

# 项目设计文档

团队名称: ipoooo6

更新记录

日期	作者	变更说明	版本号
2021.03.10	吴雨晴	添加1-4.22内容	v1.0
2021.03.10	张嘉玥	添加4.2.3-5内容	v1.1
2021.03.11	吴雨晴	添加物理包图	v1.2
2021.03.16	冯泊涓	修改接口设计和架构	v1.3
2021.03.20	冯泊涓	修改接口设计和架构	v1.4
2021.03.28	冯泊涓	迭代二更新	v2.0
2021.04.18	冯泊涓	迭代二终版	v2.1
2021.06.17	冯泊涓	迭代三终版	v3.0

## 项目设计文档

### 1.引言

#### 1.1编制目的

#### 1.2词汇表

#### 1.3参考资料

### 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

##### 4.2.1用户界面模块的分解

##### 4.2.2业务逻辑模块的分解

##### 4.2.3数据模块的分解

### 5.信息视角

#### 5.1 VO定义

#### 5.2 PO定义

# 1.引言

---

## 1.1编制目的

本文档描述了COIN知识图谱定义及可视化系统的概要设计，说明详细设计和开发的目的，同时实现和测试人员及用户的沟通。开发小组的软件系统实现与验证工作都以此文档为依据。

## 1.2词汇表

词汇名称	词汇含义	备注
COIN	知识图谱定义及可视化系统	

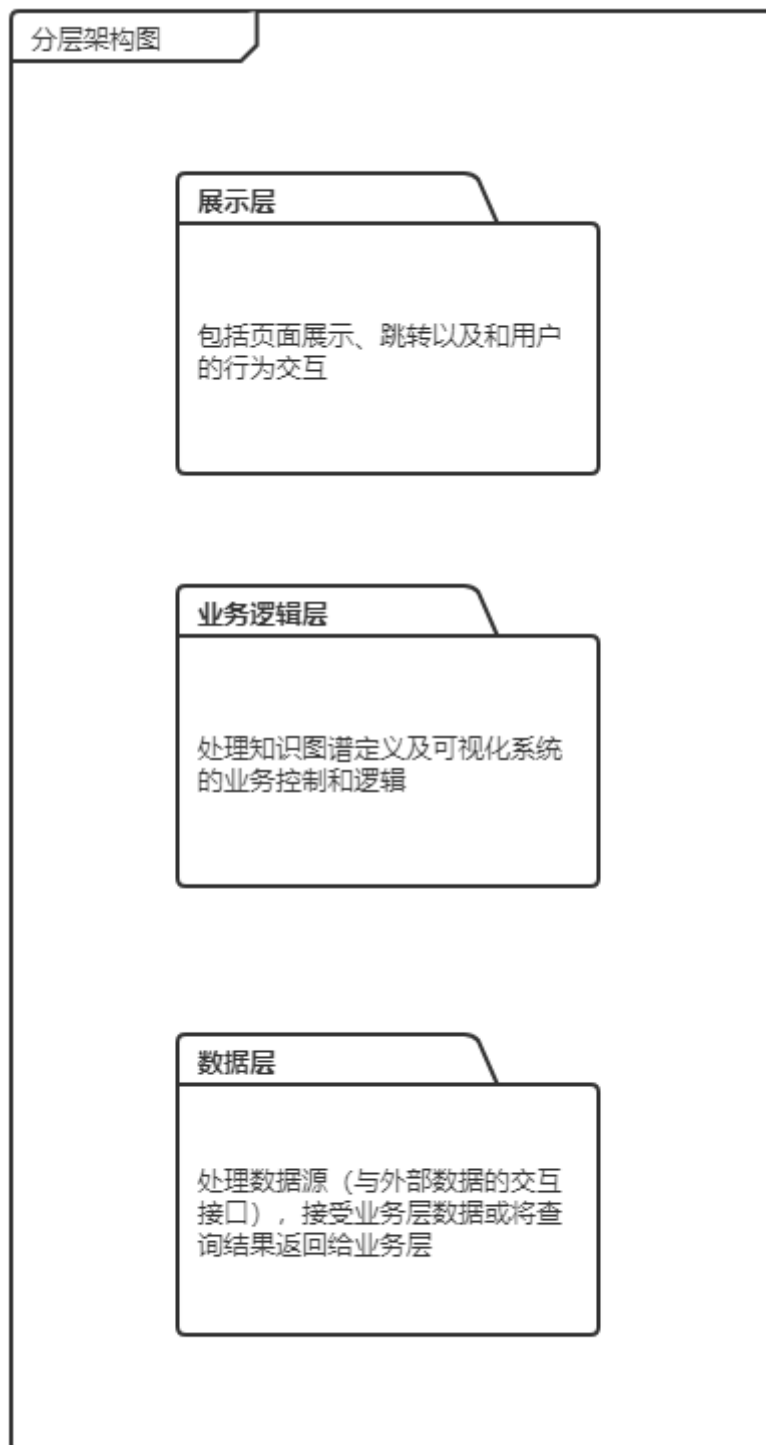
## 1.3参考资料

- 1. 软件工程与计算(卷二)——软件开发的技术基础/骆斌主编.-北京：机械工业出版社,2012.12 （2018.4重印）
- 2. COIN需求规格说明文档v1.0

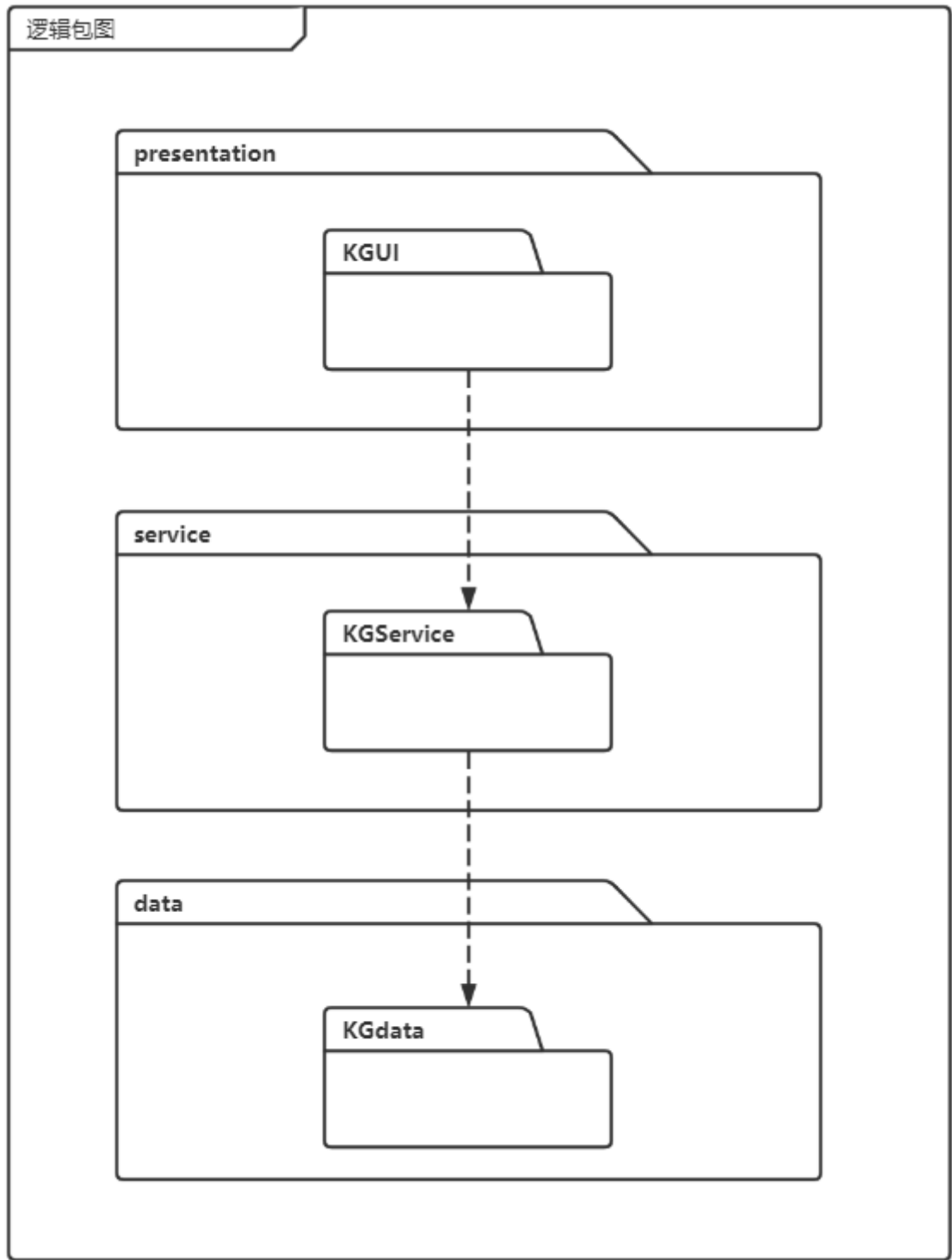
# 2.逻辑视角

---

## 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（物理）包	依赖的其他开发包
KGController	VO, KGService
KGService	VO, PO, KGData, util
KGData	util, PO
VO	
PO	
util	
conf	util
view	

### 3.2物理包图



需要的服务（需接口）	
服务名	服务
graphService.getRelativeRelation	根据节点id，获得对应关系
graphService.getByLabel	根据节点label，获得该类型节点列表
graphService.getDistinctLabels	获得数据库中所有不同的节点类型列表，及其统计数
graphService.getDistinctRelationNames	获得数据库中所有不同的关系类型列表，及其统计数
graphService.getDistinctPropertyValues	根据节点label和属性名，获得数据库中某一类型节点某一属性的所有不同取值
graphService.getTree	根据节点id，获得指定深度树结构下级图谱
userAccountService.login	用户根据手机号、密码登录
userAccountService.register	用户根据用户名、手机号、密码注册账号
userAccountService.modifyFavor	用户修改个人偏好
userAccountService.modifyPassword	用户修改密码
userCollectionService.collect	用户收藏图谱节点
userCollectionService.getCollection	根据用户id获得用户收藏列表
intelligenceService.search	用户智能推荐（搜索）图谱节点
intelligenceService.answer	智能问答

模块	职责
KGUI-admin	管理员界面中涉及的页面显示、跳转、业务逻辑

需要的服务（需接口）	
服务名	服务
graphService.addEntity	向图谱添加节点
graphService.addRelation	向图谱添加关系
graphService.updateRelationName	根据关系id，修改已有的关系名
graphService.deleteEntity	根据id，删除节点及其相关的关系
graphService.deleteRelation	根据id，删除关系
graphService.updateEntityProperty	根据id，更新节点的属性
graphService.updateRelationProperty	根据id，更新关系的属性
userAccountService.login	用户根据手机号、密码登录
intelligenceService.search	用户智能推荐（搜索）图谱节点

#### 4.2.2业务逻辑模块的分解

模块	职责
graphService	知识图谱增删改查相关逻辑



graphService.getRelativeRelation	语 法	ResponseVO getRelativeRelation(long id)
	前 置 条 件	节点id有效
	后 置 条 件	返回id对应节点的相关关系列表, [{relation1},{relation2}]
graphService.addEntity	语 法	public ResponseVO addEntity(List< String> labels, Map< String, Object> properties);
	前 置 条 件	Cookie验证用户为管理员
	后 置 条 件	返回新添加的节点id
graphService.addRelation	语 法	public ResponseVO addRelation(String name, long source, long target, Map< String, Object> properties);
	前 置 条 件	Cookie验证用户为管理员, source,target为合法节点, name不为空
	后 置 条 件	返回新添加的关系id

graphService.updateRelationName	语法	public ResponseVO updateRelationName(long id, String name);
	前置 条件	Cookie验证用户为管理员，id 合法
	后置 条件	将该id关系改名，并返回更新的 关系id
graphService.deleEntity	语法	public ResponseVO deleteEntity(EntityVO entityVO);
	前置 条件	Cookie验证用户为管理员， entityVO中id有效
	后置 条件	将实体和该实体的所有关系删 除
graphService.deleRelation	语法	public ResponseVO deleteRelation(RelationVO relationVO);
	前置 条件	Cookie验证用户为管理员， relationVO中id有效
	后置 条件	将该id关系删除
graphService.getByLabels	语法	public ResponseVO getByLabels(List< String> labels);

	前置条件	labels不为空
	后置条件	返回数据库中符合label的所有节点及其关系
graphService.updateEntityProperty	语法	public ResponseVO updateEntityProperty(long id, Map< String, Object> properties);
	前置条件	Cookie验证用户为管理员, id合法, properties不为空
	后置条件	更新数据库中对应实体的属性
graphService.updateRelationProperty	语法	public ResponseVO updateRelationProperty(long id, Map< String, Object> properties);
	前置条件	Cookie验证用户为管理员, id合法, properties不为空
	后置条件	更新数据库中对应关系的属性
graphService.getDistinctPropValues	语法	public ResponseVO getDistinctPropValues(String label,String propName);

	前置条件	label和属性名合法
	后置条件	返回数据库中该label节点该属性所有不同取值及其统计数
graphService.getDistinctLabels	语法	public ResponseVO getDistinctLabels();
	前置条件	无
	后置条件	返回数据库中所有不同label列表及其统计数
graphService.getDistinctRelationNames	语法	public ResponseVO getDistinctRelationNames();
	前置条件	无
	后置条件	返回数据库中所有不同关系名列表及其统计数
graphService.getTree	语法	public ResponseVO getTree(long id, int depth);
	前置条件	实体id有效

	后置条件	返回该实体为树根，深度为depth的关系树，depth默认为3
intelligenceService.search	语法	public ResponseVO search(String key, String label, Map< String, List< String >> props);
	前置条件	key,label,props可以为空， props必须与label对应
	后置条件	返回智能搜索结果
intelligenceService.answer	语法	public ResponseVO answer(String question);
	前置条件	question不为空
	后置条件	返回智能问答结果

模块	职责
userService	用户账号管理

userAccountService.login	语法	public ResponseVO login(String phone, String password);
	前置条件	phone为11位有效注册手机号, password合法
	后置条件	登陆成功, 添加浏览器Cookie, 返回用户信息
userAccountService.register	语法	public ResponseVO login(String username, String phone, String password);
	前置条件	username不为空, phone为11位有效注册手机号且未被注册, password合法
	后置条件	注册用户账号并加密存储密码
userAccountService.modifyFavor	语法	public ResponseVO modifyFavor(long id, String flavor, String method, String type);
	前置条件	用户已登录, 用户id合法, 用户偏好字段合法
	后置条件	修改用户偏好信息

userAccountService.modifyPassword	语法	public ResponseVO modifyPassword(long id, String originPassword, String newPassword);
	前置条件	用户已登录，用户id合法，输入的原密码正确
	后置条件	修改用户密码
userCollectionService.collect	语法	public ResponseVO collect(long id, long entityId, String recipeName);
	前置条件	用户已登录，用户id合法，输入的节点id和名字合法
	后置条件	收藏节点，增加节点热度
userCollectionService.getCollections	语法	public ResponseVO getCollections(long id);
	前置条件	用户已登录，用户id合法
	后置条件	获得用户收藏节点信息

需要的服务（需接口）	
服务名	服务
userMapper.getUserById	根据用户id检索用户信息
userMapper.getUserByPhone	根据注册手机号检索用户信息
userMapper.register	注册账号
userMapper.modifyPassword	修改密码
userMapper.modifyFavor	修改用户偏好
collectionMapper.collect	收藏节点
collectionMapper.getCollection	获得用户收藏节点列表

#### 4.2.3数据模块的分解

模块	职责
neo4jData	负责知识图谱数据的增删改查



neo4jData.getRelativeRelation	语法	public List< Map< String, Object>> getAllRelativeRelation(long id);
	前置条件	实体id合法
	后置条件	返回对应实体所有相关关系
neo4jData.insertEntity	语法	public long insertEntity(List< String> labels, Map< String, Object> properties);
	前置条件	labels和properties字段合法
	后置条件	添加实体并返回自动生成的实体id
neo4jData.insertRelation	语法	public long insertRelation(String name, long source, long target, Map< String, Object> properties);
	前置条件	source, target, name合法且不为空
	后置条件	添加从source到target名为name的实体关系并返回关系id
neo4jData.searchEntityById	语法	public Map< String, Object> searchEntityById(long id);
	前置条件	id合法
	后置条件	根据id查询实体并返回
neo4jData.updateRelationName	语法	public void updateRelationName(long id,String newName);

	前置条件	id,newName均合法
	后置条件	修改该id关系名称为newName并返回新的关系id
neo4jData.delEntity	语法	public void delEntity(long id);
	前置条件	id合法
	后置条件	删除id对应的实体
neo4jData.delRelation	语法	public void delRelation(long id);
	前置条件	id合法
	后置条件	删除该id关系
neo4jData.getByLabels	语法	public List< Map< String, Object>> getByLabels(List< String> labels);
	前置条件	labels合法
	后置条件	根据labels查询实体，返回实体列表
neo4jData.updateEntityProperty	语法	public Map< String,Object> updateEntityProperty(long id,Map< String,Object> properties);
	前置条件	properties合法
	后置条件	更新对应实体的属性
neo4jData.updateRelationProperty	语法	public Map< String,Object> updateRelationProperty(long id,Map< String,Object> properties);

	前置条件	properties合法
	后置条件	更新对应关系的属性
neo4jData.getDistinctLabels	语法	public List< Map< String,Object> > getDistinctLabels();
	前置条件	无
	后置条件	返回所有label:count
neo4jData.getDistinctRelationNames	语法	public List< Map< String,Object> > getDistinctRelationNames();
	前置条件	无
	后置条件	返回所有name:count
neo4jData.getDistinctPropValues	语法	public List< Map< String,Object> > getDistinctPropValues(String label,String propName);
	label和属性名合法	无
	后置条件	返回所有name:count
neo4jData.intelligenceSearch	语法	public List< Map< String,Object>> intelligenceSearch(String key, String filter);
	前置条件	key和filter合法

	后置 条件	返回符合搜索条件的实体列表
--	----------	---------------

模块	职责
mysqlData	负责用户信息数据相关的增删改查

userMapper.getUserById	语法	User getUserById(long id);
	前置条件	id合法
	后置条件	返回对应id的用户信息
userMapper.getUserByPhone	语法	User getUserByPhone(String phone);
	前置条件	phone合法
	后置条件	返回对应注册手机号的个人信息
userMapper.register	语法	void register(String username, String phone, String password);
	前置条件	用户名、手机号、密码合法不为空
	后置条件	注册用户账户
userMapper.modifyPassword	语法	void modifyPassword(long id, String password);
	前置条件	用户已登录且正确输入原密码

	后置条件	修改密码
userMapper.modifyFavor	语法	void modifyPassword(long id, String flavor, String method, String type);
	前置条件	用户已登录,偏好信息合法
	后置条件	修改用户偏好

## 5.信息视角

### 5.1 VO定义

ResponseVO

含义	属性	字段
调用成功标识符	boolean	success
提示信息	String	message
内容	Object	content

EntityVO

含义	属性	字段
实体id	Long	id
实体属性中的名称，用于搜索实体	String	name
实体属性中的名称，用于实体更名	String	newName
实体标签	List	labels
实体属性	Map<String,Object>	properties

## RelationVO

含义	属性	字段
关系id	Long	id
关系名	String	name
新关系名	String	newName
关系的起始实体id	Long	source
关系的终点实体id	Long	target
关系属性	Map<String,Object>	properties

## TreeVO

含义	属性	字段
本树节点id	Long	id
本树节点名称	String	name
本树节点子节点	ArrayList	children
上级节点到本节点的关系名	String	relation
本节点距离root的深度	Integer	depth

## UserVO

含义	属性	字段
用户id	Long	id
用户类型	Integer	userType
用户密码	String	password
新密码	String	newPassword
用户名	String	username
手机号	String	phoneNumber
菜谱口味偏好	List	flavor
菜谱工艺偏好	List	method
菜谱类型偏好	List	type

## 5.2 PO定义

### Entity

含义	属性	字段
实体id	Long	id
实体标签	List	labels
实体属性	Map<String,Object>	properties

### Relation

含义	属性	字段
关系id	Long	id
关系名	String	name
关系的起始实体id	Long	source
关系的终点实体id	Long	target
关系属性	Map<String,Object>	properties

### User



含义	属性	字段
用户id	Long	id
用户类型	Integer	userType
用户密码	String	password
用户名	String	username
手机号	String	phoneNumber
菜谱口味偏好	String	flavor
菜谱工艺偏好	LString	method
菜谱类型偏好	String	type