

潤沁實業

潤沁網路大學

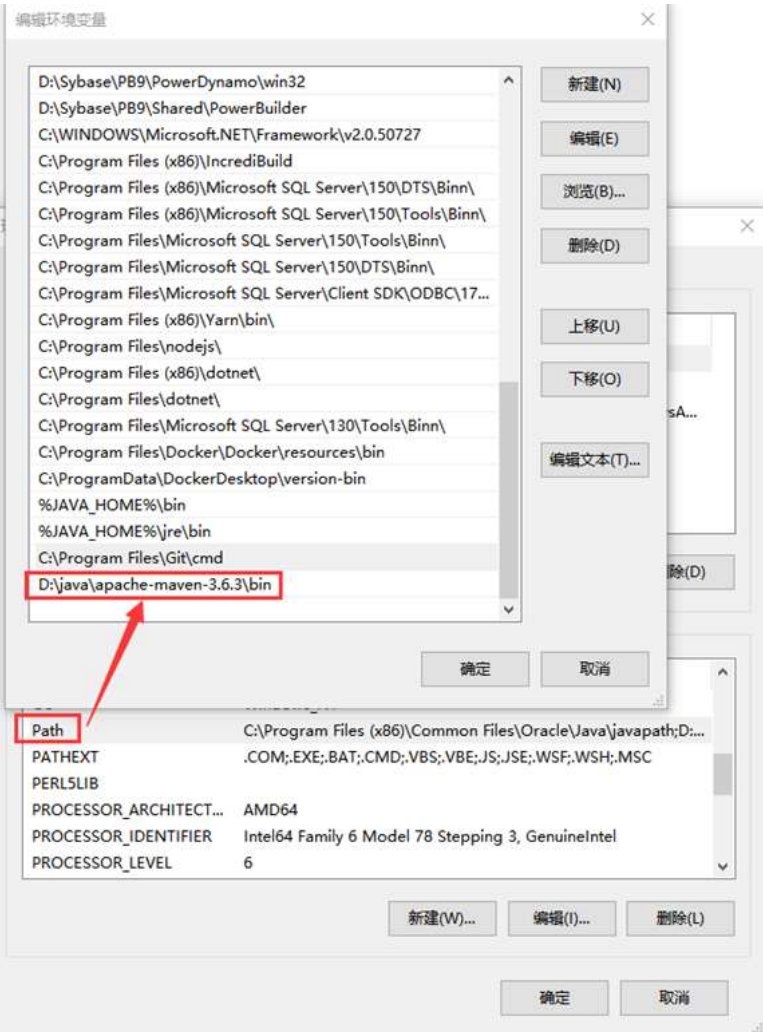
博客园 首页 新随笔 联系 订阅 管理

第八課-Channel Study For Caller Custom JAR Lib

1.下载 apache-maven

推荐3.6.3版本: <https://maven.apache.org/download.cgi>

设置系统环境变量: PATH



装好后在命令行验证:mvn -v

```
管理员: cmd
Microsoft Windows [版本 10.0.19042.746]
(c) 2020 Microsoft Corporation. 保留所有权利。

C:\WINDOWS\system32>mvn -v
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8 -Duser.language=en -Duser.country=US
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: D:\java\apache-maven-3.6.3\bin\..
Java version: 1.8.0_201, vendor: Oracle Corporation, runtime: D:\java\jdk8\jre
Default locale: en_US, platform encoding: UTF-8
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Windows\System32>
```

Maven中央仓库repo.maven.apache.org下载慢问题

编辑D:\java\apache-maven-3.6.3\conf目录下setting.xml文件, 加入阿里云的源:

公告

昵称: 潤沁網路大學
园龄: 1年10个月
粉丝: 2
关注: 0

2021年3月				
日	一	二	三	四
28	1	2	3	4
7	8	9	10	11
14	15	16	17	18
21	22	23	24	25
28	29	30	31	
4	5	6	7	8

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Mirth(18)
HL7(7)
DataBase(2)

随笔分类

DataBase(2)
HL7(7)
Mirth(15)

随笔档案

2021年2月(2)
2021年1月(20)

文章分类

Mirth(3)

最新评论

1. Re:第八課-Channel Study
Custom JAR Lib
受益匪浅!!!

阅读排行榜

```
<mirrors>
<!-- 阿里云仓库 -->
<mirror>
<id>alimaven</id>
<mirrorOf>central</mirrorOf>
<name>aliyun maven</name>
<url>https://maven.aliyun.com/nexus/content/repositories/central/</url>
</mirror>

<mirror>
<id>nexus-aliyun</id>
<mirrorOf>central</mirrorOf>
<name>Nexus aliyun</name>
<url>http://maven.aliyun.com/nexus/content/groups/public</url>
</mirror>
</mirrors>
```

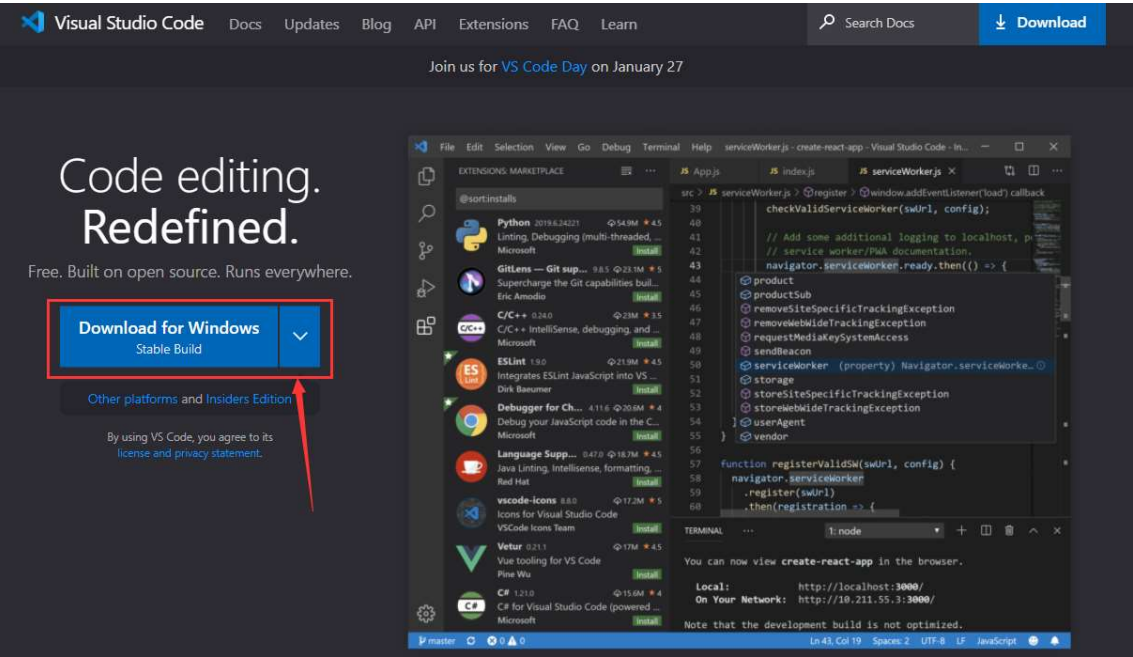
1. HL7传输协议(161)
2. 第壹課-Install: Mirth Coni 安装步骤(99)
3. 开篇:Mirth Connect系统集5)
4. HL7标准的版本(75)
5. 第三課: 信道学习Source C Destinations File Writer(60)

评论排行榜

1. 第八課-Channel Study For R Lib(1)

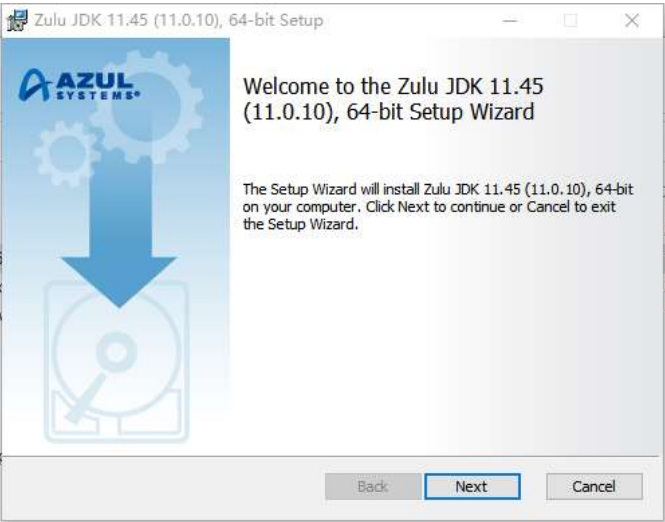
2.安装微软开源的Visual Studio Code IDE集成开发环境

下载URL地址: <https://code.visualstudio.com/>



3.安装微软官方支持的Azul OpenJDK11

下载URL地址: <https://www.azul.com/downloads/azure-only/zulu/?version=java-11-lts&os=windows&architecture=x86-64-bit&package=jdk>

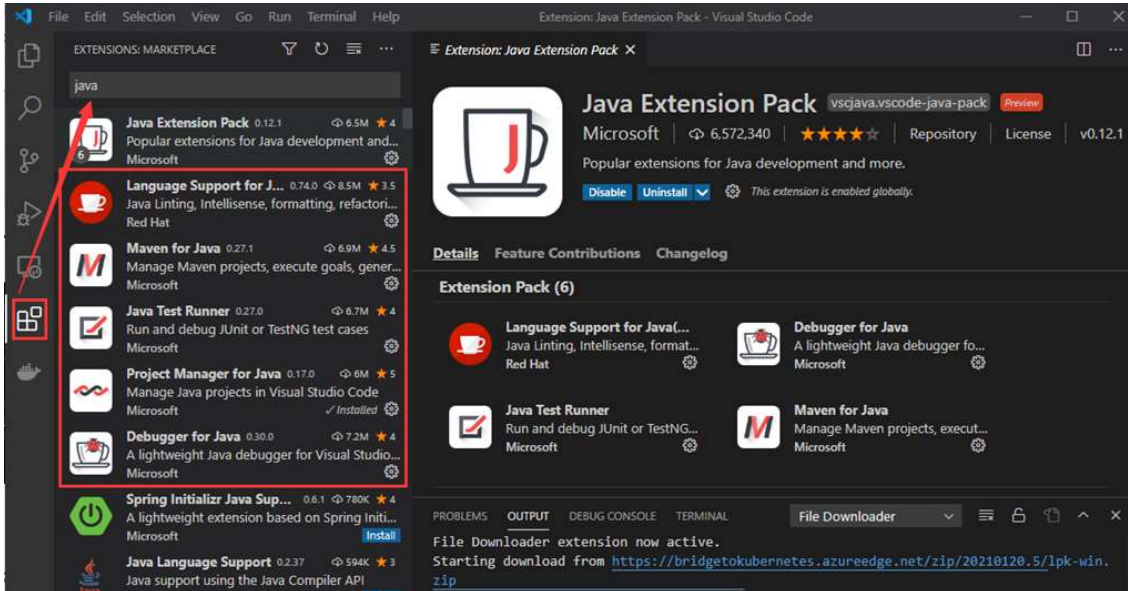


4.安装vs code的Java开发相关插件

需要安装的有如下:

- Language Support for Java(TM) by Red Hat: 通过 Eclipse™ JDT Language Server 提供 Java 语言支持
- Debugger for Java: 基于 Java Debug Server 的轻量级调试工具。
- Java Test Runner: 执行和调试 Java 测试用例。

- Maven for Java: Maven 插件。



5.创建maven项目

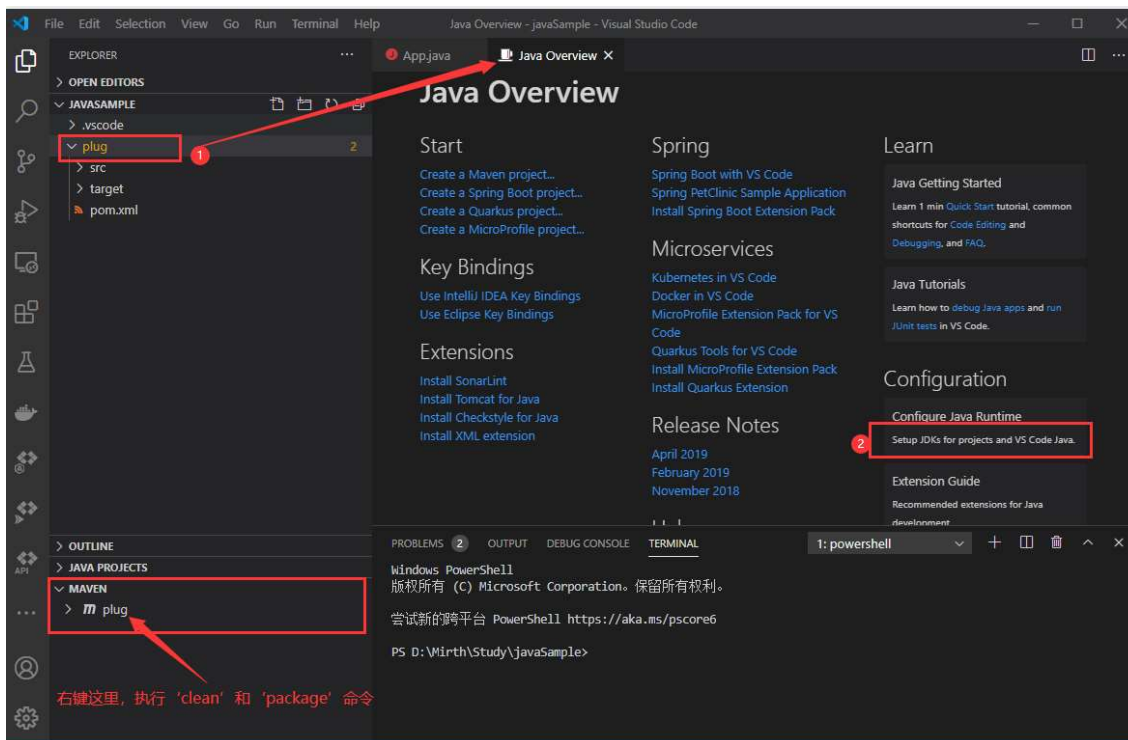
- 按下 Ctrl+Shift+P 打开命令面板, 输入java,依次选择Create Java Project、Maven;接着选择基于 maven-archetype-quickstart 创建
- 在弹出的对话框中选择一个生成项目的目录
- 选择完成后, 会开始项目的生成, 期间需要在终端中填入必要的信息。groupId组织名(java包名)、artifactId项目名、version版本号、package打包后jar的名称

```
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.
4
[INFO] -----
[INFO] Parameter: groupId, Value: eohagedu
[INFO] Parameter: artifactId, Value: plug
[INFO] Parameter: version, Value: 1.0
[INFO] Parameter: package, Value: eohagedu
[INFO] Parameter: packageInPathFormat, Value: eohagedu
[INFO] Parameter: package, Value: eohagedu
[INFO] Parameter: version, Value: 1.0
[INFO] Parameter: groupId, Value: eohagedu
[INFO] Parameter: artifactId, Value: plug
[INFO] Project created from Archetype in dir: D:\Wirth\Study\javaSample\plug
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 06:13 min
[INFO] Finished at: 2021-01-25T22:30:41+08:00
[INFO] -----
PS D:\Wirth\Study\javaSample> []
```

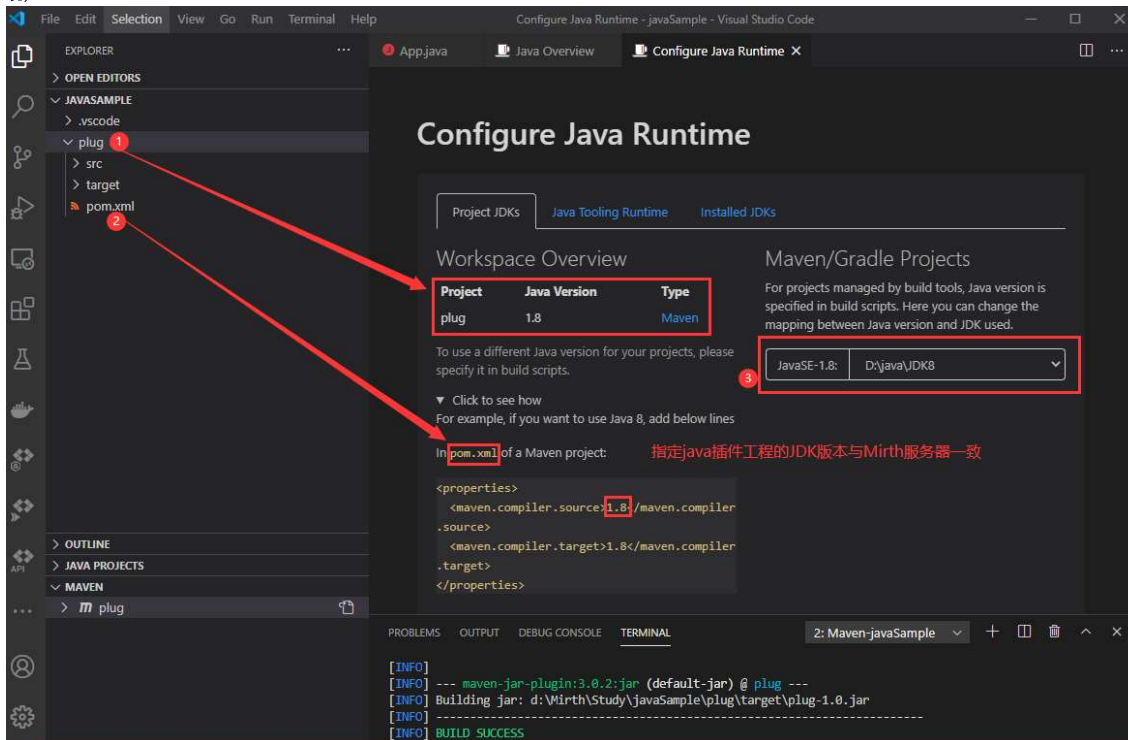
如下系列图所示: 因为Visual Studio Code需要JDK11; 但是Mirth Connect只能运行于Oracle JDK8环境下, 故在VS Code选中项目名plug, 点击'Setup JDKs for Project and VS Code java'设置不同的JDK环境:

安装微软官方支持的Azul OpenJDK11是因为Visual Studio Code调试java代码需要这个运行环境;

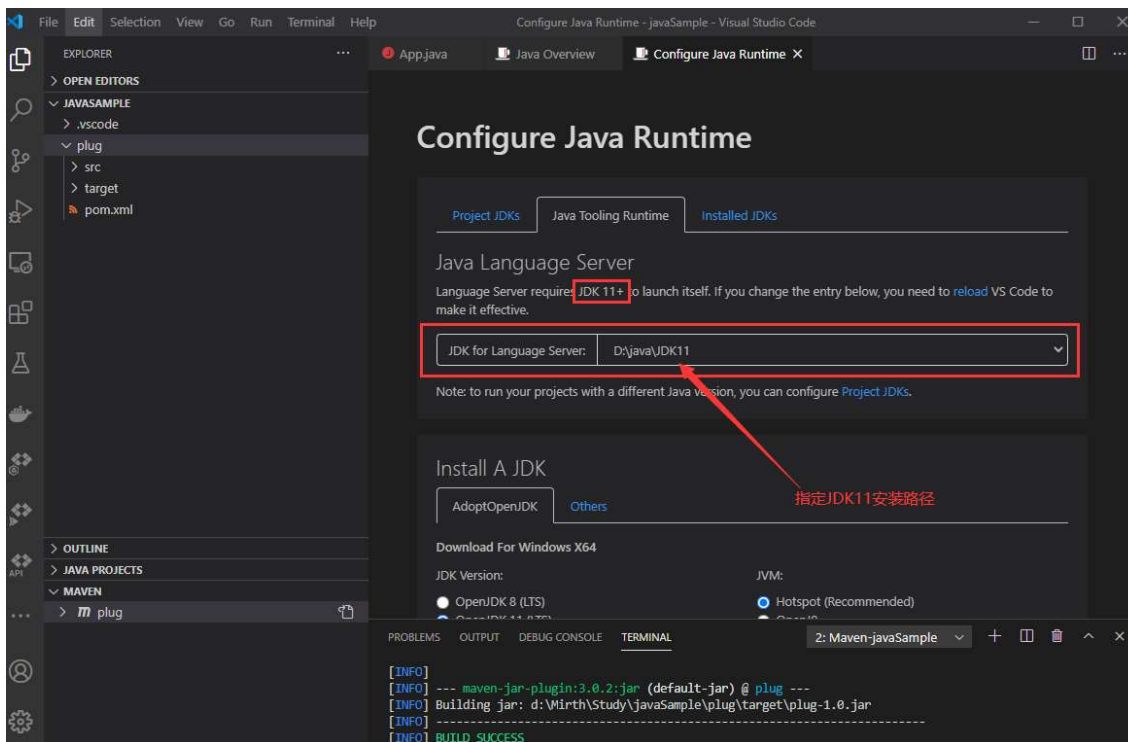
安装Oracle官方JDK8是因为Mirth Connect服务运行在JDK8下, 自定义扩展jar包的插件开发必须和MC Server使用的SDK一致。



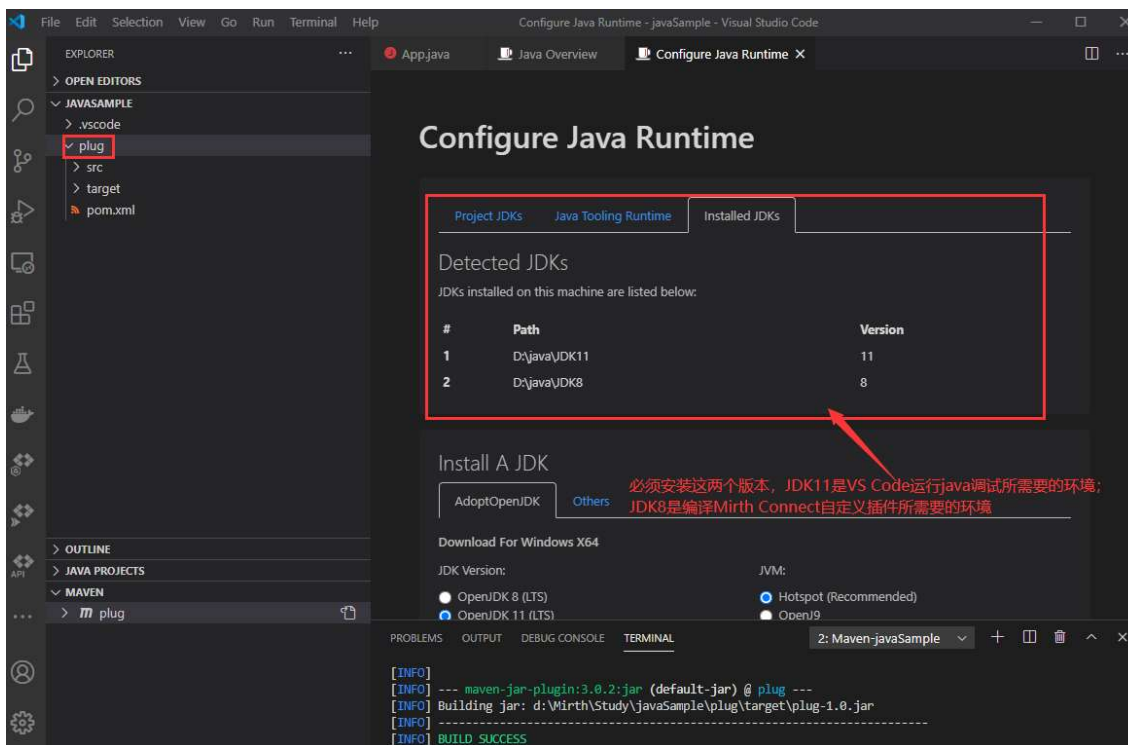
(如上图所示: 第一步, 选中自定义扩展插件plug工程;第二步, Java Overview页签中点击'Setup JDKs for Project and VS Code java'设置不同的JDK环境)



(如上图所示: 第一步, 选中自定义插件plug工程;第二步, 修改pom.xml文件使用1.8版本JDK;第三步, 设置plug插件工程project JDKs为JAVA-SE1.8 的JDK环境)

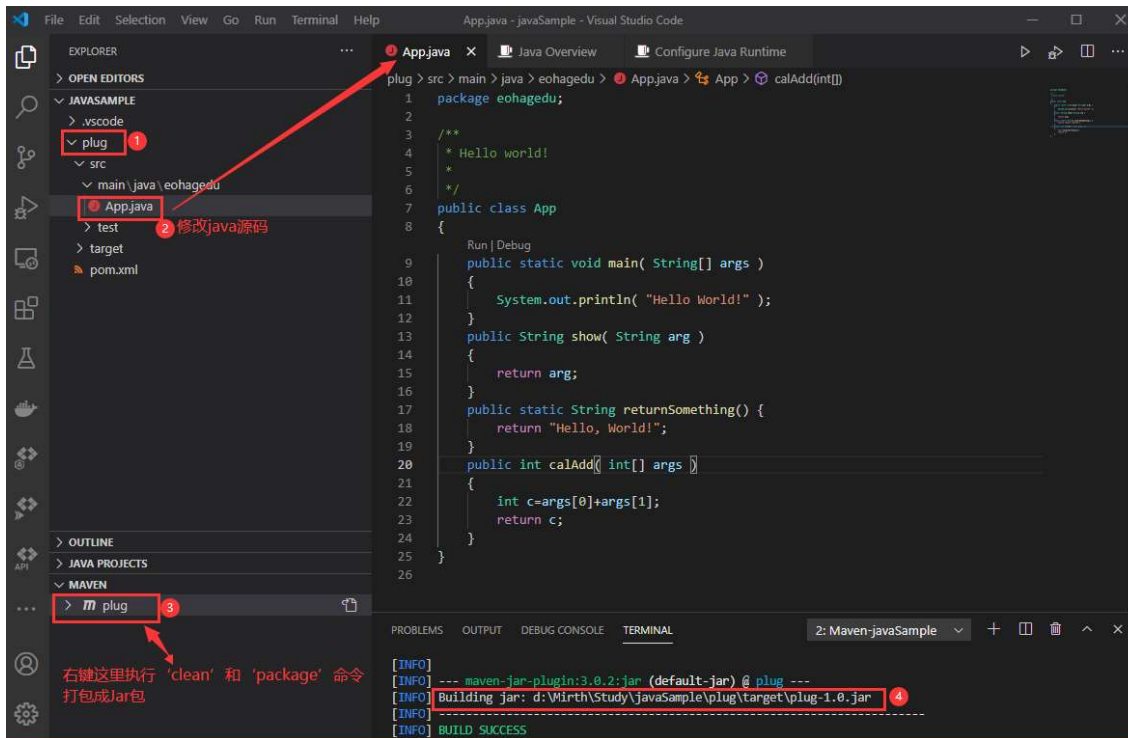


(如上图所示：第一步，选中自定义插件plug工程；第二步，设置VS Code IDE开发工具Java Tooling Runtime设置为JDK11环境)



(如上图所示：第一步，选中自定义插件plug工程；第二步，确认已经安装了JDK8和JDK11的环境)

6.修改plug插件工程app类源码并调试



创建完成后，打开目录即可

Ctrl+Shift+D可打开调试面板，进行调试；

本示例自定义Jar包插件源码如下：

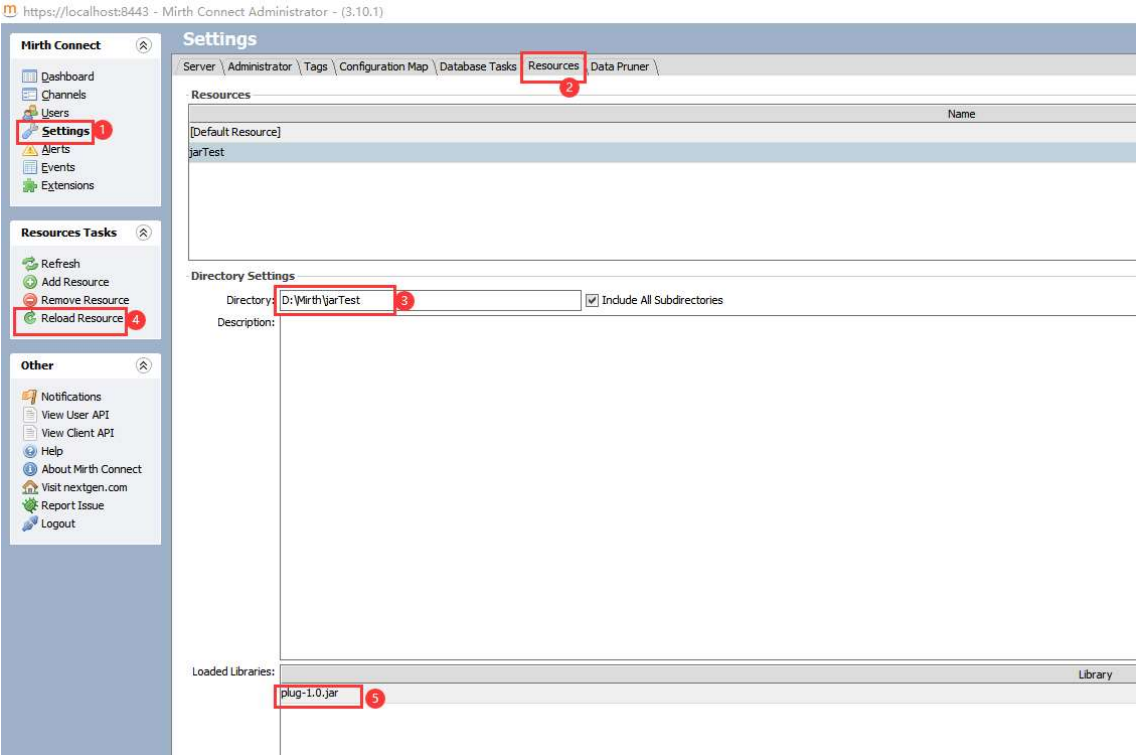
```

1  package eohagedu;
2
3  /**
4   * Hello world!
5   *
6   */
7  public class App
8  {
9      public static void main( String[] args )
10     {
11         System.out.println( "Hello World!" );
12     }
13     public String show( String arg )
14     {
15         return arg;
16     }
17     public static String returnSomething() {
18         return "Hello, World!";
19     }
20     public int calAdd( int[] args )
21     {
22         int c=args[0]+args[1];
23         return c;
24     }
25 }

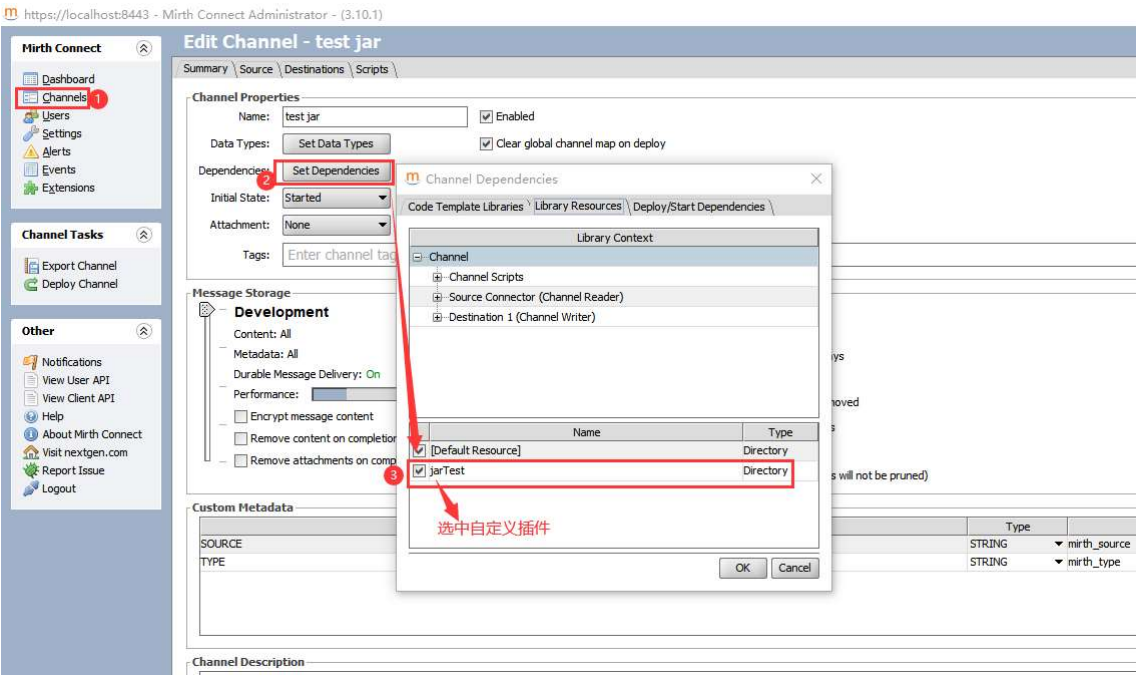
```

7.打开Mirth Connect 3.10.1驾驶舱，加载自定义扩展Jar包目录

选择左侧'Settings'菜单，点击'Resources'页签，指定第6步骤编译出的自定义Jar包插件目录为D:\Mirth\jarTest；然后点击'Reload Resource'菜单完成自定义插件导入

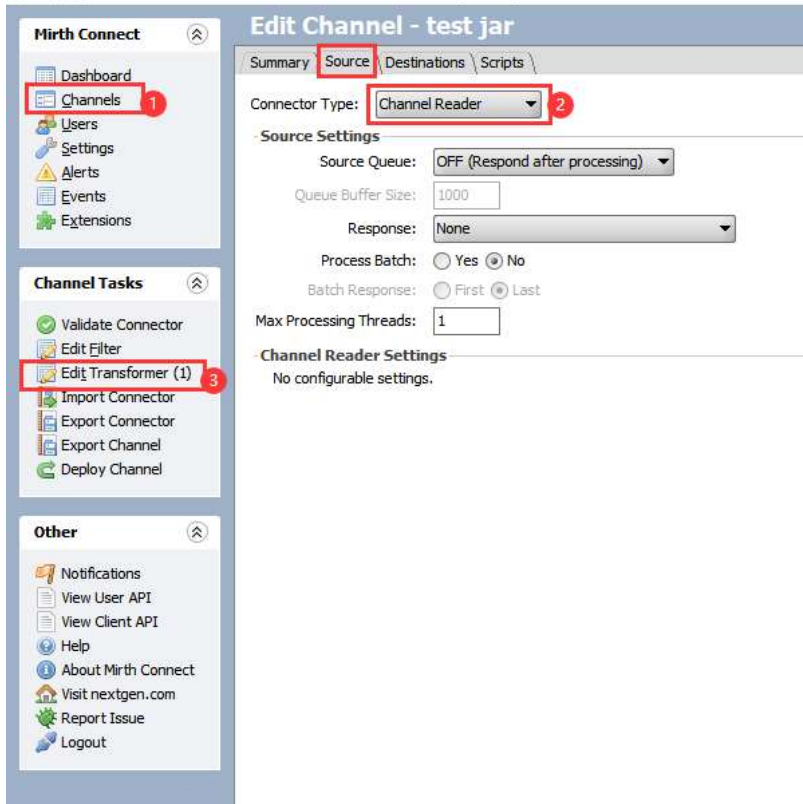


8.新建测试Mirth Connect信道

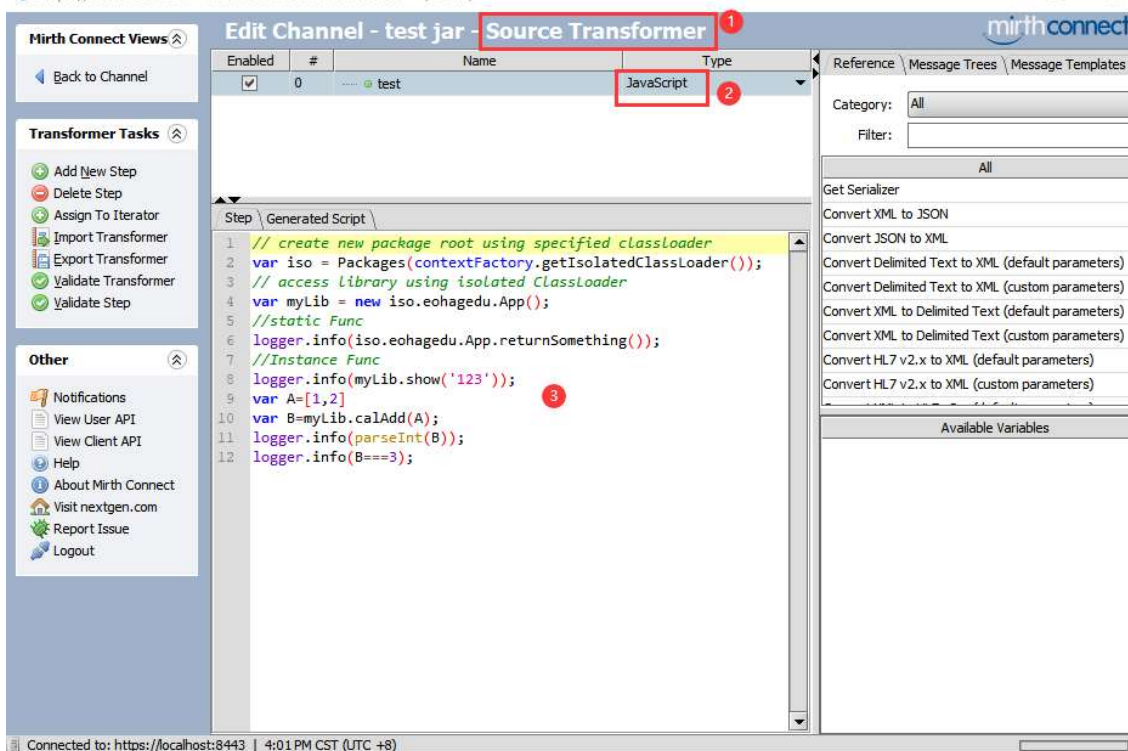


9.测试自定义扩展Jar包调用，在源端的转换中添加调用自定义Jar包的js脚本

https://localhost:8443 - Mirth Connect Administrator - (3.10.1)



https://localhost:8443 - Mirth Connect Administrator - (3.10.1)



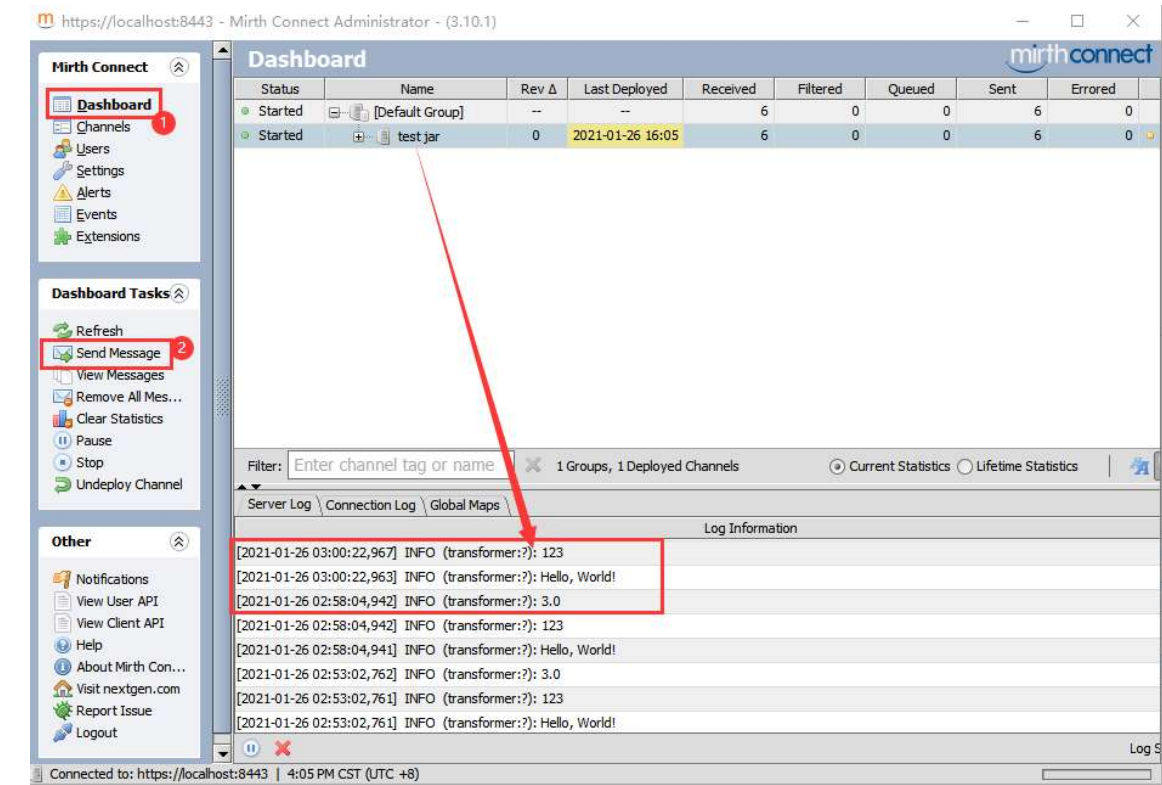
(如上图所示: 分别调用静态方法returnSomething、字符串实例方法show、数组参数实例方法calAdd)

```

1 // create new package root using specified classloader
2 var iso = Packages(contextFactory.getIsolatedClassLoader());
3 // access library using isolated ClassLoader
4 var myLib = new iso.eohagedu.App();
5 //static Func Caller
6 logger.info(iso.eohagedu.App.returnSomething());
7 //Instance string Func Caller
8 logger.info(myLib.show('123'));
9 //Instance Array Func Caller
10 var A=[1,2]
11 var B=myLib.calAdd(A);
12 logger.info(parseInt(B));
13 logger.info(B===3);

```


10.发布信道测试



大功告成！！

- 本课程总结：
- 1.Mirth Connect 信道调用自定义Jar包：可以指定任意目录作为插件目录，不需要重启MC服务即可直接在信道中使用。
 - 2.信道中指定依赖时，选中所加载的自定义Jar包目录，此信道即可使用此扩展自定义jar包
 - 3.通过js脚本调用java自定义jar包，扩展了Mirth Connect系统集成与数据交换平台的功能
 - 4.强大如斯的Mirth Connect平台，您有什么理由不好好研究呢？

欢迎大家持续关注潤沁網路大學本系列Mirth Connect课程的教学

本节课所参考网志索引：

<https://www.mirthcorp.com/community/wiki/display/mirth/How+to+create+and+invoke+custom+Java+code+in+Mirth+Connect>

<https://github.com/nextgenhealthcare/connect/issues/4324>

<https://forums.mirthproject.io/forum/mirth-connect/support/175423-javascript-error-using-apache-pdfbox-and-fontbox>

<https://forums.mirthproject.io/forum/mirth-connect/support/18965-channel-classloader-isolation>

潤沁網路大學

分类: Mirth

标签: Mirth

好文要顶

关注我

收藏该文

潤沁網路大學

关注 - 0

粉丝 - 2

- « 上一篇: [HL7传输协议](#)
- » 下一篇: [第九課-Channel Study For Caller Client Api](#)

posted @ 2021-01-26 16:12 潤沁網路大學 阅读(45) 评论(1) 编辑 收藏

评论列表

- #1楼 2021-01-26 16:40 少年郎1

回复 引用 删除
- 受益匪浅！！

支持(0) 反对(0)

发表评论

刷新评论 刷新页面 返回顶部

编辑预览

B🔗🔼🔽🖼

支持 Markdown

✍

自动补全

提交评论退出

[Ctrl+Enter]快捷键提交

- 【推荐】大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载!
- 【推荐】亚马逊云科技在线研讨会：借助神经网络实现实时欺诈检测
- 【推荐】华为开发者联盟--邀友同注册，解锁阶梯“豪”礼
- 【推荐】限时秒杀！国云大数据魔镜，企业级云分析平台

园子动态：

- 发起一个开源项目：博客引擎 fluss
- 云计算之路-新篇章-出海记：开篇
- 博客园2005年6月1日首页截图

最新新闻：

- 黄峥勇退：一年之内卸任CEO和董事长 想去“寻找幸福”
 - 百度二次上市，三重价值
 - 快手三年游戏路，路在何处？
 - 谷歌涂鸦庆祝爱尔兰圣帕特里克节
 - NASA的SMA轮胎技术即将商用 30倍于钢的可恢复应变
- » 更多新闻...