

# 阅读低代码平台Mendix之 RestServices-3 Publish REST

一起来读开源工业软件系列4

# 0.1 回顾上次讨论内容

## 1 先导知识

- 1.1 面向对象程序设计的特点
- 1.2 REST (Representational state transfer)
- 1.3 Java多线程
- 1.4 防御式编程
- 1.5 UML：活动图、类图、时序图

## 2 阅读RestConsume

- 应用面向对象程序设计方法
- 检查复杂度和思考可能的改善方法

## 3 建议学习内容

## 0.2 视频的观看对象

- 想学习如何阅读软件程序
- 熟悉Java语言的基础知识
- 了解Web应用系统架构
- 熟悉Java代码规范
- 了解Java开发工具

# 讨论内容

## 1 先导知识

- Microflows
- REST服务的类别

## 2 阅读Rest Publish

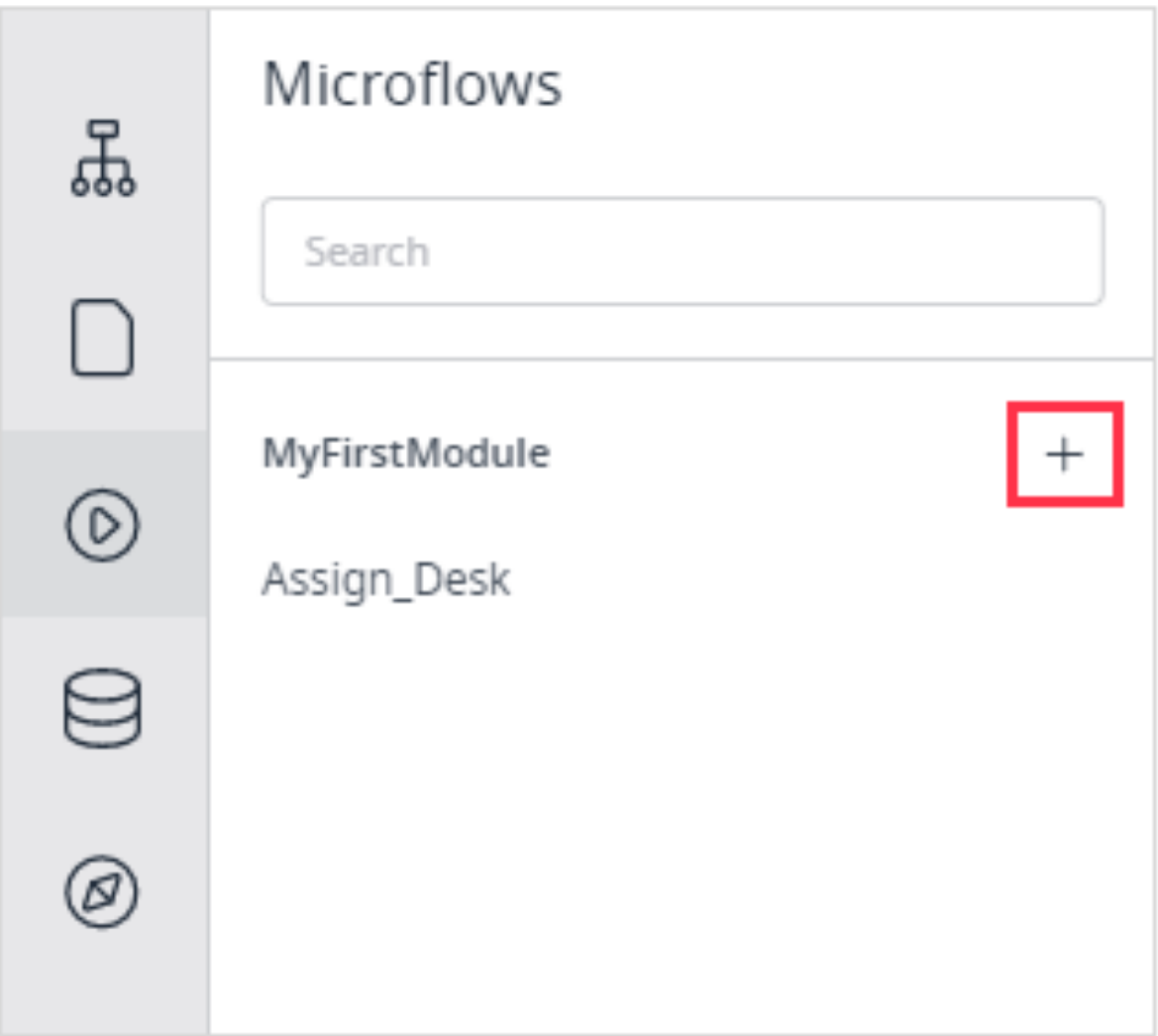
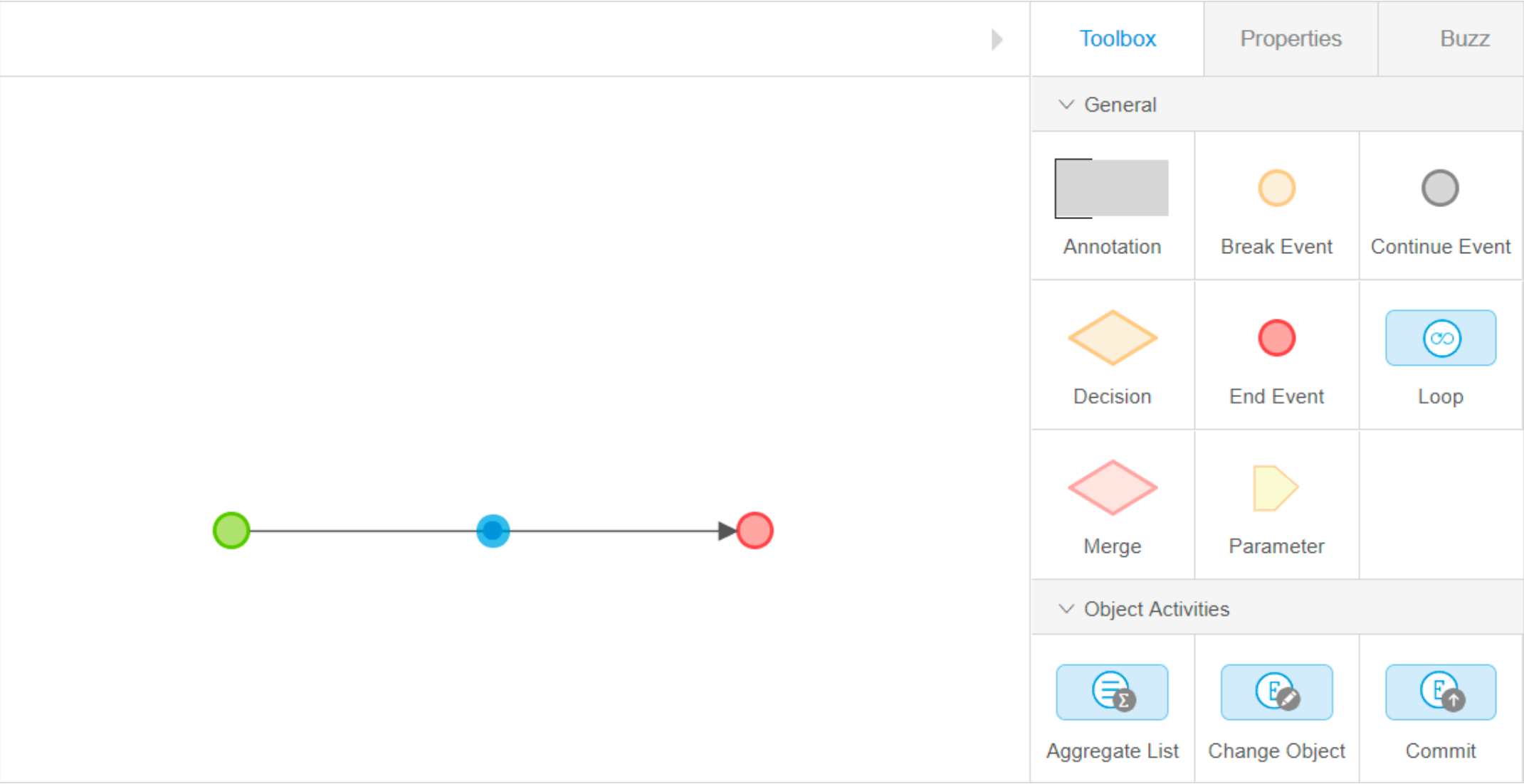
- 代码结构
- 应用面向对象程序设计方法
- 检查复杂度和思考可能的改善方法

## 3 建议学习内容

# 1 先导知识

# Microflow

- <https://docs.mendix.com/studio/microflows/>
- 概念：微流是一种表达文本程序代码的可视化方式。一个微流可以执行一些动作，如创建和改变对象，显示页面，和作出选择。一个微流看起来像一个**流程图**。
- 微流: 默认会创建一个开始事件（用绿点表示的微流的起点）和一个结束事件（用红点表示的微流的终点）。开始和结束事件由一个序列流（一条带箭头的线）连接，可以添加新的事件和活动。
- 使用场景
  - 要改变/扩展按钮的标准行为
  - 要添加自定义逻辑到你的应用程序
  - 要与其他系统、数据库、网络服务等整合
- 案例：
  - 检查终端用户输入的值，并向终端用户显示错误信息或另一个页面。
  - 正在创建一个待办事项列表，当列表上的一个项目的状态改变时，想使用自定义逻辑要在Studio中查看你的应用程序的微流，点击左边菜单栏的微流图标。



# REST服务

方法	描述
GET	
POST	
PUT	
DELETE	

## 2 阅读代码



# Mendix之RestServices

- REST service module for Mendix. Supports consuming and publishing REST based services and real-time data synchronization. Supports JSON, form-encoded, multipart and binary data transport. Mendix的REST服务模块。支持消费和发布基于REST的服务和实时数据同步。支持JSON、表格编码、多部分（或多段）和二进制数据传输。
- 文档链接：
  - <https://docs.mendix.com/refguide/integration/>
  - <https://docs.mendix.com/refguide/integration/rest-services/>
  - <https://docs.mendix.com/refguide/consumed-rest-services/>
  - <https://docs.mendix.com/refguide/published-rest-services/>
  - <https://docs.mendix.com/howto/integration/publish-rest-service/>
- 文件格式 <https://forum.mendix.com/p/questions/11540474045285303>
  - MPK: MPR + bundles files (JS, Java, Themes, data snapshot)
    - Can be used for exporting and sharing of full projects but also single Documents or Modules
  - MPR: The Mendix Model definition
  - MDA: Deployment package

# RestServices代码结构

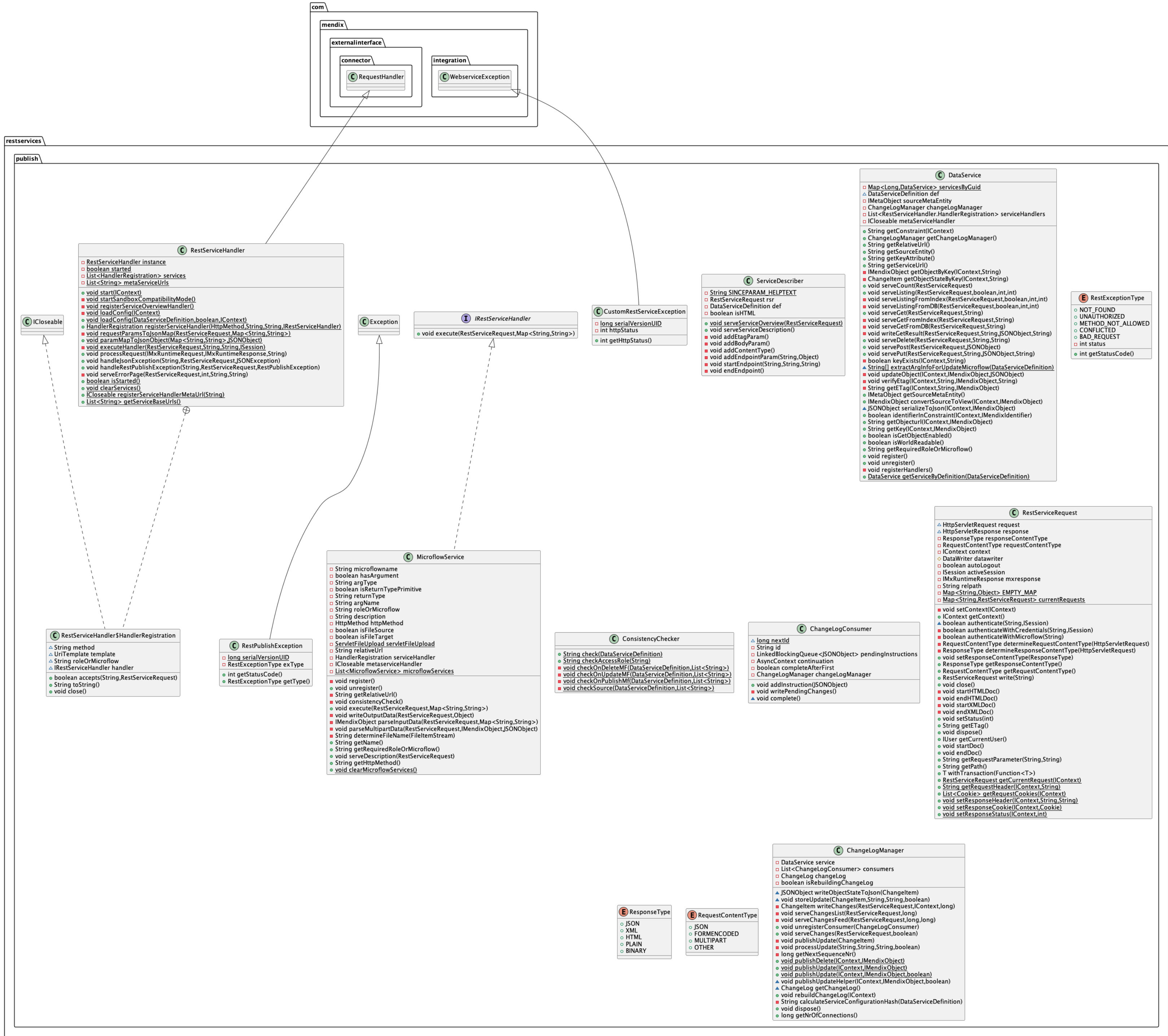
文件名	主要内容	文件格式
DIST	发布版本	.mpk mendix package包含一个Mendix项目或模块。
docs	文档	.md
images	图片	.png
javasource	communitycommons,restservices, system,tests,unittesting	.java
antisamy	用于防止Web应用程序中的跨站点脚本攻击（XSS）和其他安全漏洞	.xml，包名为resources.communitycommons
theme	主题风格	css,scss,json,ico,html

# 环境信息

- IntelliJ IDEA 2023.1.2 (Ultimate Edition) Build #IU-231.9011.34, built on May 16, 2023 Licensed to Shanghai XXXX Info Tech Co., Ltd. / XXXXXX Subscription is active until June 18, 2023.
- Runtime version: 17.0.6+10-b829.9 aarch64 VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o. macOS 13.4
- GC: G1 Young Generation, G1 Old Generation Memory: 4096M
- Cores: 8
- Metal Rendering is ON
- Non-Bundled Plugins:
  - plantuml-parser (0.0.5)
  - Statistic (4.2.7)
  - com.github.kisstkondoros.codemetrics (1.0.4)
  - com.caldonazzi.alessandro.uml (0.2.0)
  - com.alibaba.p3c.xenoamess (2.1.1.5x)
  - PlantUML integration (6.1.0-IJ2022.2)
  - MetricsReloaded (1.12)
  - SequenceDiagram (3.0.5)
  - com.kn.diagrams.generator.generator (2022.2.0)
- Kotlin: 231-1.8.21-IJ9011.34

# 阅读步骤

1. 寻找文档
2. 使用类图和序列图快速了解代码
3. 先读关键类再读辅助类
4. 检查代码复杂度和思考可能的改善方法



# RestServices/Publish代码结构

类名	功能	公共方法
RestServiceRequest	服务请求	getContext() setResponseContentType(ResponseType responseType) getResponseContentType() getRequestContentType() write(String data) close() setStatus(int status) getETag() dispose() getCurrentUser() startDoc() endDoc() getRequestParameter(String param, String defaultValue) getPath() withTransaction(final Function<T> worker) throws Exception getCurrentRequest(IContext context) getRequestHeader(IContext context, String headerName) getRequestCookies(IContext context) setResponseHeader(IContext context, String headerName, String value) setResponseCookie(IContext context, Cookie cookie) setResponseStatus(IContext context, int status)
RestPublishException	服务发布异常	getStatusCode() getType()

# RestServices/Publish代码结构

类名	功能	公共方法
DataService	数据服务	<div>getConstraint(IContext context)</div> <div>getChangeLogManager()</div> <div>getRelativeUrl()</div> <div>getSourceEntity()</div> <div>getKeyAttribute()</div> <div>getServiceUrl()</div> <div>serveCount(RestServiceRequest rsr) throws CoreException, RestPublishException</div> <div>serveListing(RestServiceRequest rsr, boolean includeData, int offset, int limit) throws Exception</div> <div>serveGet(RestServiceRequest rsr, String key)</div> <div>serveDelete(RestServiceRequest rsr, String key, String etag)</div> <div>servePost(RestServiceRequest rsr, JSONObject data) throws Exception</div> <div>servePut(RestServiceRequest rsr, String key, JSONObject data, String etag)</div> <div>getSourceMetaEntity()</div> <div>convertSourceToView(IContext context, IMendixObject source)</div> <div>identifierInConstraint(IContext c, IMendixIdentifier id) throws CoreException</div> <div>getObjecturl(IContext c, IMendixObject obj)</div> <div>getKey(IContext c, IMendixObject obj)</div> <div>isGetObjectEnabled()</div> <div>isWorldReadable()</div> <div>getRequiredRoleOrMicroflow()</div> <div>register()</div> <div>unregister()</div> <div>getServiceByDefinition(DataServiceDefinition def)</div>

# RestServices/Publish代码结构

类名	功能	公共方法
RestServiceHandler	服务处理器	start(IContext context) throws Exception loadConfig(DataServiceDefinition def, boolean throwOnFailure, IContext context) registerServiceHandler(HttpMethod method, String templatePath, String roleOrMicroflow, IRestServiceHandler handler) paramMapToJsonObject(Map<String, String> params, JSONObject data) processRequest(IMxRuntimeRequest req, IMxRuntimeResponse resp, String var3) handleJsonException(String requestStr, RestServiceRequest rsr, JSONException je) handleRestPublishException(String requestStr, RestServiceRequest rsr, RestPublishException rre) isStarted() clearServices() registerServiceHandlerMetaUrl(final String serviceBaseUrl) getServiceBaseUrls()
MicroflowService	微流服务	unregister() execute(final RestServiceRequest rsr, Map<String, String> params) getName() getRequiredRoleOrMicroflow() serveDescription(RestServiceRequest rsr) getHttpMethod() clearMicroflowServices()



# RestServices/Publish代码结构

类名	功能	公共方法
ChangeLogConsumer	变更日志消费者	addInstruction(JSONObject json)
ChangeLogManager	变更日志管理器	dispose() getNrOfConnections() rebuildChangeLog(IContext context) throws CoreException publishUpdate(IContext context, IMendixObject source, boolean checkConstraint) publishUpdate(IContext context, IMendixObject source) publishDelete(IContext context, IMendixObject source) serveChanges(RestServiceRequest rsr, boolean asFeed) throws IOException, CoreException, RestPublishException unregisterConsumer(ChangeLogConsumer lpsession)
ConsistencyChecker	检查一致性	check(DataServiceDefinition def) checkAccessRole(String accessRole)
ServiceDescriber	描述服务	serveServiceOverview(RestServiceRequest rsr) serveServiceDescription()
CustomRestServiceException	定制服务异常	getHttpStatus()
IRestServiceHandler	服务处理器接口	execute(RestServiceRequest rsr, Map<String, String> params)

# 建议练习

- 用UML类图和序列图快速了解代码：
  - <https://plantuml.com/>
  - <https://plugins.jetbrains.com/plugin/7017-plantuml-integration>
  - <https://plugins.jetbrains.com/plugin/15524-plantuml-parser>
  - <https://plugins.jetbrains.com/plugin/15991-plantuml-diagram-generator>
  - <https://vanco.github.io/SequencePlugin/>
- 用代码复杂度工具检查代码复杂度
  - MetricsReloaded: <https://plugins.jetbrains.com/plugin/93-metricsreloaded>
  - CodeMetrics: <https://plugins.jetbrains.com/plugin/12159-codemetrics> 在程序中显示复杂度

# 总结

## 1 先导知识

- Microflows
- REST服务的类别

## 2 阅读Rest Publish

- 代码结构
- 应用面向对象程序设计方法
- 检查复杂度和思考可能的改善方法

## 3 建议学习内容

下一节，  
阅读RestServices/Actions-Util