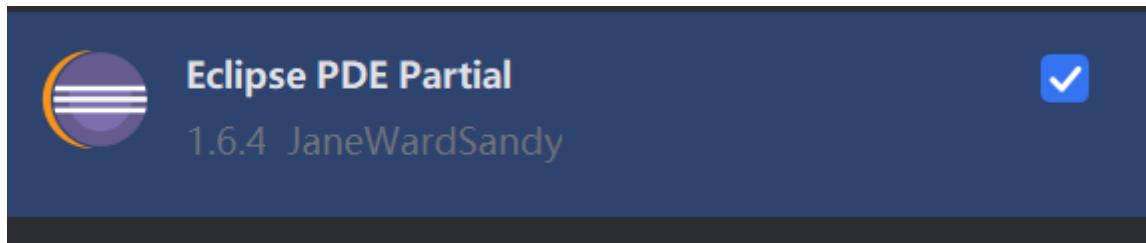


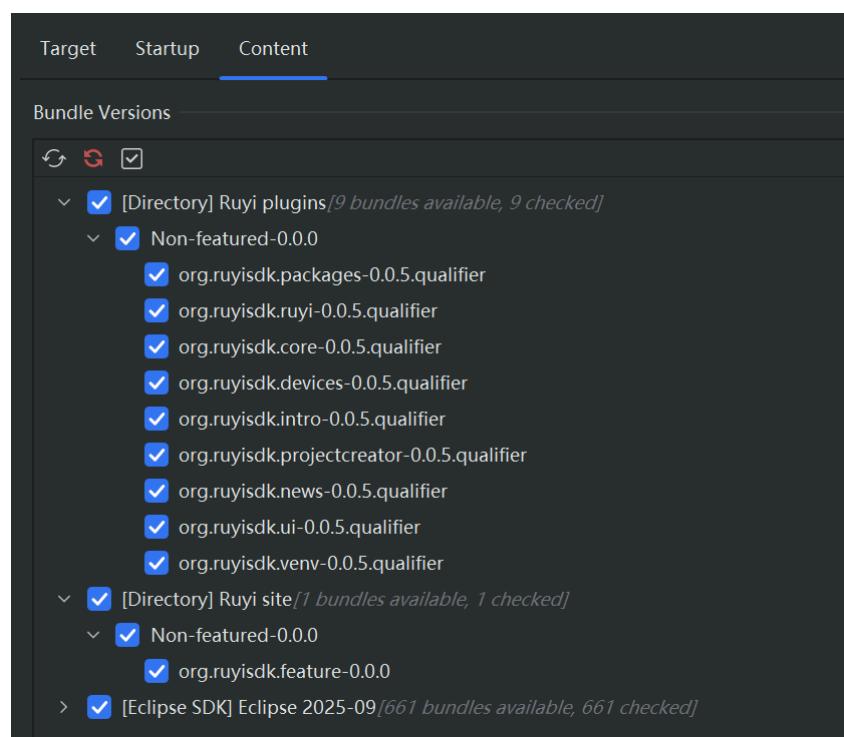
首先需要我们下载Eclipse PDE 插件, [Eclipse PDE Partial Plugin for JetBrains IDEs | JetBrains 市场](#)

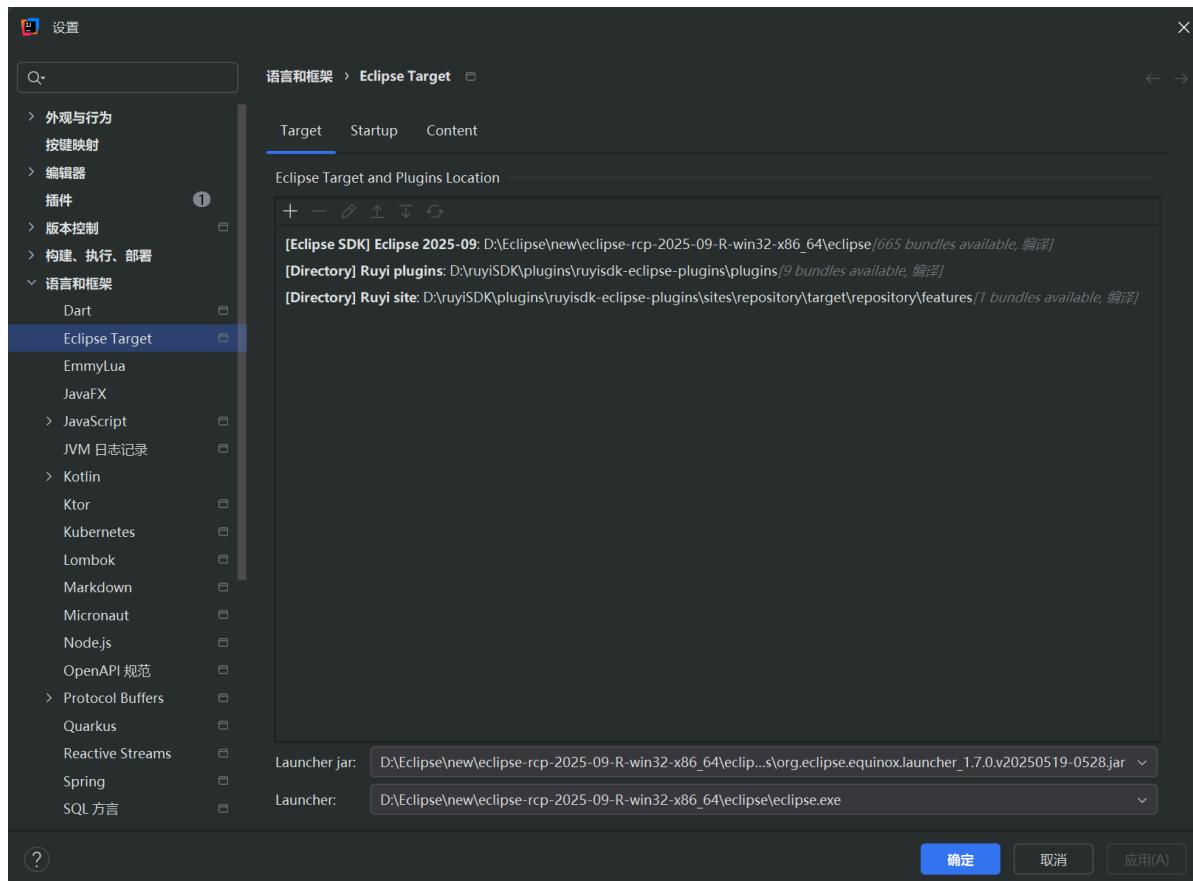


# 1、配置 Eclipse Target

步骤:

1. 打开配置: File → Settings → Languages & Frameworks → Eclipse Target
2. 添加 Eclipse 安装路径:
  - 点击 + → Eclipse Installation
  - 选择 Eclipse 安装目录 (包含 eclipse.exe 的目录)
  - 别名: Eclipse 2025-09 (最低要求2024-09)
3. 添加项目plugins目录:
  - 点击 + → Ruyi plugins
  - 路径: xx\plugins
4. 添加项目构建输出:
  - 点击 + → Ruyi site
  - 路径: xx\sites\repository\target\repository\plugins
5. 验证配置:
  - 切换到 Content 标签页
  - 确认能看到 Eclipse SDK 的 661 个 bundles





## 2. 创建 Eclipse Application Partial 运行配置

步骤：

1. 打开运行配置：Run → Edit Configurations...
2. 添加配置：点击 + → Eclipse Application Partial
3. 配置参数：
  - 名称：Debug RuyiSDK Eclipse Plugins
  - 运行于：本地机器
  - Product：org.eclipse.epp.package.rcp.product
  - Application：org.eclipse.ui.ide.workbench
  - JRE：选择 JDK 17-21
  - 虚拟机选项：  
-Xmx1024m  
-Dosgi.requiredJavaVersion=17

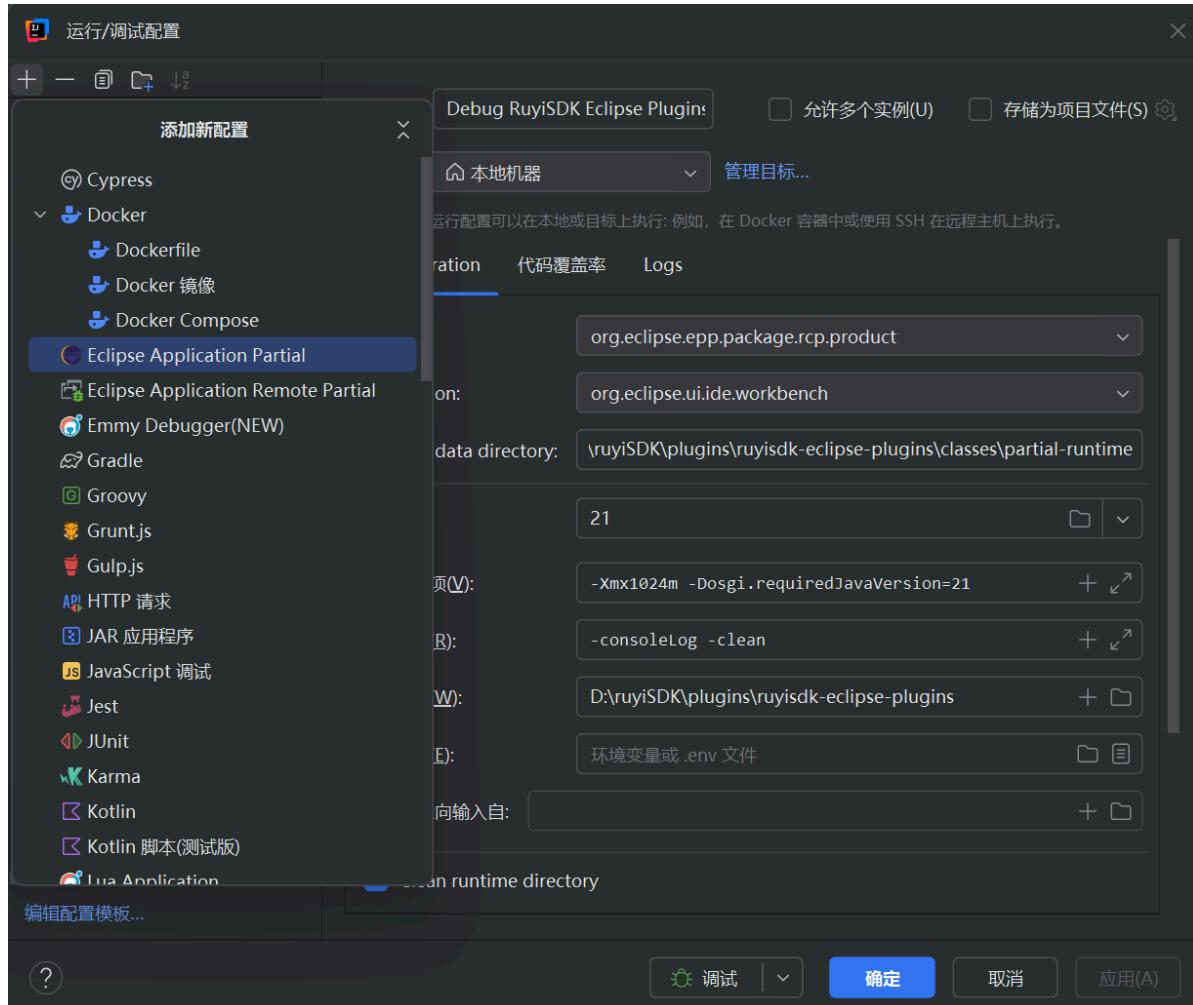
- 程序参数(可选)：  
-consoleLog  
-clean

- 工作目录：保持自动生成的路径
- Clean runtime directory (勾选)

#### 4. 配置 Before launch:

- 展开 "执行前(B)" 部分
- 确保有 "构建" 任务

#### 5. 应用并保存



## 3. 为模块添加 Eclipse PDE Partial Facet

**目的：**让运行配置识别插件模块并从编译输出加载

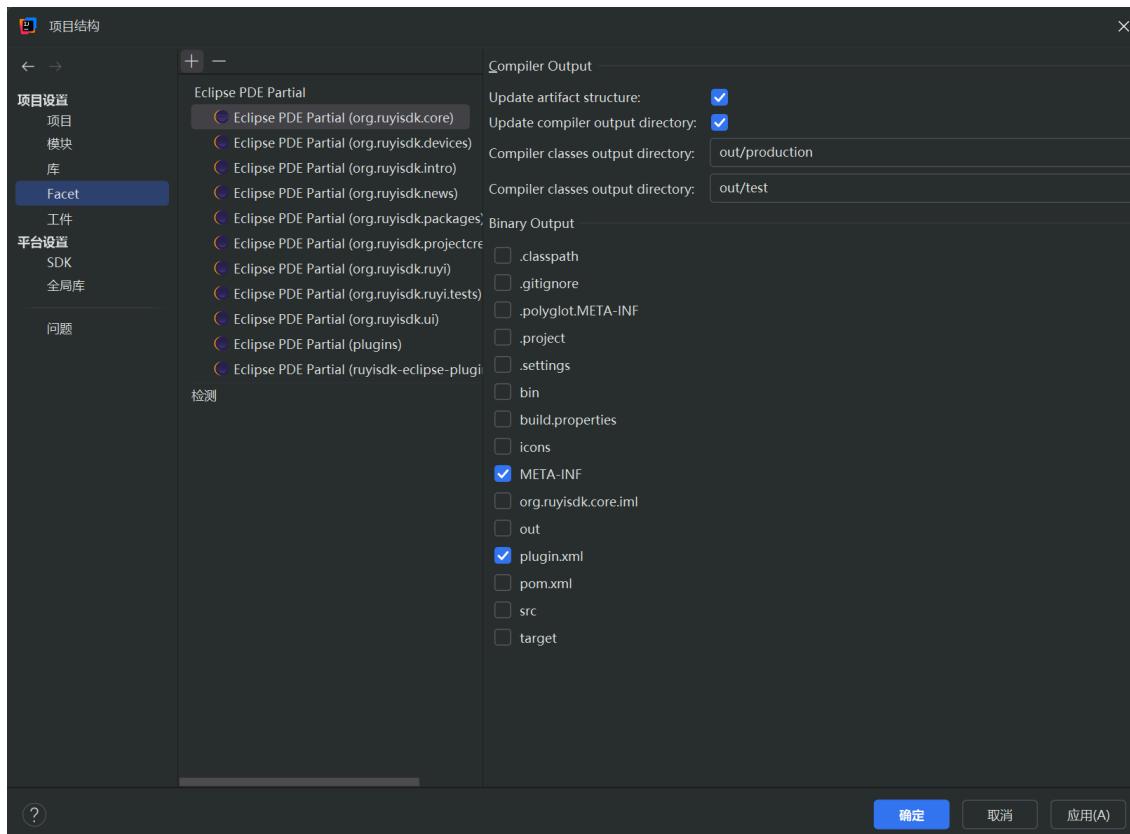
**操作步骤：**

1. 打开项目结构: `File → Project Structure → Modules`
2. 选择 **plugins** 父模块
3. 添加 **Facet**:

- 点击 `+` → Facet → Eclipse PDE Partial
- 在 Binary Output 中勾选所有 8 个插件模块:
  - org.ruyisdk.core
  - org.ruyisdk.devices
  - org.ruyisdk.intro
  - org.ruyisdk.news
  - org.ruyisdk.packages
  - org.ruyisdk.projectcreator

■ org.ruyisdk.ruyi

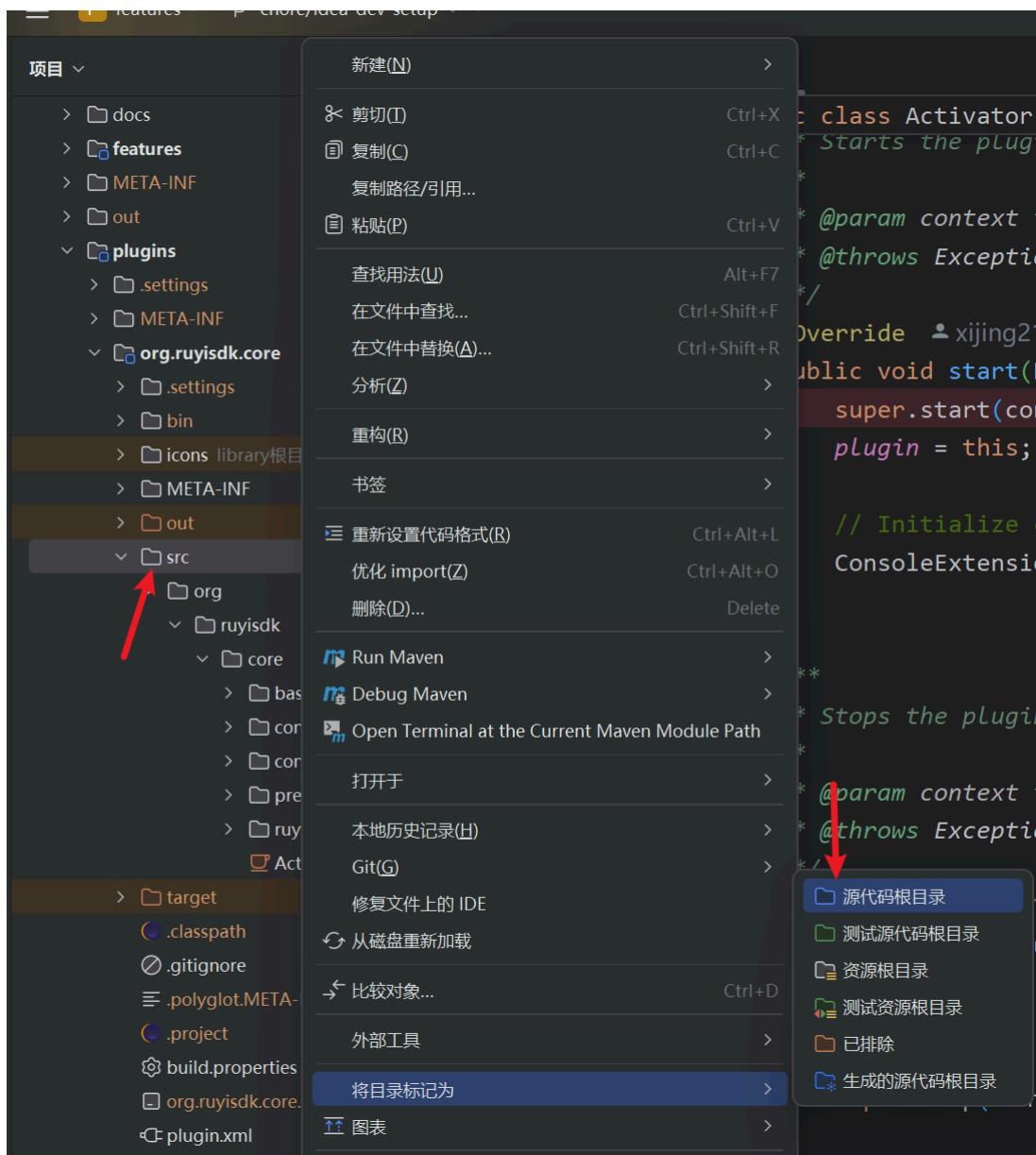
■ org.ruyisdk.ui



### 注意事项:

- Compiler Output 路径: `out/production` 和 `out/test` (IDEA 默认)
- 实际 Maven 输出在 `target/classes`, 但 Facet 配置会自动处理

# 4、最后需要把plugins里的src目录手动设置为源代码根目录



然后右键单击项目/模块 -> 选择 Resolve Workspace 以重新构建依赖树、库、构件等。

成功样例：

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with a tree view of files and folders.
- Editor:** Displays the `Activator.java` file content. The code is annotated with Javadoc and includes an `@Override` annotation above the `start` method. The `super.start(context);` line is highlighted in blue, indicating it is selected or being debugged.
- Debug View:** Located in the bottom-left corner, it shows the current stack trace for the running application. The stack trace starts with `main@1 在 ...ain 正在运行` and lists several frames from the `org.ruyisdk.core` package, such as `Activator`, `BundleContextImpl$2`, `AccessController`, and `doPrivileged$1`.
- Bottom Status Bar:** Shows the current file is `Activator.java`, the encoding is UTF-8, there are 4 个空格, and the line number is 32-1.