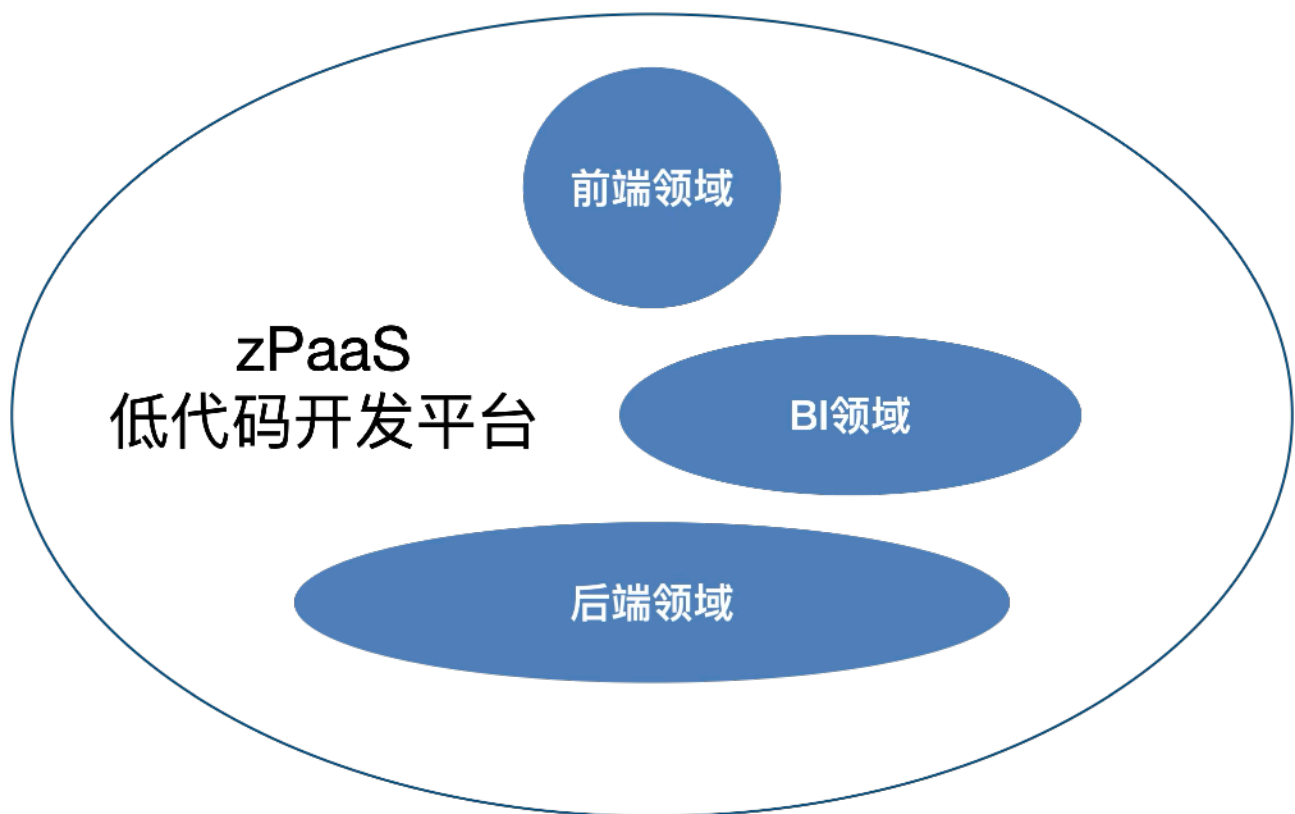


# 核心领域对象及组织形式

接上一篇《低代码设计目标及核心设计》，本篇主要介绍一下zPaaS低代码开发平台的领域划分、核心领域对象及这些领域对象在低代码平台中的组织及管理形式。

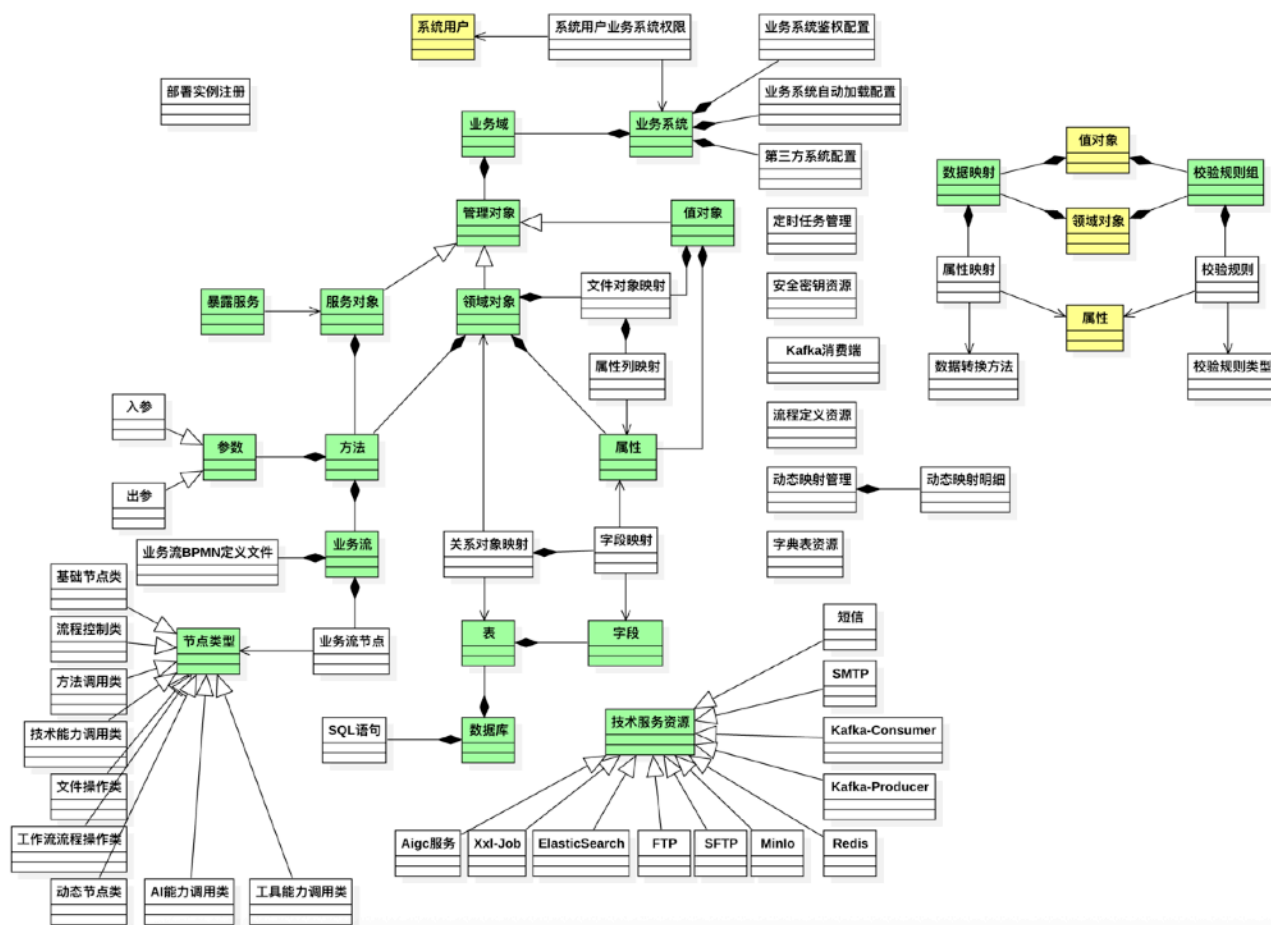
## 1. 领域划分



根据zPaaS低代码平台的定位和设计目标，将平台划分为后端、前端以及BI端三个业务领域：

- **后端领域**：实现服务端核心业务的低代码开发能力，包括业务对象管理、业务逻辑编排、工作流程编排等，以及相应的解析执行引擎。
- **BI端领域**：实现BI服务端核心业务的低代码开发能力，包括数据模型、数据集、指标、标签、筛选器等对象的管理，以及相应的解析执行引擎。
- **前端领域**：实现前端页面的低代码开发能力，包括自定义页面、模板页面、统计报表、可视化图表等页面类型的低代码开发能力，以及相应的解析执行引擎。

## 2. 后端领域的核心领域对象

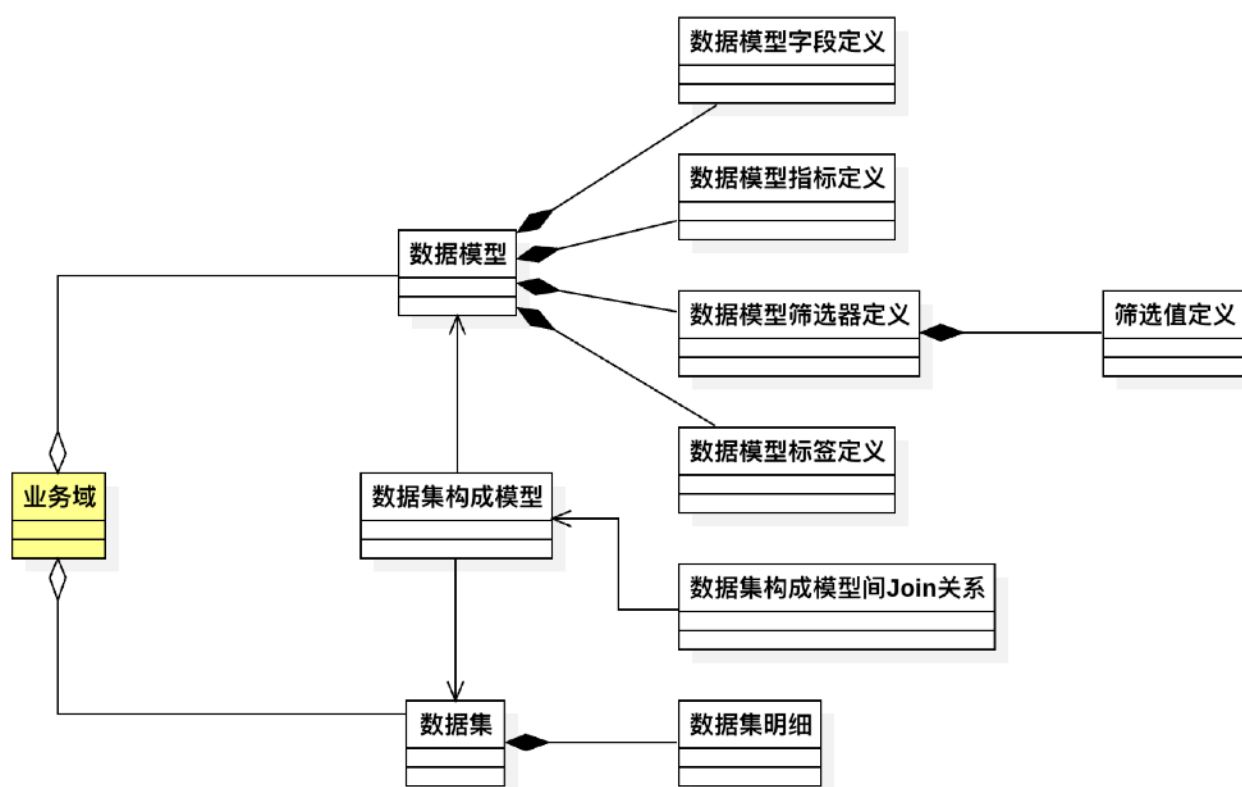


后端领域结合领域驱动、面向对象以及分层设计的思想，以暴露服务--服务对象/领域对象--方法--业务流--ORM--数据库表作为主线，整合工作流、参数校验、数据转换、文件对象转换以及其他常用的各种技术服务能力，提供后端核心业务的低代码开发能力：

- **业务系统**：后端领域的顶级业务对象，对应实际业务中的项目/系统/子系统。低代码平台管理的所有业务对象都以业务系统为顶级维度进行管理，包括前端、后端以及BI端所有的业务对象。
- **业务域**：业务系统下业务对象的逻辑管理单位，可以理解为系统下的业务模块。
- **暴露服务**：将服务对象中的方法按HTTP REST的方法对外进行暴露，供前端/外部进行调用。暴露服务通过统一的FrontController对外提供服务，FrontController可以理解为分层设计中的控制层。
- **服务对象**：对应用领域驱动设计中的应用服务、分层设计中的服务层。服务对象只允许包含方法，不允许包含属性。服务对象的核心是用来组织和实现对外暴露/跨领域对象的服务方法。
- **领域对象**：对应领域驱动设计中的领域对象（简化了聚合根以及值对象）、面向对象设计中的业务对象、分层设计中的业务逻辑层。领域对象采用充血模型，同时包含对象属性和对象方法。
- **值对象**：也叫值传递对象，对应分层设计中的ValueObject/DTO，用于在前端/控制层/服务层/业务逻辑层之前的数据传递。值对象

- **方法**：对应开发领域中的对象方法/函数，主要用于承载业务逻辑的实现，包括方法定义、出参、入参以及业务流等实体组成。
- **业务流**：由一个或多个业务流节点，通过编排的方式，实现后端的各种业务逻辑。是后端最核心的领域对象，通过业务流的各类节点，串连参数校验、数据转换、文件对象转换、ORM、数据库、工作流以及其他常用的各种技术服务能力。
- **ORM对象关系映射**：通过ORM实现基本的数据库增删改查能力，ORM串联了领域对象与数据库表。
- **数据库/表/字段**：对应关系数据库中的Schema/Table/Column。
- **数据映射**：通过数据映射实现领域对象/值对象之间的数据转换。
- **校验规则**：通过校验规则实现暴露服务入参、领域对象、值对象的参数校验能力。
- **文件对象映射**：通过文件对象映射实现文件与对象之间的数据转换能力，即Excel/JSON/文本文件的数据导入/导出能力。
- **技术服务资源**：通过技术服务资源将常用的各种技术能力引入低代码，再结合业务流编排中对应的节点，实现对各种技术能力的调用。

### 3. BI端的核心领域对象

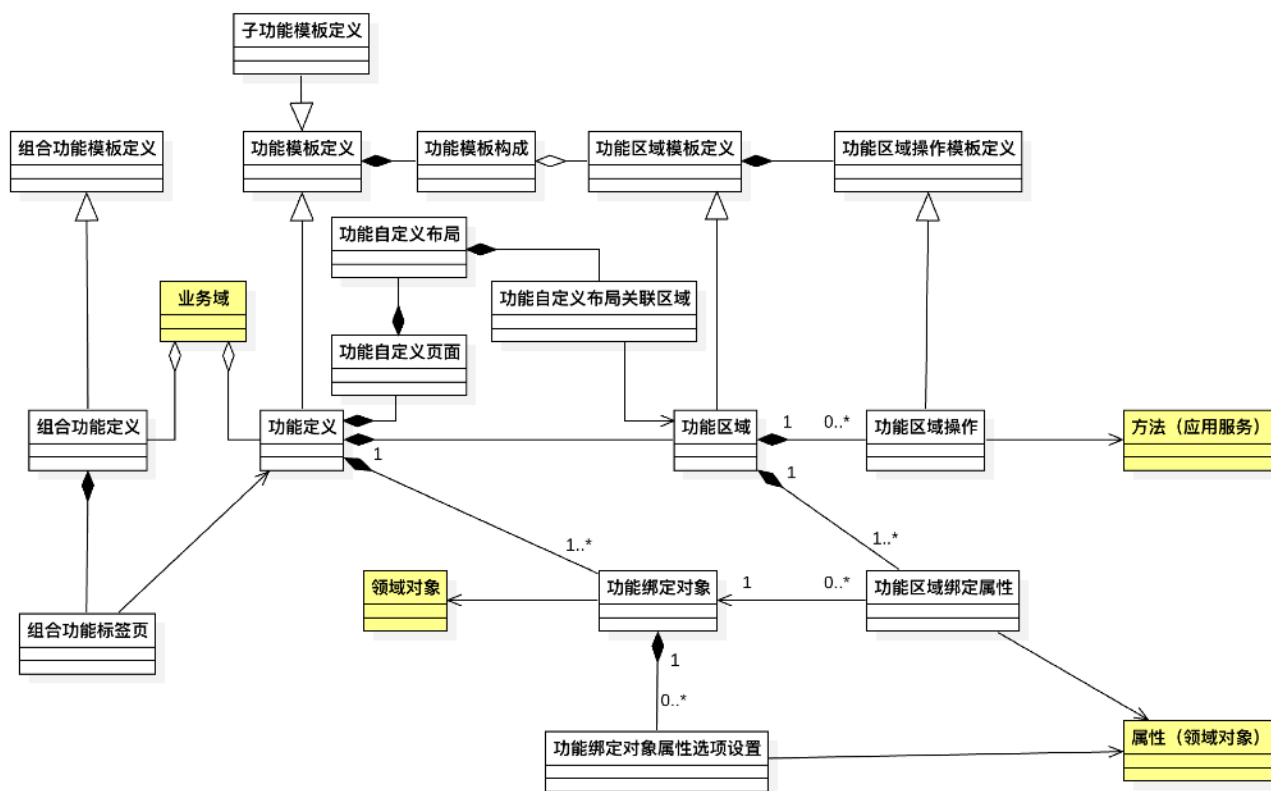


为了区分BI端不同层次的使用人员（偏技术/偏业务），将BI端领域的核心领域对象分为数据模型和数据集两层：

- **数据模型**：数据模型主要面向有一定技术能力的人员使用，是对物理数据模型的抽象，对应物理的关系型数据库表/NoSql数据库表/数据文件，抽象为数据模型以及模型字段，同时在此基础这上，针对BI领域的特点，抽象了指标、标签以及筛选器等领域对象。

- **数据集**：在数据模型基础上抽象的领域对象，主要偏向业务人员使用，是进行BI统计分析的直接对象。一个数据集由一个或多个数据模型组成，数据集明细字段对应数据模型的字段/字标/标签/筛选器。当一个数据集由多个数据模型组成时，需要指定不同数据模型之间的JOIN关系。

#### 4. 前端的核心领域对象

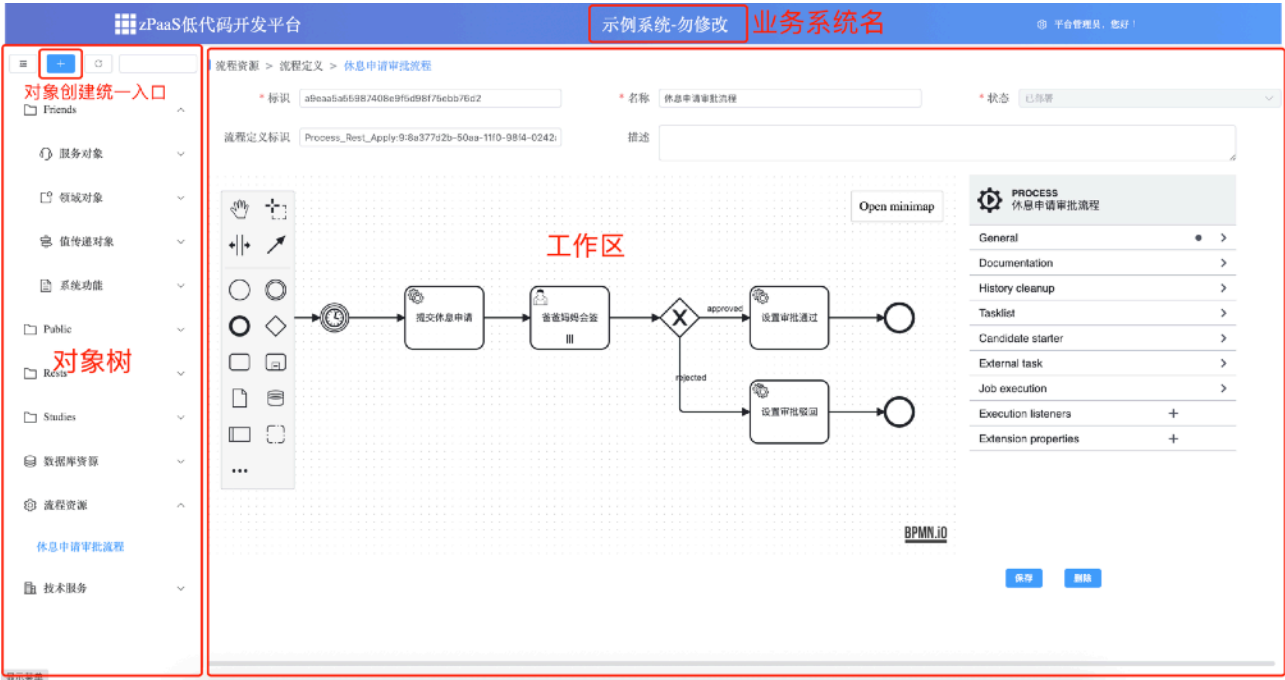


前端领域的核心是功能模板以及功能模板为实现相应能力抽象的各类领域对象：

- **组合功能及组合功能模板**：将多个功能组合成一个更大的功能，如标签页式的组合功能，一个标签页对应一个功能。
- **功能及功能/子功能模板**：对应一个具体的功能，按各类功能的特点进行抽象，形成功能模板，用于功能的快速实现。目前已经实现的功能模板包括简单（单表）管理功能模板、多表管理功能模板、查询选择子功能模板、单表增/改/查子功能模板、多表增/改/查子功能模板、统计报表功能模板、可视化图表功能模板以及自定义功能模板。自定义功能模板是一类特殊的功能模板，可以自定义实现功能的布局与交互，支持表单/报表/图表的混合编排。
- **功能绑定对象及对象属性选项设置**：功能对应后端的业务对象，支持领域对象和值传递对象。支持在业务对象以及属性上进行共性逻辑的设置，包括下拉选项、属性间联动、弹出设置、翻译以及高级查询能力（模糊查询、范围查询、多值查询以及比较查询等）等选项的设置，可以避免重复的配置。功能绑定对象对应后端的领域对象/值对象。
- **功能区域/区域操作/区域绑定属性**：对应功能中的子页面以及页面中编排的属性和按钮。如简单管理功能模板，将一个完整的功能抽象为查询区、结果区、新增区、编辑区

和明细区。功能区域的操作一般会绑定后端的暴露服务方法。功能区域绑定属性对应后端领域对象/值对象的属性。

5. zPaaS低代码开发平台的对象组织及管理形式



以上是低代码开发平台的工作台，主要由对象树以及工作区两个部分组成。业务对象的创建由对象创建统一入口进入支持19类业务对象的创建：



## 6. 其他

低代码平台的源码已经提交到GitHub：<https://github.com/zjyzju/zPaaS-lowcode>，后续将通过更多的文章，逐步对低代码平台的设计理念/思想、具体的设计、部署以及操作说明进行细化，敬请关注。

另外在一台云服务器上部署了一个演示环境，有兴趣的小伙伴可以通过演示环境进行试用：

访问地址：<https://129.153.118.144:11443>

试用账户：`demo/adl28m2@ma(a!*D2`

`demo1/adl28m2@ma(a!*D2`

注：（1）该演示环境部署在一台免费的云服务器上，且位于国外，在国内访问响应会比较慢

（2）系统管理子系统的菜单框架引用的@micro-zoe/micro-app@1.0.0-rc.24组件，该组件在不同浏览器中存在兼容性问题，苹果的safari浏览器访问一切正常，但是windows的Edge和谷歌的Chrome会存在页面刷不出来的情况，需要多等一会儿（有解决方案的可以邮件发给我，多谢！<https://github.com/jd-opensource/micro-app/issues/1559#issuecomment-2784997902>）。

如果有低代码平台以及AI相关的讨论，可以发送邮件到我的邮箱：[zjyzju@163.com](mailto:zjyzju@163.com)。