器开关，并且勾选上该工程的Enable Monitor勾选框，则会在OpsRestCaller.java文件中会打开服务器的监听端口，如图3-25所示。

打开监视器界面示意图



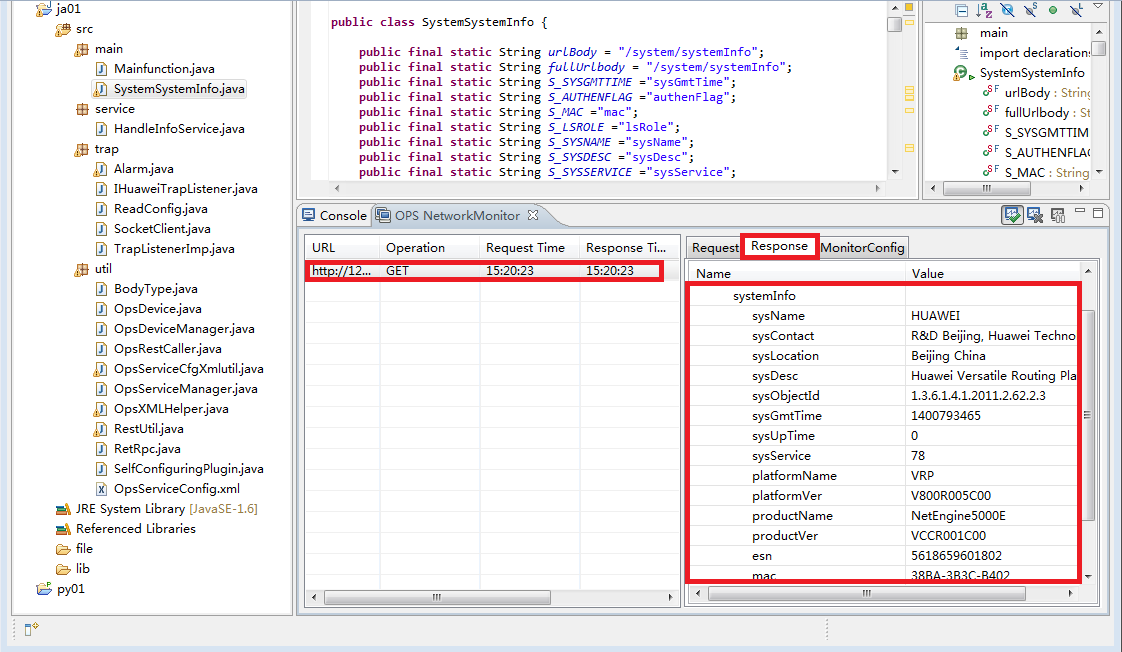
在SystemSystemInfo.java文件中右击，选择Run As Java Application运行，如图3-26所示。

选择Run As Java Application运行界面示意图



可以在监视器视图中的Response中监视Restful Api文件运行情况，显示了基本的请求统计数据，获取相关URL、发送的请求和响应的次数、请求方式等信息，如图3-27所示。

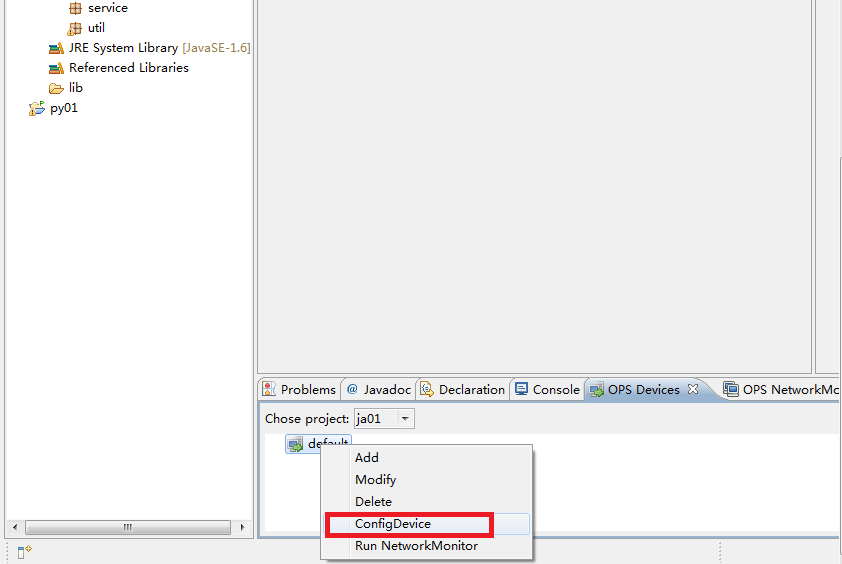
监视Restful Api文件运行示意图



## 数据源管理

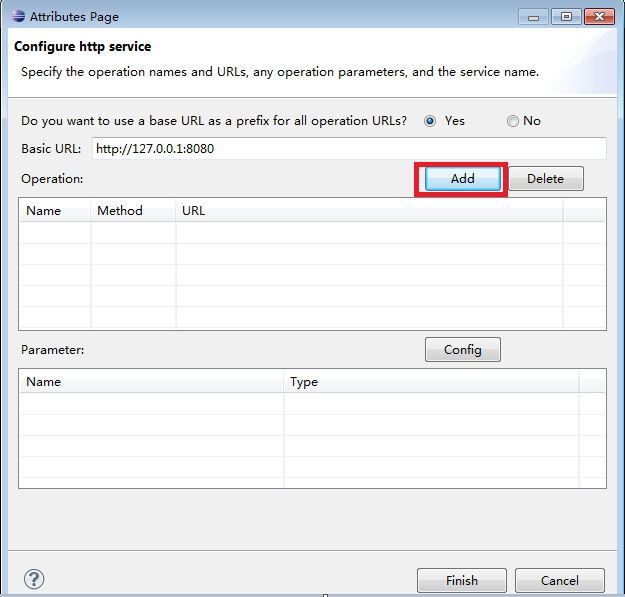
在OPS Devices中选择工程，在工程上点击右键，选择ConfigDevice，如图3-28所示。

打开数据源界面示意图



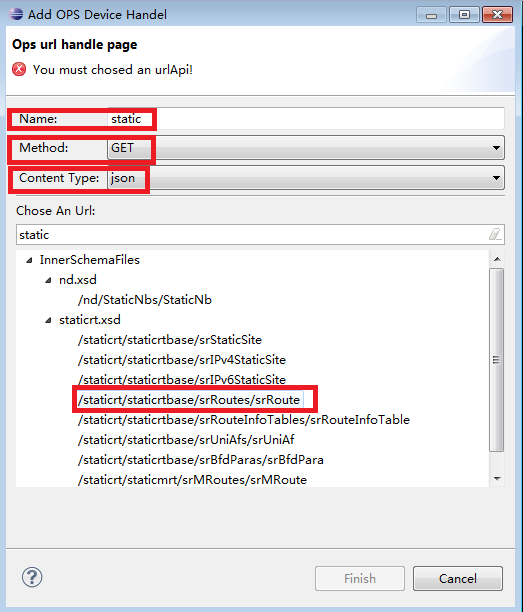
点击Add按钮，如图3-29所示。

数据源管理界面示意图1



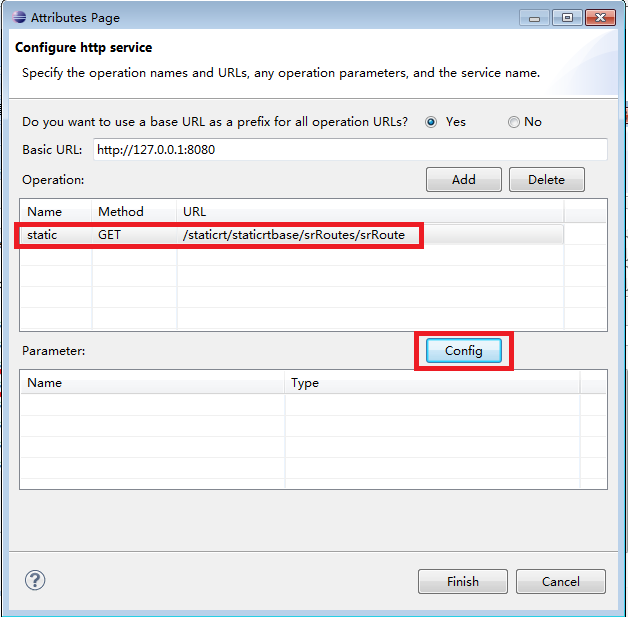
填写处理方法的名称、请求方法以及返回的信息格式，并且选择一个schema文件，如图3-30所示。

数据源管理界面示意图2



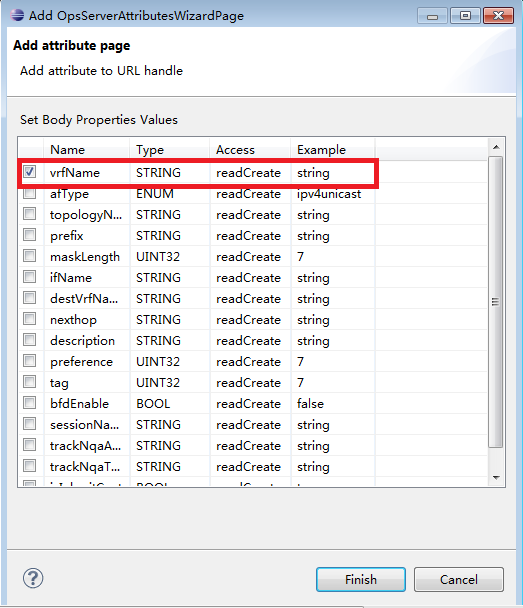
选中添加的方法，点击Config按钮，对方法进行配置，如图3-31所示。

数据源管理界面示意图3



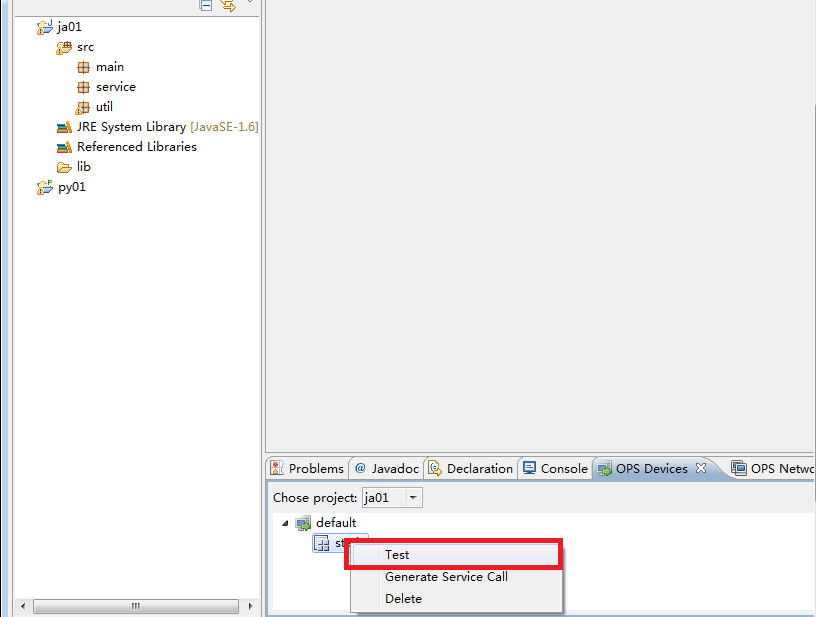
选择body体的属性参数，如图3-32所示。

选择body体参数示意图



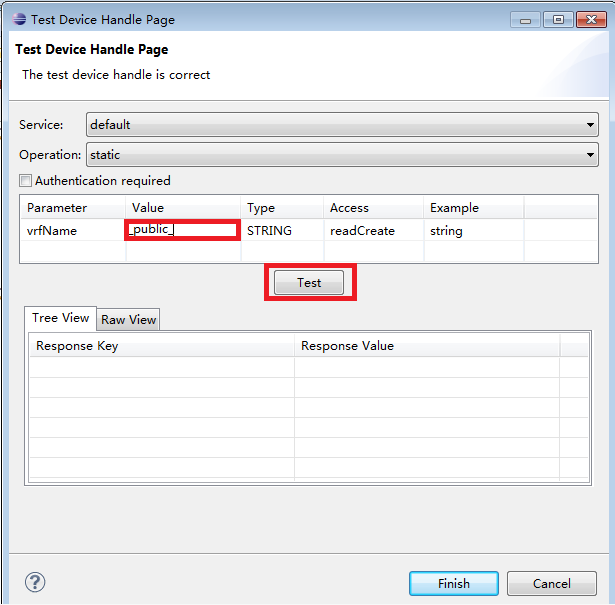
在添加好的方法中点击邮件，选择Test，如图3-33所示。

对添加的方法选择Test按钮示意图1



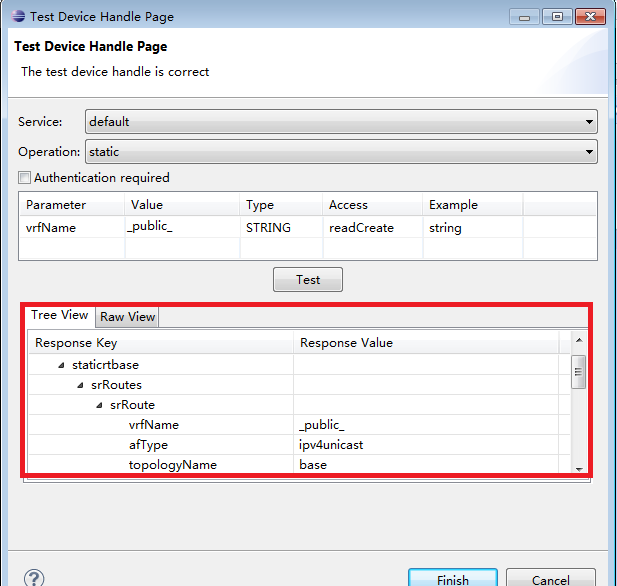
填写参数值，点击Test按钮。如图3-34所示。

对添加的方法选择Test按钮示意图2



在树中返回获取到的信息，如图3-35所示。

返回获取信息示意图



说明：

OpsDev-java工程中对数据源的增加、删除，修改的方法和OpsDev-python中的方法一致，在此不重复说明。