

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
 - GraphController
 - Kafka Controller
 - ModifyController
 - NREController
 - AccountService
 - GraphService
 - JsonTransService
 - mailService
 - modifyService
 - NREService

5. 信息视角

- 5.1 DTO的定义
- 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](#)

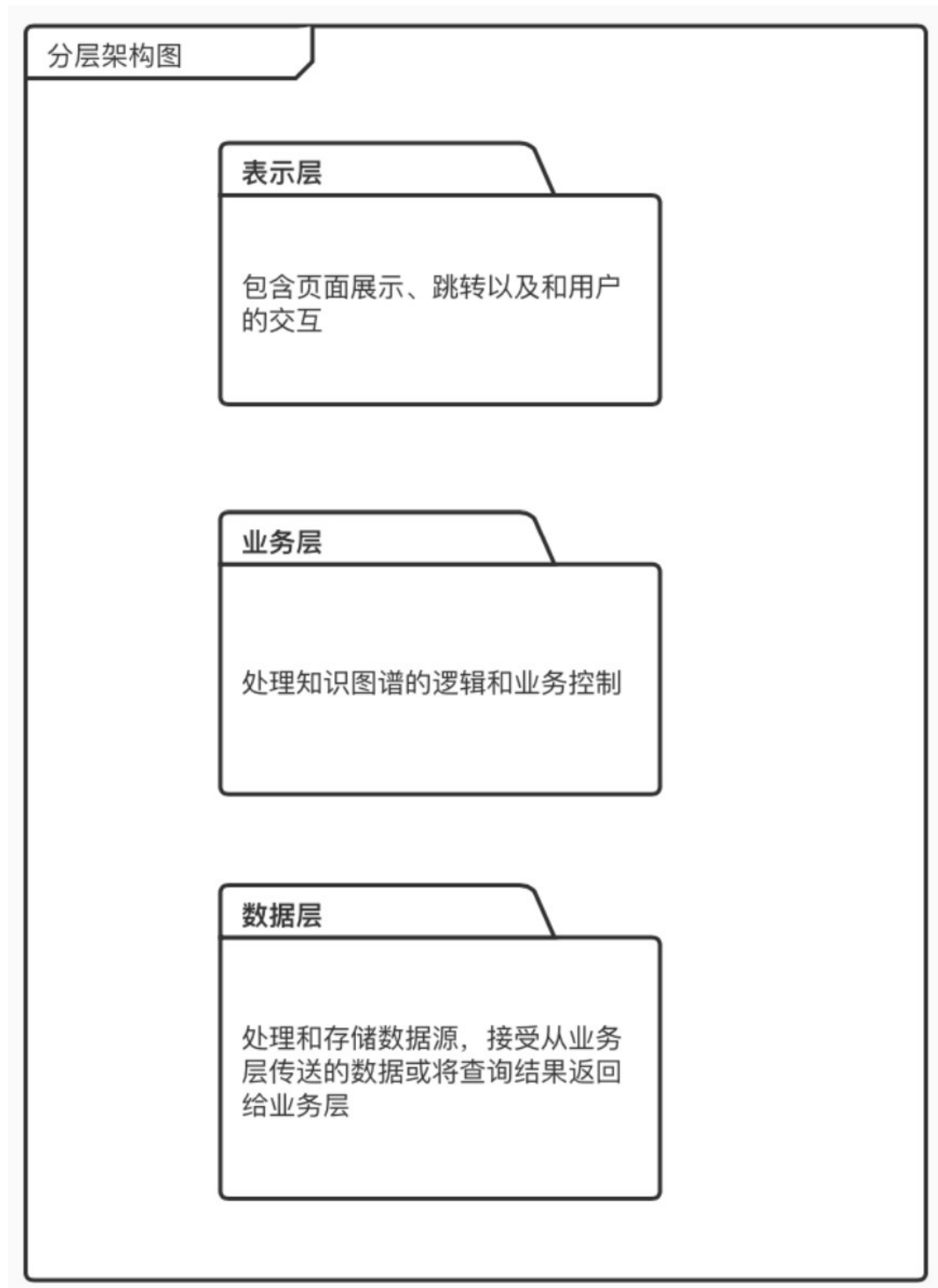
[软件工程与计算\[三\]——团队与软件开发实践, 邵栋, 刘嘉, 张谨玉, 黄董](#)

1.4 名词与术语

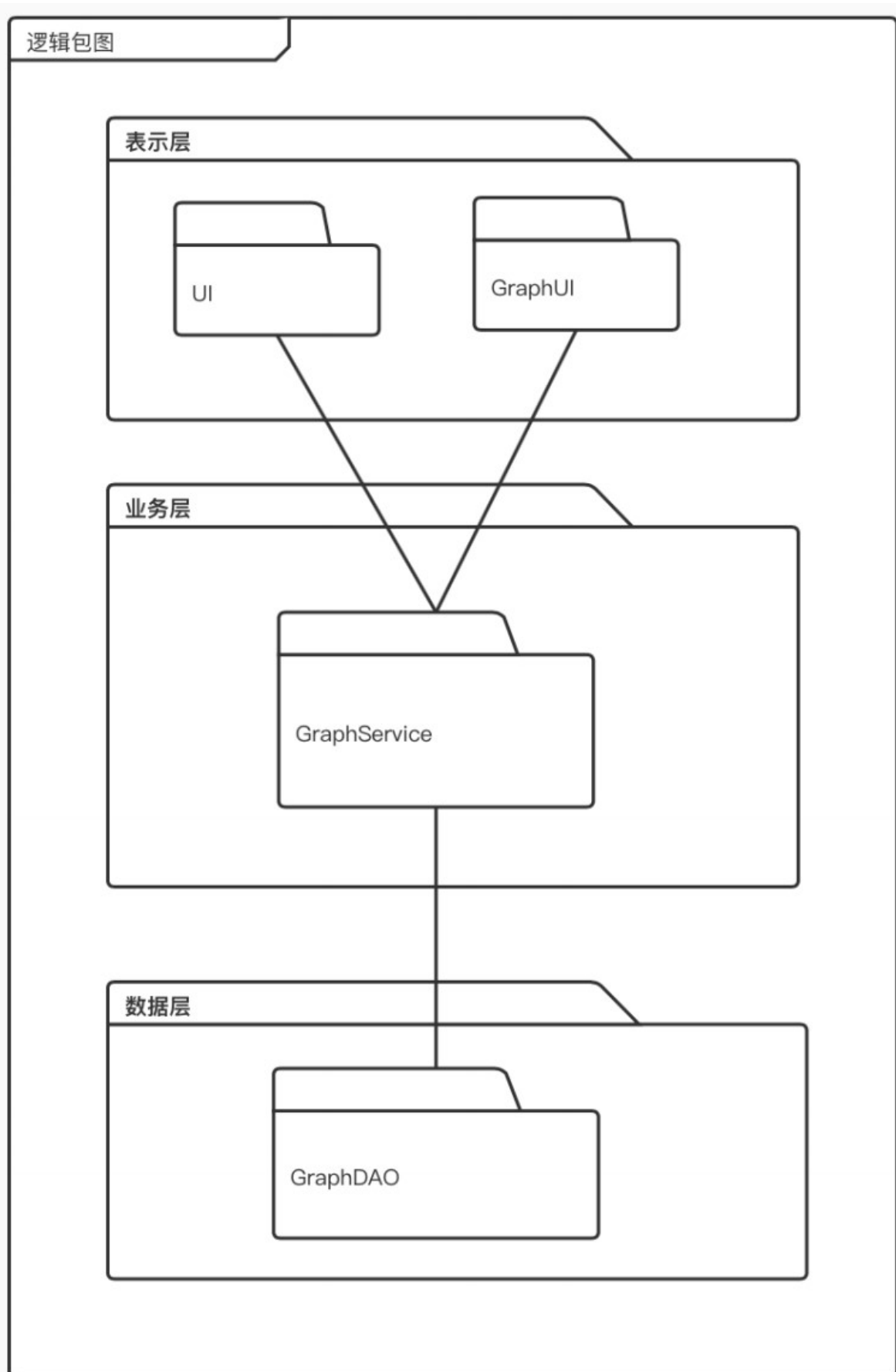
nioCoin: 项目名称

2. 逻辑视角

2.1 分层架构图



2.2 逻辑包图

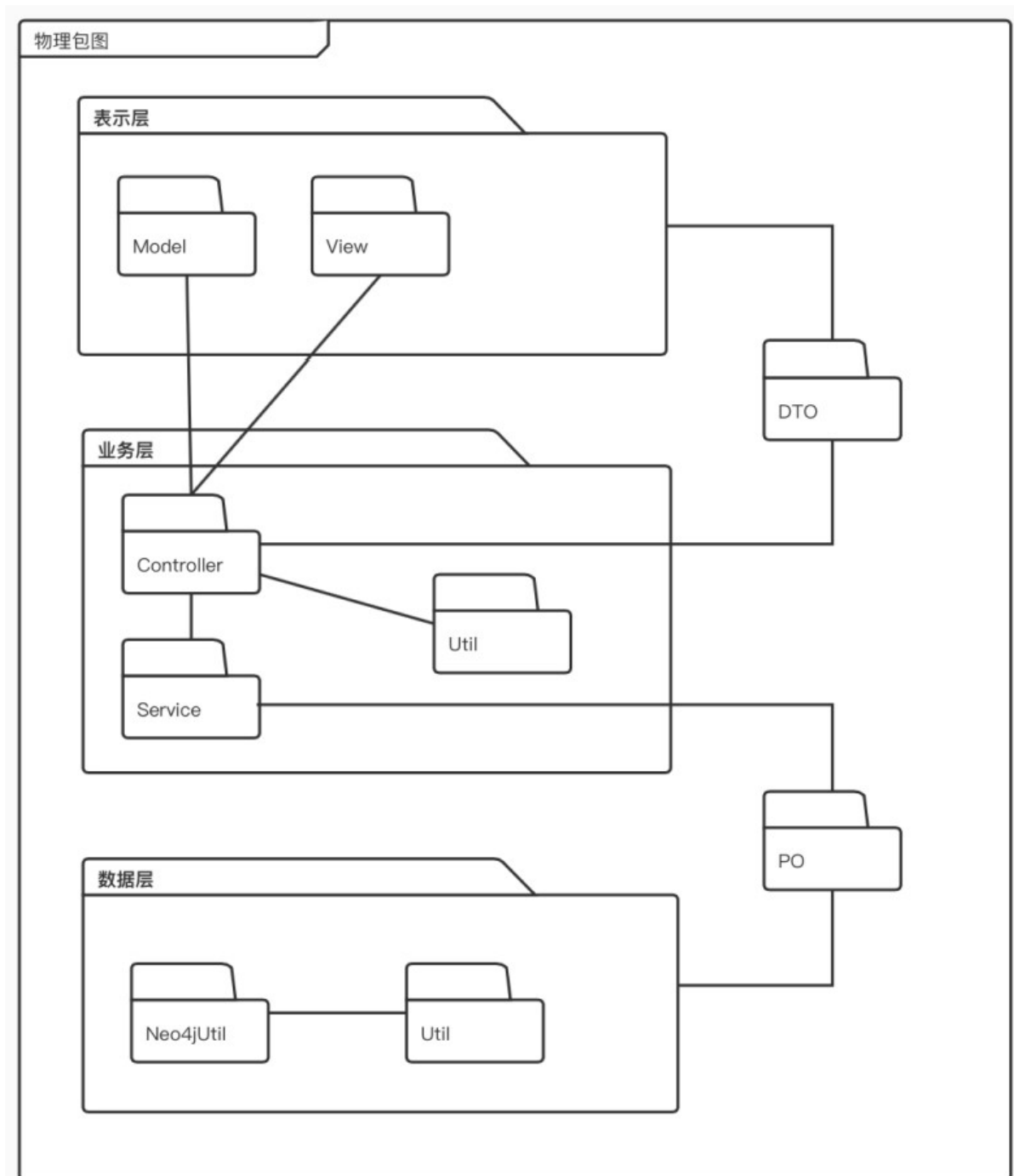


3. 组合视角

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 |
| GraphController | 图谱数据实体Restful接口 | 控制器层 |
| KafkaController | Kafka消息传输Restful接口 | 控制器层 |
| ModifyController | 修改图谱数据实体Restful接口 | 控制器层 |
| NREController | NRE传输文本数据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 |
| ModifyService | 解析文本获得的实体、关系的修改 | 业务层 |
| MailService | 发送邮件验证码 | 业务层 |
| NREService | 传输NRE解析结果 | 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 |
| setGraphStarredAPI(userId) | 收藏当前图谱 |
| sendPartTxtAPI(nreDTO) | 发送部分txt用于实体关系抽取 |
| sendFullTxtAPI(nreDTO) | 发送完整txt用于实体关系抽取 |
| getStatusAPI(graphId) | 获取图谱创建状态 |
| setThumbAPI(saveThumbDTO,image) | 设置图谱预览截图 |
| modifyNodesAPI(modifyNodeDTO) | 修改预处理的实体 |
| modifyRelationsAPI(modifyRelationDTO) | 修改预处理的关系 |

UserController

| 供接口 | 语法 | URL | 功能 |
|--------------------|--|---------------------------------|-----------------------------|
| getEmailVerCode | getEmailVerCode(@RequestParam String emailAddr) | @GetMapping("/getEmailCode") | 获取邮箱验证码 |
| verifyUserEmail | verifyUserEmail(@RequestBody verifyUserEmailDTO verifyUserEmailDTO) | @PostMapping("/verifyEmail") | 验证邮箱 |
| forgetPassword | forgetPassword(@RequestBody forgetPasswordDTO forgetPasswordDTO) | @PostMapping("/forgetPass") | 用户忘记密码，获取用户密码协助用户登陆 |
| register | register(@RequestBody userDTO userDTO) | @PostMapping("/register") | 用户注册 |
| login | login(@RequestBody userDTO userDTO) | @PostMapping("/login") | 用户登陆 |
| deleteGraph | deleteGraph(@RequestParam long kgId, @RequestParam long userId) | @GetMapping("/deleteGraph") | 删除图谱 |
| test | 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 |
| getNewGraphId | 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 |
| updateGraphName | updateGraphName(@RequestBody graphRenameDTO graphRenameDTO) | @PostMapping("/graphRename") | 对图谱进行重命名 |
| getAllGraph | 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 |
| getAllGraphByOrder | getAllGraphByOrder(@RequestParam long userId) | @GetMapping("/allGraphByOrder") | 根据userId按顺序获取当前用户的所有知识图谱 |

| 供接口 | 语法 | URL | 功能 |
|--------------------|---|---------------------------------|-----------------------------|
| getAllStarredGraph | getAllStarredGraph(@RequestParam long userId) | @GetMapping("/starredGraph") | 根据 userId 获取用户所有收藏的图谱 |
| setStarred | setStarred(@RequestParam long userId, @RequestParam long graphId, @RequestParam int starFlag) | @GetMapping("/setGraphStarred") | 用户收藏指定 graphId 的图谱 |
| checkAuth | checkAuth(@RequestParam long userId, @RequestParam long graphId) | @GetMapping("/checkAuth") | 检查用户权限 |
| changePassword | changePassword(@RequestBody passwordDTO passwordDTO) | @PostMapping("/changePass") | 用户修改当前密码 |
| changeUsername | changeUsername(@RequestBody usernameDTO usernameDTO) | @PostMapping("/changeUsername") | 用户修改当前用户名 |
| uploadTest | uploadTest(@RequestPart("data")) | @PostMapping("/uploadImg") | 用户上传头像 |

GraphController

| 接口名 | 接口 | URL | 功能 |
|--------------|---|-------------------------------|---|
| transJSON | transJSON(@RequestParam long kgId) | @GetMapping("/getGraph") | 根据 kgId返回所有的节点和关系信息 |
| getGraphName | getGraphName(@RequestParam long userId, @RequestParam long graphId) | @GetMapping("/getGraphName") | 根据 userId 和 graphID 返回当前图谱名 |
| updateGraph | updateGraph(@RequestBody updateGraphDTO updateGraphDTO) | @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 |
| setThumb | setThumb(@RequestPart("data") SaveThumbDTO saveThumbDTO, @RequestPart("image") MultipartFile image) | @PostMapping(value = "thumb") | 保存图谱的截图 |

Kafka Controller

| 接口名 | 语法 | URL | 功能 |
|--------------|------------------------|---|---------------------|
| kafkaSending | kafkaSending() | @GetMapping("/") | 通过Kafka向python端发送消息 |
| listenKafka | listenKafka(String in) | @KafkaListener(id = "ID", topics = "py2java") | 通过Kafka接收python端的结果 |

ModifyController

| 接口名 | 接口 | URL | 功能 |
|-----------------|--|----------------------------------|-----------|
| modifyNode | modifyNode(@RequestBody nodeDTO node) | @PostMapping("/node") | 修改单个实体的内容 |
| modifyNodes | modifyNodes (@RequestBody modifyNodeDTO modifyNodeDTO) | @PostMapping("/modifyNodes") | 批量修改实体的内容 |
| modifyRelation | modifyRelation(@RequestBody relationDTO relation) | @PostMapping("/relation") | 修改单个关系的内容 |
| modifyRelations | modifyRelations (@RequestBody modifyRelationDTO modifyRelationDTO) | @PostMapping("/modifyRelations") | 批量修改关系的内容 |

NREController

| 接口名 | 语法 | URL | 服务 |
|----------------|--|-------------------------|---------------------|
| nreText | nreText(@RequestBody nreDTO nreDTO) | @PostMapping("/first") | 向nre文本处理端第一次发送txt文本 |
| nreSecond | nreSecond(@RequestBody nreDTO nreDTO) | @PostMapping("/second") | 向nre文本处理端第二次发送txt文本 |
| getGraphStatus | getGraphStatus(@RequestParam long graphId) | @GetMapping("/status") | 获取图谱的状态 |

AccountService

| 接口名 | 语法 | 功能 |
|----------|----------------------------------|------|
| register | String register(userDTO userDto) | 用户注册 |
| login | User login(userDTO userDto) | 用户登录 |

GraphService

| 接口名 | 语法 | 功能 |
|--------------------|---|------------------------|
| parseJson | String parseJson(graphDTO data, long kgId) | 解析JSON数据 |
| GenerateNewGraph | long GenerateNewGraph(long userId, String graphName) | 生成新的图谱 |
| updateGraphName | String updateGraphName(graphInfoDTO newGraph) | 更新图谱名称 |
| getAllGraph | List getAllGraph(long userId) | 获取用户的全部图谱信息 |
| getGraphName | String getGraphName(long userId, long graphId) | 获取图谱名称 |
| setAttributeKeys | String setAttributeKeys(Map<String, String> attributes) | 设置图谱相关的属性（如节点id、节点名称等） |
| getAllGraph | List getAllGraph(long userId) | 获取指定用户所拥有的所有图谱 |
| getAllGraphByOrder | List getAllGraphByOrder(long userId) | 按顺序获取指定用户的所有图谱 |
| getStarredGraph | List getStarredGraph(long userId) | 获取指定用户的所有收藏的图谱 |
| setGraphStarred | String setGraphStarred(long userId, long graphId, int starFlag) | 用户收藏指定图谱 |
| deleteGraph | String deleteGraph(long userId, long graphId) | 删除图谱 |
| checkAuth | boolean checkAuth(long userId, long graphId) | 检查用户权限 |
| getGraphName | String getGraphName(long userId, long graphId) | 获取特定用户指定 graphID的图谱名称 |
| setGraphStatus | setGraphStatus(long graphId, GraphStatus graphStatus) | 设置指定图谱的状态 |
| getGraphStatus | String getGraphStatus(long graphId) | 获得指定图谱的状态 |
| setThumbPicture | setThumbPicture(SaveThumbDTO saveThumbDTO, InputStream stream) | 设置保存图谱的缩略截图 |

JsonTransService

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

mailService

| 接口名 | 语法 | 功能 |
|-----------------------|--|---------------|
| sendMail | void sendMail(String sender, JavaMailSenderImpl mailSender, String receiver) | 为正在注册的用户发送验证码 |
| verifyUserAndRegister | void verifyUserAndRegister(userDTO userDto) | 验证用户输入的验证码并注册 |

modifyService

| 接口名 | 语法 | 功能 |
|-----------------|---|--------|
| modifyNode | String modifyNode(nodeDTO node) | 修改单个节点 |
| modifyRelation | String modifyRelation(relationDTO relation) | 修改单条关系 |
| modifyNodes | String modifyNodes(modifyNodeDTO modifyNodeDTO) | 批量修改节点 |
| modifyRelations | String modifyRelations(modifyRelationDTO modifyRelationDTO) | 批量修改关系 |

NREService

| 接口名 | 语法 | 功能 |
|----------------|---|----------------------|
| pythonNRE | void pythonNRE(nreDTO nreDTO) | 向文本处理端发送txt |
| transformJson | void transformJson(String ereJson) | 将json数据转换为图谱所需要的数据格式 |
| createEntity | Entity createEntity(long graphId, String nodeLabel) | 根据获取的数据生成节点 |
| createRelation | Relation createRelation(long graphId, String node1Label, String node2Label, String relationLabel) | 根据获取到的数据生成关系 |

5. 信息视角

5.1 DTO的定义

node

| 字段 | 属性 | 含义 |
|----------------|---------------------|------|
| id | String | 节点id |
| label | String | 节点标签 |
| sysAttributes | Map<String, String> | 系统属性 |
| userAttributes | Map<String, String> | 用户属性 |

relation

| 字段 | 属性 | 含义 |
|----------------|---------------------|------|
| source | String | 源节点 |
| target | String | 目标节点 |
| label | String | 关系 |
| sysAttributes | Map<String, String> | 系统属性 |
| userAttributes | Map<String, String> | 用户属性 |

nodeDTO

| 字段 | 属性 | 含义 |
|------------|--------|------|
| actionType | int | 操作类型 |
| graphId | long | 图谱id |
| nodeId | long | 节点id |
| newLabel | String | 新的标签 |
| oldLabel | String | 旧的标签 |

relationDTO

| 字段 | 属性 | 含义 |
|------------|--------|-----------|
| actionType | int | 操作类型 |
| graphId | long | 图谱id |
| nodeId | long | 节点id |
| newLabel | String | 新的标签 |
| oldLabel | String | 旧的标签 |
| node1Label | String | 连接的节点1的标签 |
| node2Label | String | 连接的节点2的标签 |

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 |
| imgURL | String | 缩略图地址 |
| starred | String | 是否被收藏 |
| owned | String | 图谱的归属 |

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 |
| verCode | String | 验证码 |

forgetPasswordDTO

| 字段 | 属性 | 含义 |
|----------|--------|------|
| username | String | 用户名称 |
| password | String | 登录密码 |

passwordDTO

| 字段 | 属性 | 含义 |
|----------|--------|------|
| username | String | 用户名称 |
| oldPass | String | 旧密码 |
| newPass | String | 新密码 |

graphRenameDTO

| 字段 | 属性 | 含义 |
|--------------|--------|------|
| graphId | long | 图谱id |
| graphNewName | String | 新图谱名 |

modifyNodeDTO

| 字段 | 属性 | 含义 |
|---------|------|-----------|
| data | List | 需要修改的节点列表 |
| graphId | long | 图谱id |

modifyRelationDTO

| 字段 | 属性 | 含义 |
|---------|------|-----------|
| data | List | 需要修改的关系列表 |
| graphId | long | 图谱id |

saveThumbDTO

| 字段 | 属性 | 含义 |
|---------|--------|------|
| name | String | 图谱名 |
| graphId | long | 图谱id |

updateGraphDTO

| 字段 | 属性 | 含义 |
|-------|------|------|
| nodes | List | 节点列表 |
| edges | List | 关系列表 |
| kgId | long | 图谱id |

nreDTO

| 字段 | 属性 | 含义 |
|---------|--------|---------|
| text | String | 需要传输的文本 |
| graphId | long | 图谱id |

5.2 PO的定义

User

| 字段 | 属性 | 含义 |
|----------|--------|------|
| id | int | 用户id |
| email | String | 用户邮箱 |
| password | String | 用户密码 |
| username | String | 用户名 |
| imgUrl | String | 用户头像 |

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 |