

软件架构设计文档

软件架构设计文档

1. 引言

- 1.1 编写目的
- 1.2 对象与范围
- 1.3 参考文献
- 1.4 名词与术语

2. 逻辑视角

- 2.1 分层架构图
- 2.2 逻辑包图

3. 组合视角

- 3.1 物理包的划分
- 3.2 物理包图

4. 接口视角

- 4.1 模块的职责
- 4.2 模块的接口规范

Component

Api

UserController

AccountService

GraphService

JsonTransService

5. 信息视角

- 5.2 PO的定义

1. 引言

1.1 编写目的

本文档的目标读者是团队内部的开发者与管理者，用于指导之后的代码开发/测试/维护工作。

1.2 对象与范围

本文为知识图谱软件 `nioCoin` 的系统架构设计文档，采用多视图模型描述了系统架构的各方面，用于指导开发工程，记录开发决策，为系统维护者提供指导。

1.3 参考文献

[维基百科：4+1架构视图模型](#)

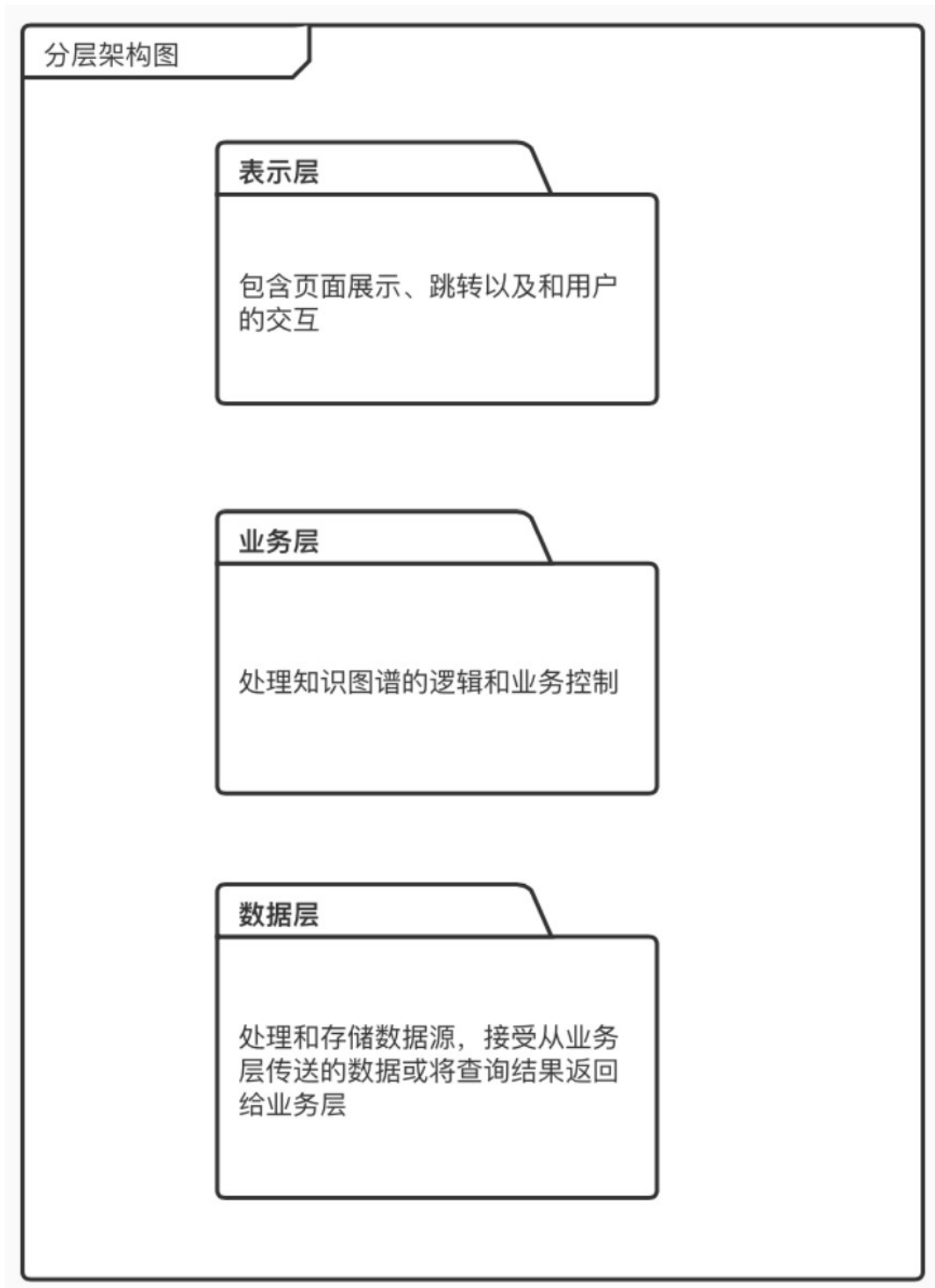
[软件工程与计算\[三\]——团队与软件开发实践，邵栋，刘嘉，张谨玉，黄莹](#)

1.4 名词与术语

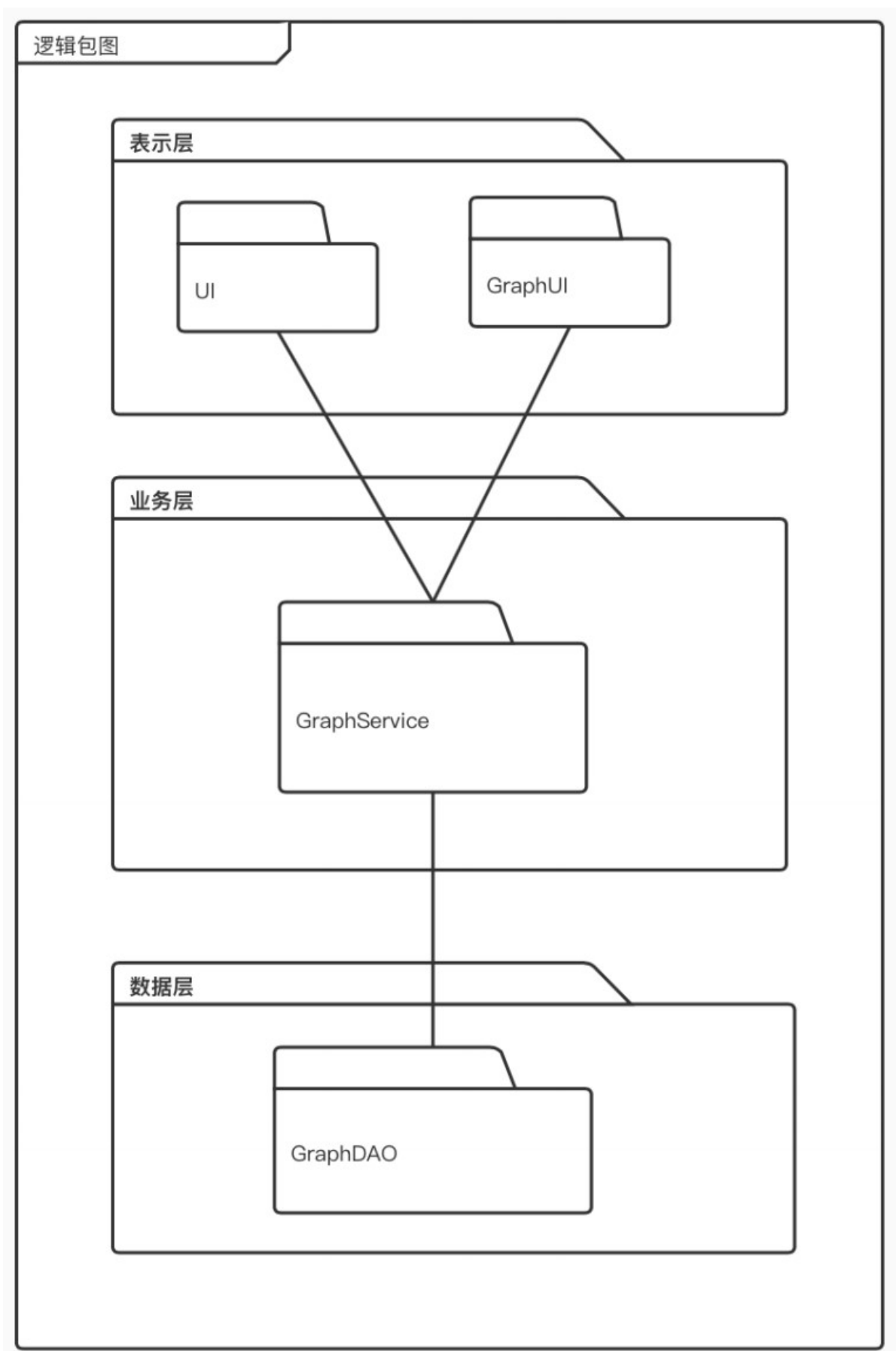
`nioCoin`: 项目名称

2. 逻辑视角

2.1 分层架构图



2.2 逻辑包图

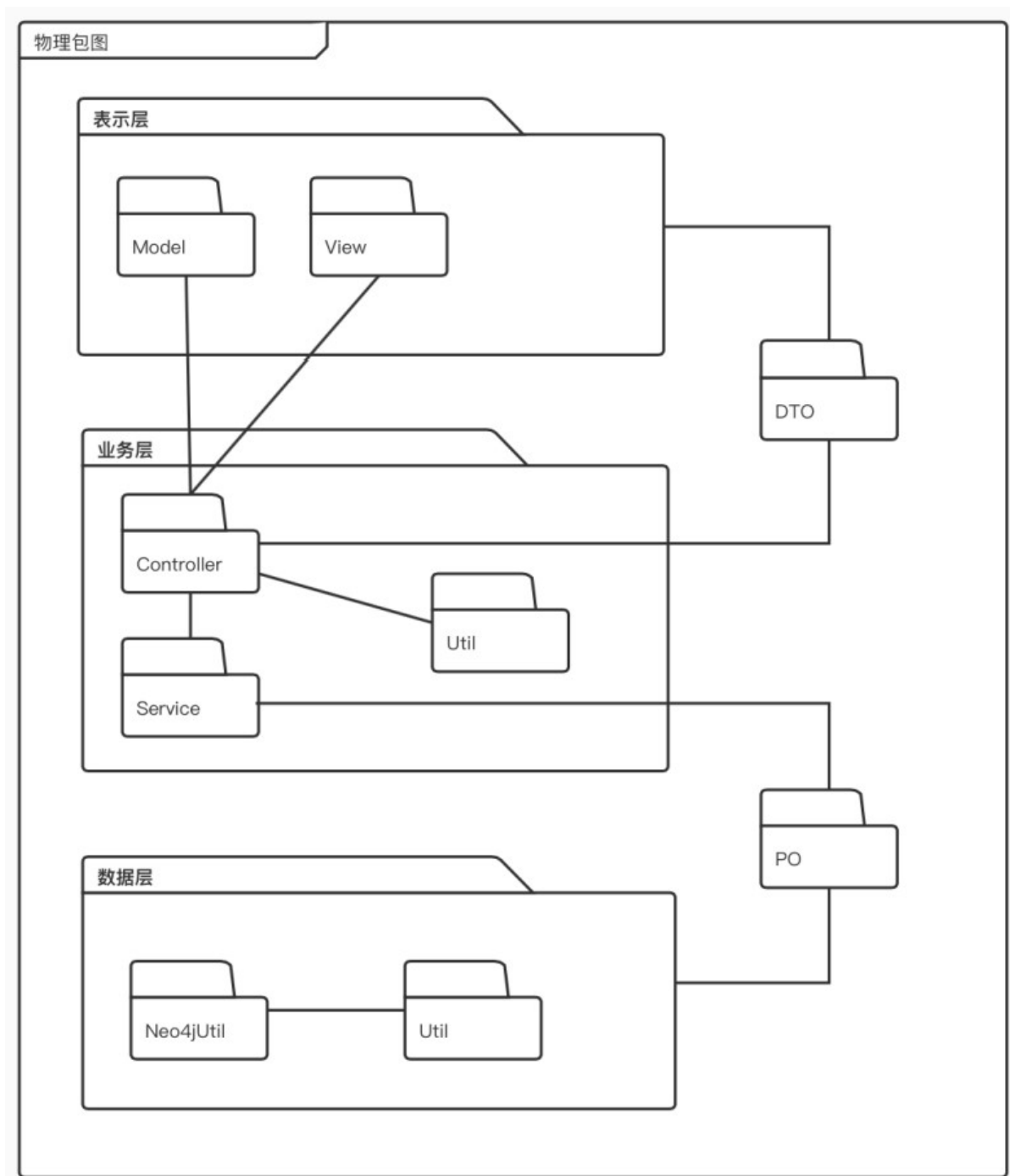


3. 组合视角

3.1 物理包的划分

开发包	依赖的其他开发包
controller	dto, service, util
service	dto, mapper, po, util
util	po
dto	无
po	无
config	无
mapper	po, dto

3.2 物理包图



4. 接口视角

4.1 模块的职责

模块	职责	层级
Component	界面的布局/展示/逻辑 (Vue SFC)	展示层
Api	统一管理接口调用	展示层
UserController	用户数据实体 Restful 接口	控制器层
AccountService	用户账户的登录与注册	业务层
GraphService	图数据库增删改查	业务层
JsonTransService	Json 对象的解析与存储	业务层

4.2 模块的接口规范

Component

需接口统一由 Api 模块提供。

Api

接口	功能
getGraphDataAPI()	返回当前数据库内容
updateGraphDataAPI(kgId, graphDTO)	更新kgId对应的图谱信息
generateNewGraphAPI(userId, graphName)	新建空图谱
GetGraphNameAPI(userId, graphId)	获取图谱名称
loginAPI(userDTO)	用户登录
registerAPI(userDTO)	用户注册
getAllGraphAPI(userId)	获取用户所有图谱信息
renameGraphAPI(userId, graphInfoDTO)	重命名图谱
deleteGraphAPI(kgId, userId)	删除用户kgId对应的图谱

UserController

接口	URL	功能
Res register(@RequestBody userDTO userDTO)	@PostMapping("/register")	注册用户
Res login(@RequestBody userDTO userDTO)	@PostMapping("/login")	用户登录
Res transJSON(@RequestParam long kgId)	@GetMapping("/getGraph")	根据kgId 返回所有的 节点和 关系信息
Res getGraphName(@RequestParam long userId, @RequestParam long graphId)	@GetMapping("/getGraphName")	根据 userId和 graphID 返回当前 图谱名
Res clearCurrent(@RequestParam long kgId, @RequestParam long userId)	@GetMapping("/deleteGraph")	清空指定 的图谱的 所有数据
Res test()	@GetMapping("/test")	此接口仅 作为服务 器上的后 端运行的 检测端 口，没有 用处
Res updateGraph(@RequestParam long kgId, @RequestBody graphDTO graphDTO)	@PostMapping("/graph")	更新以及 导入图谱 时使用的 接口，接 收json数 据格式， 对图谱进 行更新或 者导入
Res getNewGraphId(@RequestParam long userId, @RequestParam String graphName)	@GetMapping("/newGraphId")	在新建一 张图谱时 首先使 用，将创 建一个 kgId供前 端使用
Res updateGraphName(@RequestParam long userId, @RequestBody graphInfoDTO graphInfoDTO)	@PostMapping("/graphRename")	对图谱进 行重命名

接口	URL	功能
Res getAllGraph(@RequestParam long userId)	@GetMapping("/allGraph")	根据指定的userId 获取当前用户所有的知识图谱

AccountService

接口	功能
String register(userDTO userDto)	用户注册
User login(userDTO userDto)	用户登录

GraphService

接口	功能
tring parseJson(graphDTO data, long kgId)	解析JSON数据
long GenerateNewGraph(long userId, String graphName)	生成新的图谱
String updateGraphName(graphInfoDTO newGraph)	更新图谱名称
List getAllGraph(long userId)	获取用户的全部图谱信息
String getGraphName(long userId,long graphId)	获取图谱名称

JsonTransService

接口	功能
JSONObject generateJSONByKgId(long kgId)	通过kg_id生成JSON
void setKeysAndValues(JSONArray relationJSONArray, Map<String, Object> relation, JSONObject relationJSONObject, JSONObject relationAttributeJSONObject)	设置属性键值

5. 信息视角

node

字段	属性	含义
id	String	节点id
label	String	节点标签
attributes	Map<String, String>	节点属性

relation

字段	属性	含义
source	String	源节点
target	String	目标节点
label	String	关系
attributes	Map<String, String>	关系属性

graphDTO

字段	属性	含义
nodes	List	节点列表
edges	List	关系列表

graphInfoDTO

字段	属性	含义
graphId	String	图谱id
graphName	String	图谱名称
time	Timestamp	时间戳

userDTO

字段	属性	含义
username	String	用户名称
password	String	登录密码
email	String	注册邮箱

5.2 PO的定义

Entity

字段	属性	含义
id	Long	实体id
kg_id	Long	图id
label	String	实体名
attributeKeys	String	属性键
attributeValues	String	属性值

Relation

字段	属性	含义
entity1_id	Long	实体1id
entity2_id	Long	实体2id
kg_id	Long	图谱id
similarity	double	相似度
relationName	String	关系名称
attributeKeys	String	属性键
attributeValues	String	属性值