

智能家用植物养护IoT平台

目录

智能家用植物养护IoT平台

- 目录
- 变更记录
- 1 引言
 - 1.1 编制目的
- 2 产品描述
- 3 逻辑视角
- 4 组合视角
- 4.1 开发包图
- 4.2 运行时进程
- 5 接口视角
 - 5.1 controller层模块职责划分
 - 5.2 service层的分解
 - 5.2.1 service层模块职责划分
 - 5.2.1.2 service模块的接口规范
 - 5.3 dao层的分解
 - 5.3.1 dao层模块的职责
 - 5.3.2 dao层模块的接口规范
- 6 信息视角
 - 6.1 数据持久化对象

变更记录

修改人员	日期	变更原因	版本号
许杨	2021.4.14	最初草稿	v1.0
许杨	2021.4.18	更新数据库设计及设备管理部分内容	v2.0
许杨、陈迪新	2021.4.21	更新规则引擎模块，物模型数据库设计及相关逻辑实现设计	v3.0

1 引言

1.1 编制目的

本报告介绍了智能家用植物养护IoT平台的设计信息，达到指导后续软件构造的目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

2 产品描述

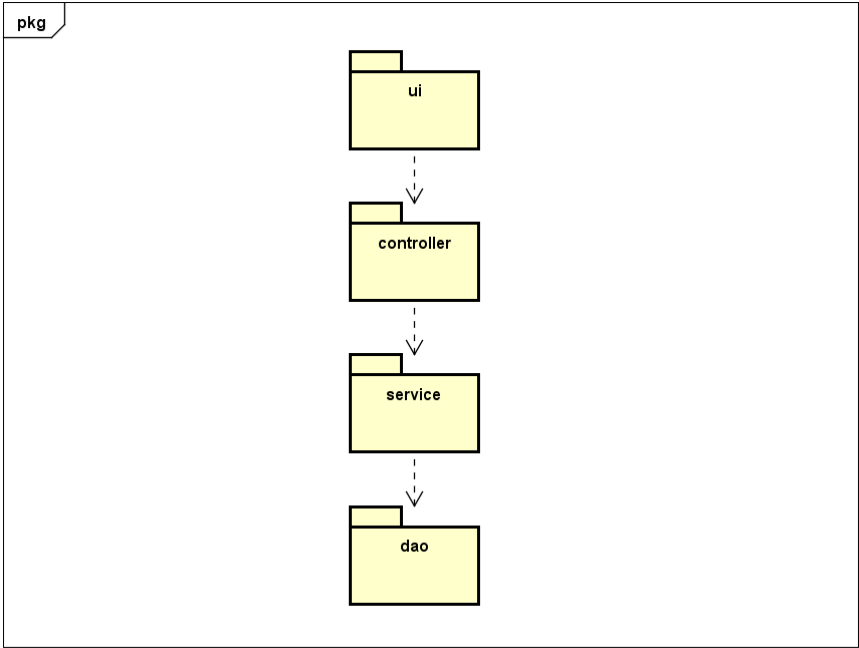
随着全国城市化的不断发展，家庭养生的观念深入人心，家庭种菜种花的现象变得十分常见，然而常见的家用种植方式往往会因为疏于管理和养护，导致植被的不良生长甚至死亡。

针对这样的情况我们打算设计一款可辅助植物生长的基于物联网云技术的家用植物养护应用，通过连接传感器，远程监控家用绿植的生长情况，并进行远程遥控，以解决以上存在的问题。

3 逻辑视角

植物养护IoT平台中使用了分层的体系结构风格，系统分为四层（ui层，controller层，service层，dao层），能够很好地示意整个高层抽象。ui层包含web界面的实现，controller层负责web请求的转发，service层包含业务逻辑处理的实现，dao层负责提供数据库访问和数据持久化。

分层体系结构的逻辑视角设计方案如下面的图所示。



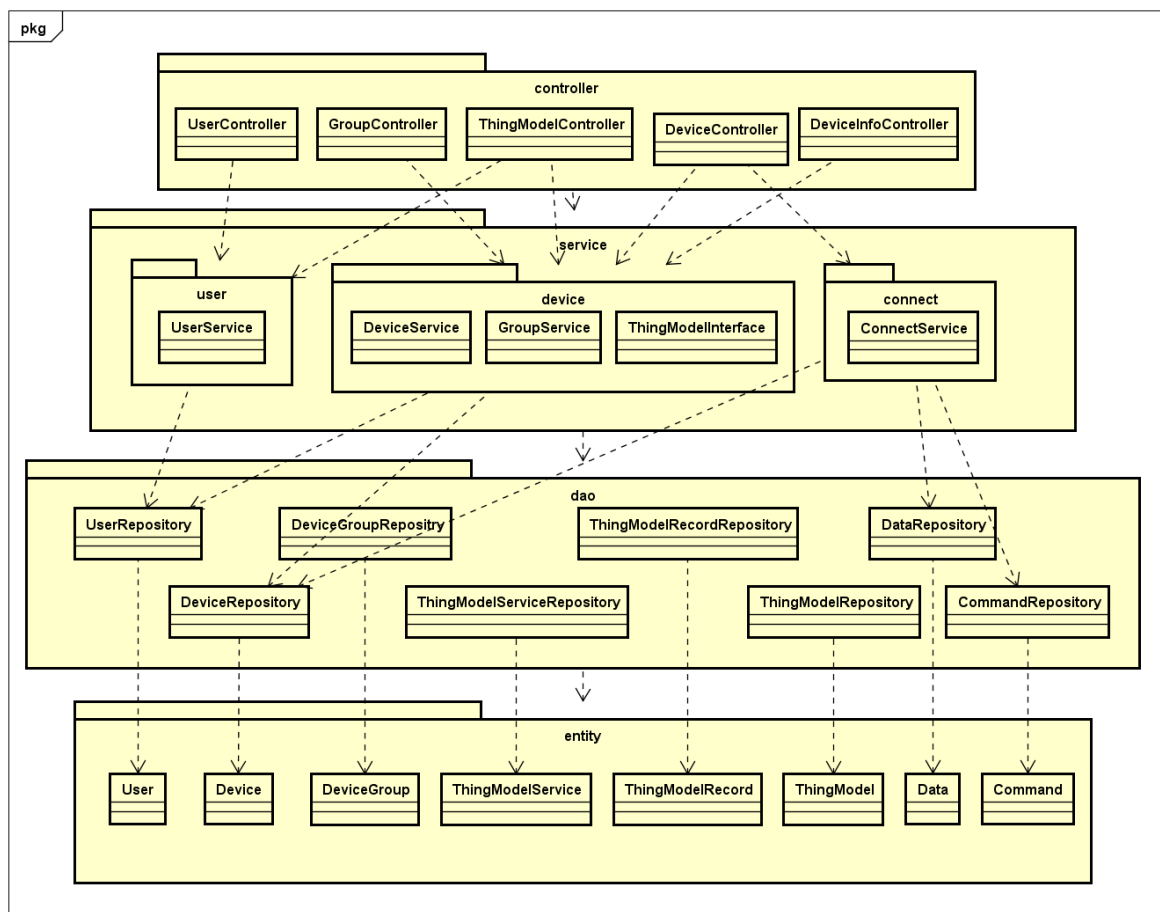
powered by Astah

图3-1 参照体系结构风格的包图表达逻辑视角

4 组合视角

4.1 开发包图

植物养护IoT平台客户端开发使用了vue框架，服务端开发包图如图 4-1-1 所示。

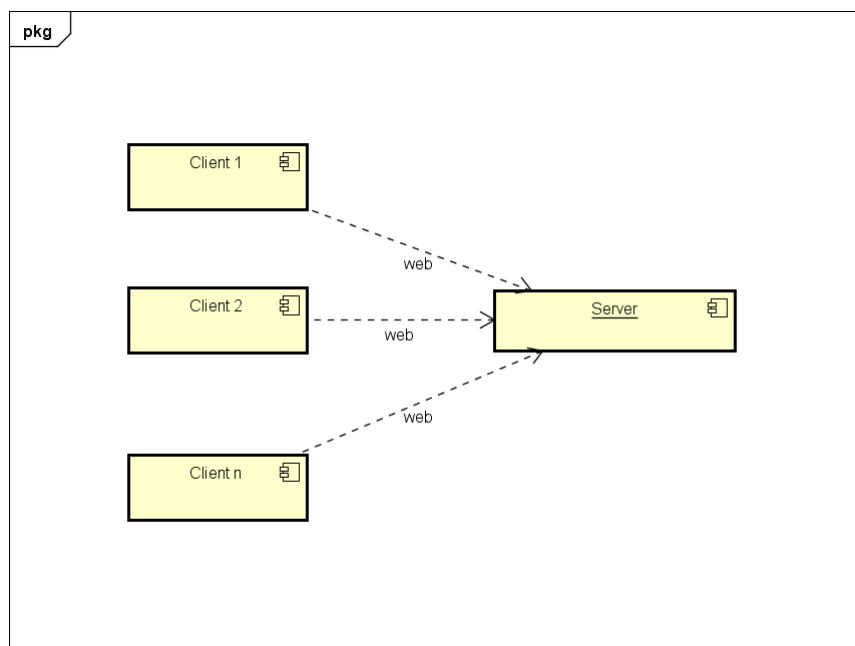


powered by Astah

图4-1-1 植物养护IoT平台服务端开发包图

4.2 运行时进程

在植物养护IoT平台中，会有多个客户端进程和一个服务器端进程，其进程图如图 4-2-1 所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程在服务器端机器上运行。



powered by Astah

图4-2-1 进程图

5 接口视角

5.1 controller层模块职责划分

植物养护IoT平台的controller层主要负责转发web页面请求，根据功能模块划分为DeviceController、DeviceInfoController、GroupController、ThingModelController、UserController等。

表 5-1-1 controller层模块的职责

Controller	职责
DeviceController	设备管理
DeviceInfoController	设备信息的展示
GroupController	设备组的管理、信息展示
ThingModelController	物模型的管理、信息展示
UserController	用户管理
DataController	传感器数据展示

5.2 service层的分解

service层包含多个业务逻辑处理对象，根据功能划分为若干模块。

5.2.1 service层模块职责划分

植物养护IoT平台的service层主要负责业务逻辑处理，根据功能主要划分为5个模块：连接管理、设备管理、数据分析、规则引擎、用户管理。

service层各模块的职责如表 5-2-1-1 所示。

表 5-2-1-1 service层模块的职责

模块	职责
连接管理	负责平台和终端设备之间的实时通信，支持终端设备的接入、数据上报，远程命令交互等功能
设备管理	设备的增加和删除，管理设备的状态，对设备进行分组管理，以及采集设备的日志信息等功能
规则引擎	负责检测需要监控的数据，如温度、湿度等，与规则匹配，当数据异常时发出告警并控制设备采取可能的措施，同时支持用户创建、修改规则设置
数据分析	
用户管理	负责用户的登入、登出

5.2.1.2 service模块的接口规范

表 5-2-1-2 连接管理模块的接口规范

接口		
Connect.sendCommand	语法	String sendCommand(Long deviceId, String[] commands, Double[] values);
	前置条件	设备存在，命令合法
	后置条件	向设备下发命令
Connect.sendCommand	语法	Map<Long, String> sendCommand(Long[] deviceIdList, String[] commands, Double[] values);
	前置条件	设备存在，命令合法
	后置条件	向设备下发命令
Connect.getData	语法	List<DataRecord> getData(long userId);
	前置条件	用户存在
	后置条件	展示传感器数据
Connect.getData	语法	List<DataRecord> getData(long userId, String dataType, String start, String end);
	前置条件	用户存在
	后置条件	展示传感器数据

表 5-2-1-3 设备管理模块的接口规范

接口		
Device.addDevice	语法	long addDevice(Long userId, DeviceType type, ThingModelRequest modelRequest);
	前置条件	用户存在
	后置条件	创建设备
Device.deleteDevice	语法	void deleteDevice(Long deviceId);
	前置条件	设备存在
	后置条件	删除设备
Device.disableDevice	语法	void disableDevice(Long deviceId);
	前置条件	设备存在
	后置条件	禁用设备
Device.enableDevice	语法	boolean enableDevice(Long deviceId);
	前置条件	设备存在
	后置条件	启用设备
Device.getDeviceInfo	语法	DeviceInfo getDeviceInfo(long deviceId);

接口		
	前置条件	设备存在
	后置条件	查看设备信息
Device.getDeviceInfoList	语法	List<DeviceInfo> getDeviceInfoList(long userId);
	前置条件	用户存在
	后置条件	查看用户的设备信息列表
Device.getDeviceLog	语法	DeviceLog getDeviceLog(long deviceId);
	前置条件	设备存在
	后置条件	查看设备日志
Device.getDeviceLogList	语法	List<DeviceLog> getDeviceLogList(long userId);
	前置条件	用户存在
	后置条件	查看设备日志
Device.addGroup	语法	boolean addGroup(Long userId, String name, Long[] deviceIdList);

接口		
	前置条件	用户存在
	后置条件	设置设备群组
Device.persistThingModel	语法	ThingModelInfo persistThingModel(Long userId, ThingModelRequest request);
	前置条件	用户存在
	后置条件	创建物模型
Device.getThingModel	语法	List<ThingModelInfo> getThingModel(Long userId);
	前置条件	用户存在
	后置条件	查看物模型

表 5-2-1-4 用户管理模块的接口规范

接口		
User.login	语法	public LoginResult login(Long userId, String password);
	前置条件	无
	后置条件	用户登入
User.logout	语法	public void logout(Long userId);
	前置条件	无
	后置条件	用户登出

表 5-2-1-5 规则引擎模块的接口规范

接口		
Rule.addRule	语法	public Long addRule(RuleType type, Long user_id, Double data, RuleAction rule_state;
	前置条件	无
	后置条件	返回生成的规则id
Rule.deleteRule	语法	public void deleteRule(Long id);
	前置条件	无
	后置条件	删除某条规则
Rule.enableRule	语法	public void enableRule(Long id);
	前置条件	无
	后置条件	启用某条规则
Rule.disableRule	语法	public void disableRule(Long id);
	前置条件	无
	后置条件	禁用某条规则
Rule.showAllRulesByUserId	语法	public ArrayList showAllRulesByUserId(Long id);

接口		
	前置条件	无
	后置条件	根据用户id，查找该用户建立的所有规则
Rule.showTempRules	语法	public ArrayList showTempRules(Long id);
	前置条件	无
	后置条件	根据用户id，查找该用户建立的所有温度规则
Rule.showHumidityRules	语法	public ArrayList showHumidityRules(Long id);
	前置条件	无
	后置条件	根据用户id，查找该用户建立的所有湿度规则
Rule.showCO2Rules	语法	public ArrayList showCO2Rules(Long id);
	前置条件	无
	后置条件	根据用户id，查找该用户建立的所有CO2含量相关规则
Rule.showLIRules	语法	public ArrayList showLIRules(Long id);

接口		
	前置条件	无
	后置条件	根据用户id，查找该用户建立的所有光照强度相关规则
RE_Temperature.checkTempThreshold	语法	public boolean checkTempThreshold(Long userId, Double temp_data);
	前置条件	无
	后置条件	返回温度规则的阈值检测和比较结果
RE_Temperature.sendTempWarningSignal	语法	public boolean sendTempWarningSignal(Long userId, int signal);
	前置条件	无
	后置条件	温度告警
RE_Humidity.checkHumidityThreshold	语法	public boolean checkHumidityThreshold(Long userId, Double temp_data);
	前置条件	无
	后置条件	返回湿度规则的阈值检测和比较结果
RE_Temperature.sendHumidityWarningSignal	语法	public boolean sendHumidityWarningSignal(Long userId, int signal);

接口		
	前置条件	无
(其它类似)	后置条件	湿度告警

5.3 dao层的分解

dao层主要给service层提供数据访问服务，包括对于持久化数据的增、删、改、查。各业务逻辑需要的服务由各repository接口提供。持久化数据的保存采用MySQL进行保存。

5.3.1 dao层模块的职责

dao层模块的职责如表 5-3-1 所示。

表 5-3-1 dao层模块的职责

模块	职责
UserRepository	基于JpaRepository提供对User的增、删、改、查服务
DeviceRepository	基于JpaRepository提供对Device的增、删、改、查服务
DataRepository	基于JpaRepository提供对Data的增、删、改、查服务
CommandRepository	基于JpaRepository提供对Command的增、删、改、查服务
ThingModelRepository	基于JpaRepository提供对ThingModel的增、删、改、查服务
ThingModelServiceRepository	基于JpaRepository提供对ThingModelService的增、删、改、查服务
ThingModelRecordRepository	基于JpaRepository提供对ThingModelRecord的增、删、改、查服务
DeviceGroupRelationRepository	基于JpaRepository提供对DeviceGroupRelation的增、删、改、查服务
DeviceOnOffRecordRepository	基于JpaRepository提供对DeviceOnOff的增、删、改、查服务
DeviceGroupRepository	基于JpaRepository提供对DeviceGroup的增、删、改、查服务
UserThingModelRelationRepository	基于JpaRepository提供对UserThingModelRelation的增、删、改、查服务
RuleRepository	基于JpaRepository提供对Rule的增、删、改、查服务

5.3.2 dao层模块的接口规范

dao层模块的接口规范如JpaRepository规范。

6 信息视角

6.1 数据持久化对象

系统的主要Entity类如下所示：

- User
 - 用于记录平台的用户信息
- | 列 | 描述 | 类型 | 备注 |
|----------|------|--------|------|
| id | 用户id | long | 自增主键 |
| password | 用户密码 | string | |
- Device
 - 用于记录平台连接的设备信息
- | 列 | 描述 | 类型 | 备注 |
|---------------|------------|---------|-----------------|
| device_id | 设备id | long | 自增主键 |
| user_id | 对应用户id | long | 外键，连接User |
| type | 设备类型 | string | |
| topic | 设备对应MQTT主题 | string | |
| register_time | 注册时间 | time | |
| state | 设备状态 | string | |
| is_online | 设备是否在线 | boolean | |
| thing_model | 物模型 | int | 外键，连接ThingModel |
- DeviceGroup
 - 用于记录设备群组信息
- | 列 | 描述 | 类型 | 备注 |
|-----------|--------|--------|-----------|
| id | id | long | 自增主键 |
| user_id | 对应用户id | long | 外键，连接User |
| groupName | 群组名称 | string | |
- DeviceGroupRelation
 - 用于记录设备群组与设备的对应信息

- | 列 | 描述 | 类型 | 备注 |
|-----------|------|------|------------------|
| id | id | long | 自增主键 |
| device_id | 设备id | long | 外键，连接Device |
| group_id | 群组id | long | 外键，连接DeviceGroup |
- DeviceOnOffRecord
 - 用于记录设备上线、下线信息
 - | 列 | 描述 | 类型 | 备注 |
|-----------|-------|--------|-------------|
| id | id | long | 自增主键 |
| device_id | 设备id | long | 外键，连接Device |
| time | 发生时间 | time | |
| action | 上下线动作 | string | |
- Data
 - 用于记录设备上报的数据
 - | 列 | 描述 | 类型 | 备注 |
|-----------|--------|--------|--------------------------|
| id | 数据id | string | 主键
格式: time-device_id |
| time | 数据上报时间 | string | |
| device_id | 设备id | long | 外键，连接Device |
| type | 数据类型 | string | |
| value | 数值 | double | |
- Command
 - 用于记录向设备下发的命令
 - | 列 | 描述 | 类型 | 备注 |
|-----------|--------|--------|--------------------------|
| id | 数据id | string | 主键
格式: time-device_id |
| time | 数据上报时间 | string | |
| device_id | 设备id | long | 外键，连接Device |
| type | 命令类型 | string | |
| command | 命令内容 | string | |
- ThingModel
 - 物模型，用来定义设备的服务

- | 列 | 描述 | 类型 | 备注 |
|-------------|-------|--------|------|
| model_id | 物模型id | int | 自增主键 |
| model_name | 物模型名称 | string | |
| device_type | 设备类型 | string | |
- ThingModelService
 - 记录物模型的服务
 - | 列 | 描述 | 类型 | 备注 |
|--------------|----|--------|----|
| service_name | | string | 主键 |
| type | | int | |
| description | 描述 | string | |
| unit | 单位 | string | |
| quantity | | double | |
- ThingModelRecord
 - 记录ThingModelService和物模型的对应关系
 - | 列 | 描述 | 类型 | 备注 |
|--------------|----|--------|------------------------|
| record_id | id | long | 自增主键 |
| thing_model | | int | 外键，连接ThingModel |
| service_name | | string | 外键，连接ThingModelService |
- UserThingModelRelation
 - 记录用户创建的物模型的对应关系
 - | 列 | 描述 | 类型 | 备注 |
|-------------|--------|------|-----------------|
| id | id | long | 自增主键 |
| user_id | 对应用户id | long | 外键，连接User |
| thing_model | 物模型id | int | 外键，连接ThingModel |
- Rule
 - 用于记录规则数据
 - | 列 | 描述 | 类型 | 备注 |
|----------------|------|--------|-----------|
| id | 用户id | long | 自增主键 |
| type | 用户密码 | string | |
| user_id | 用户id | long | 外键，连接user |
| threshold_data | 阈值 | double | |
| state | 状态 | String | 自定义类型 |