

# 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

## 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和逻辑设计方案如下面的图所示。

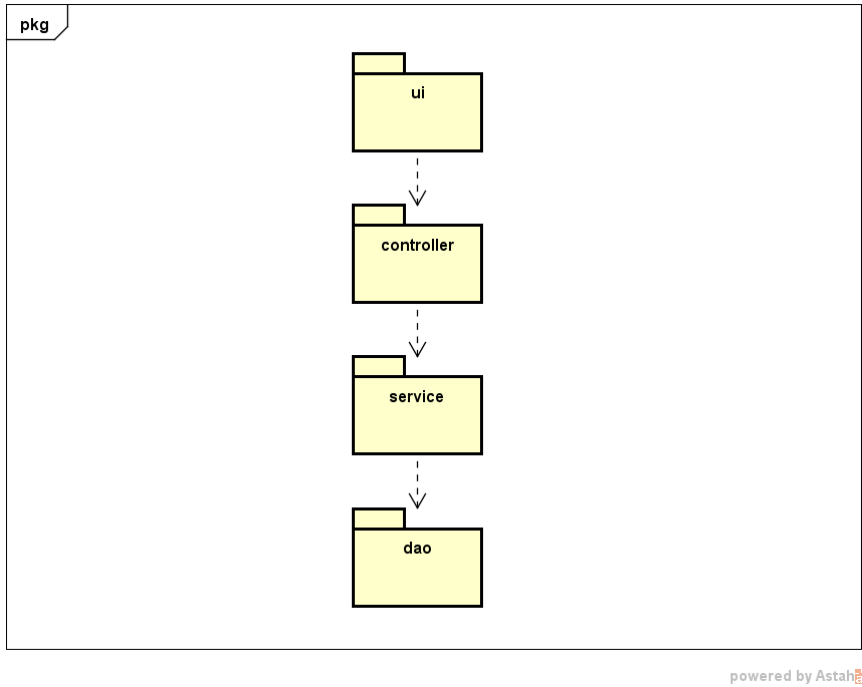


图3-1 参照体系结构风格的包图表达逻辑视角

TODO

图3-2 软件体系结构逻辑设计方案

### 4 组合视角

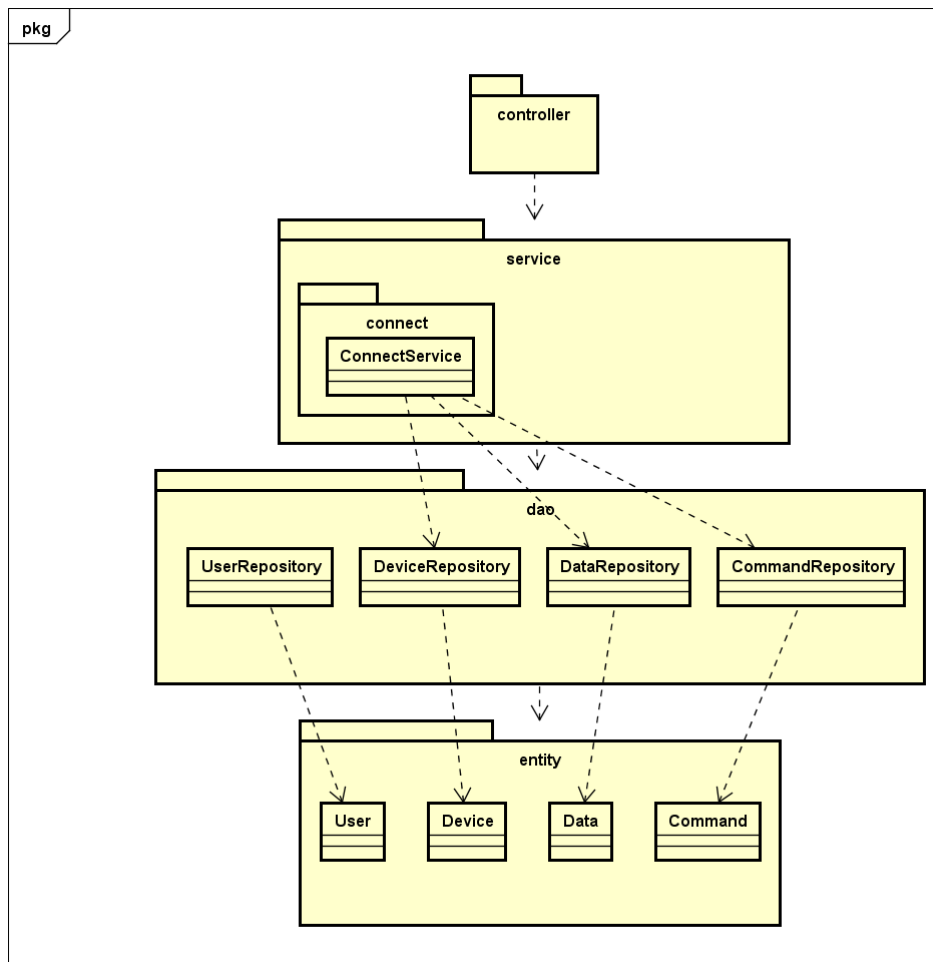
#### 4.1 开发包图

植物养护IoT平台客户端开发包图如下面的图 4-1-1 所示，服务端开发包图如图 4-1-2 所示。

TODO

图4-1-1 植物养护IoT平台客户端开发包图

TODO

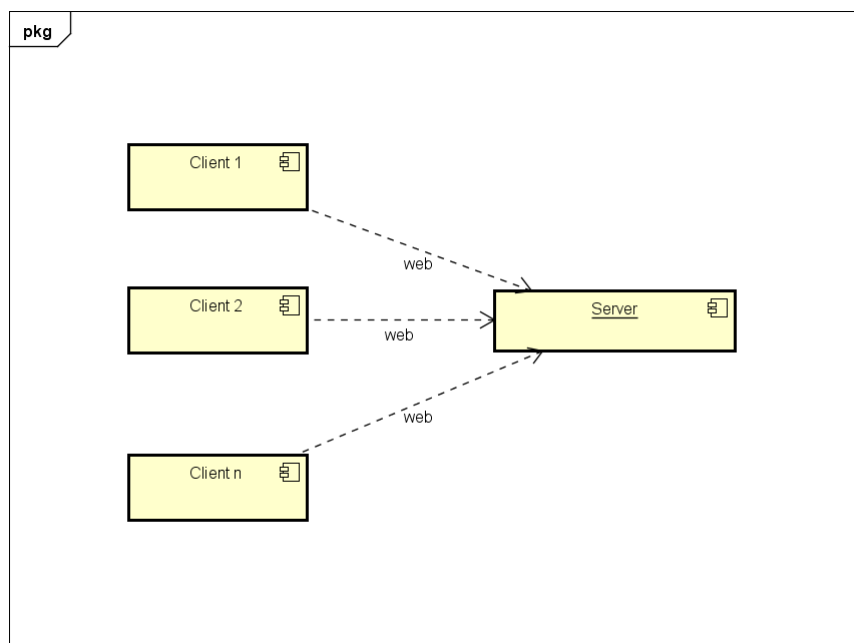


powered by Astah

图4-1-2 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，根据前端web页面划分为

// TODO

### 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	
规则引擎	
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	public boolean sendCommand(Long deviceId, String command, CommandType type);
	前置条件	设备存在，命令合法
	后置条件	向设备下发命令

## 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

### 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Entitiy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	

- 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<br>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	命令类型	string	
command	命令内容	string	