软件体系结构文档

版本号	修改时间	内容
v1.0	2023.5.30	初步完成
v2.0	2023.7.7	增加订单的支付与取消

1. 引言

1.1. 编制目的

1. 本文档详细完成了对123o6的概要设计，达到指导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
2. 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。

1.2. 词汇表

词汇名称	词汇含义	备注
l23o6	车票预订系统	无

1.3. 参考资料

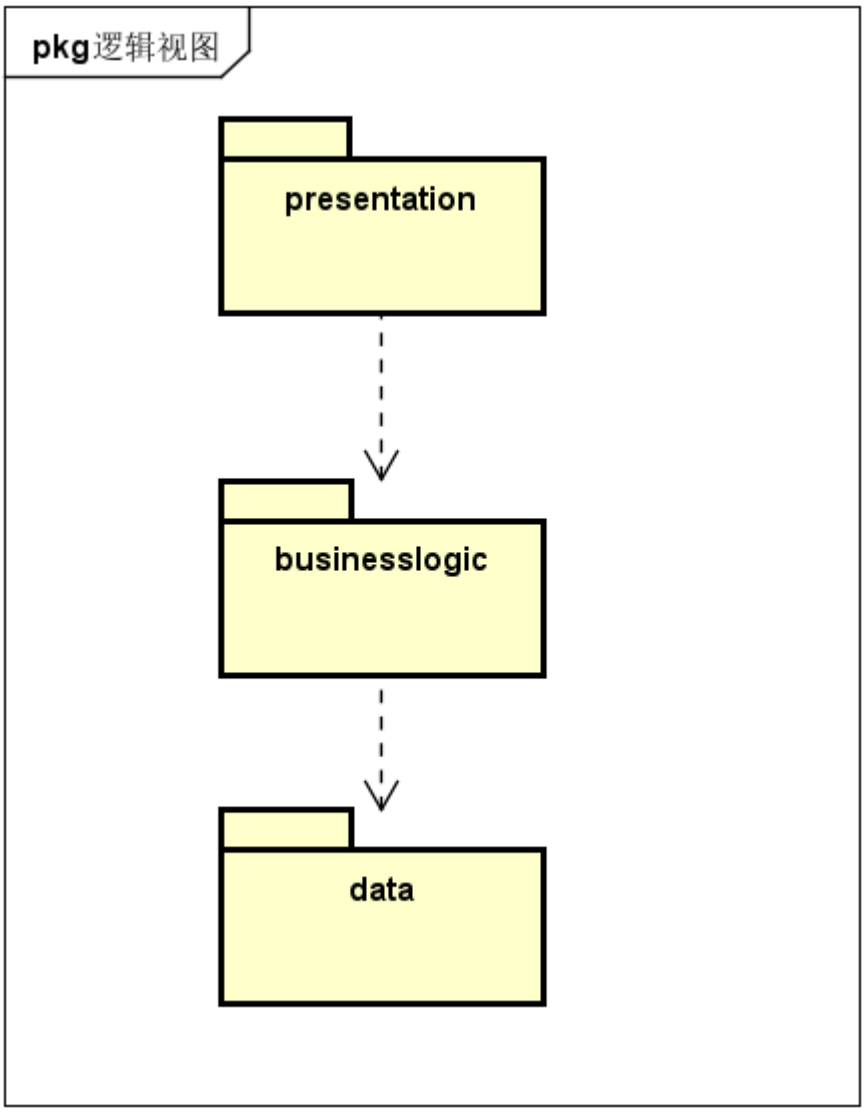
1. IEEE标准
2. 123o6用例描述文档V1.0/v2.0
3. 123o6软件需求规格说明文档V1.0/v2.0

2. 产品概述

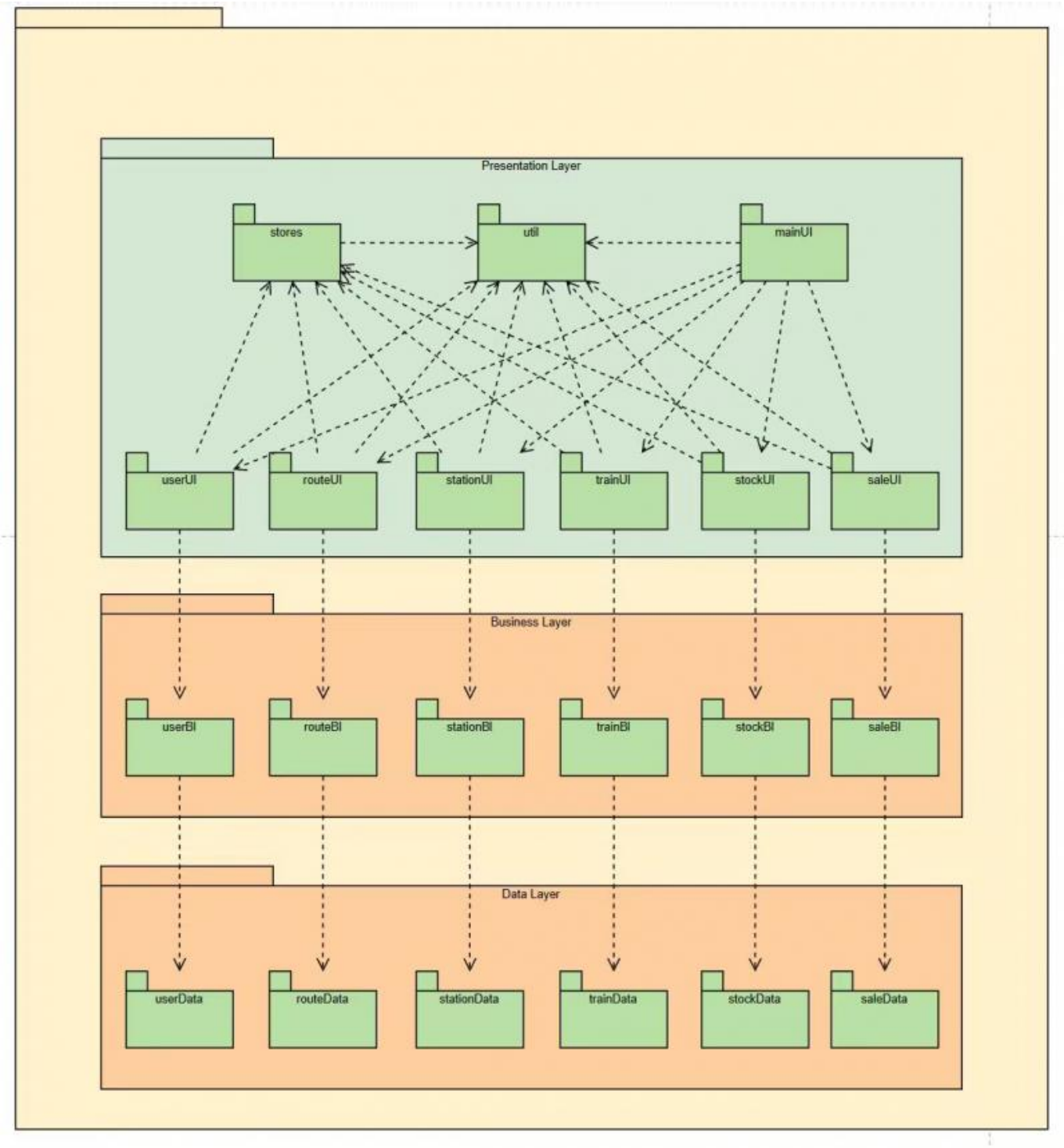
1. 参考123o6用例描述文档和123o6需求规格说明文档中对产品的概括描述。

3. 逻辑视图

- 在123o6中，选择了分层体系结构风格，将系统分为3层(展示层、业务逻辑层、数据层)能够很好地示意整个高层抽象。展示层包含GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的了逻辑视图和逻辑设计方案如下图所示。
- 参照体系结构风格的包图表达逻辑视角



- 软件体系结构逻辑设计方案



4. 组合视角

4.1. 开发包图

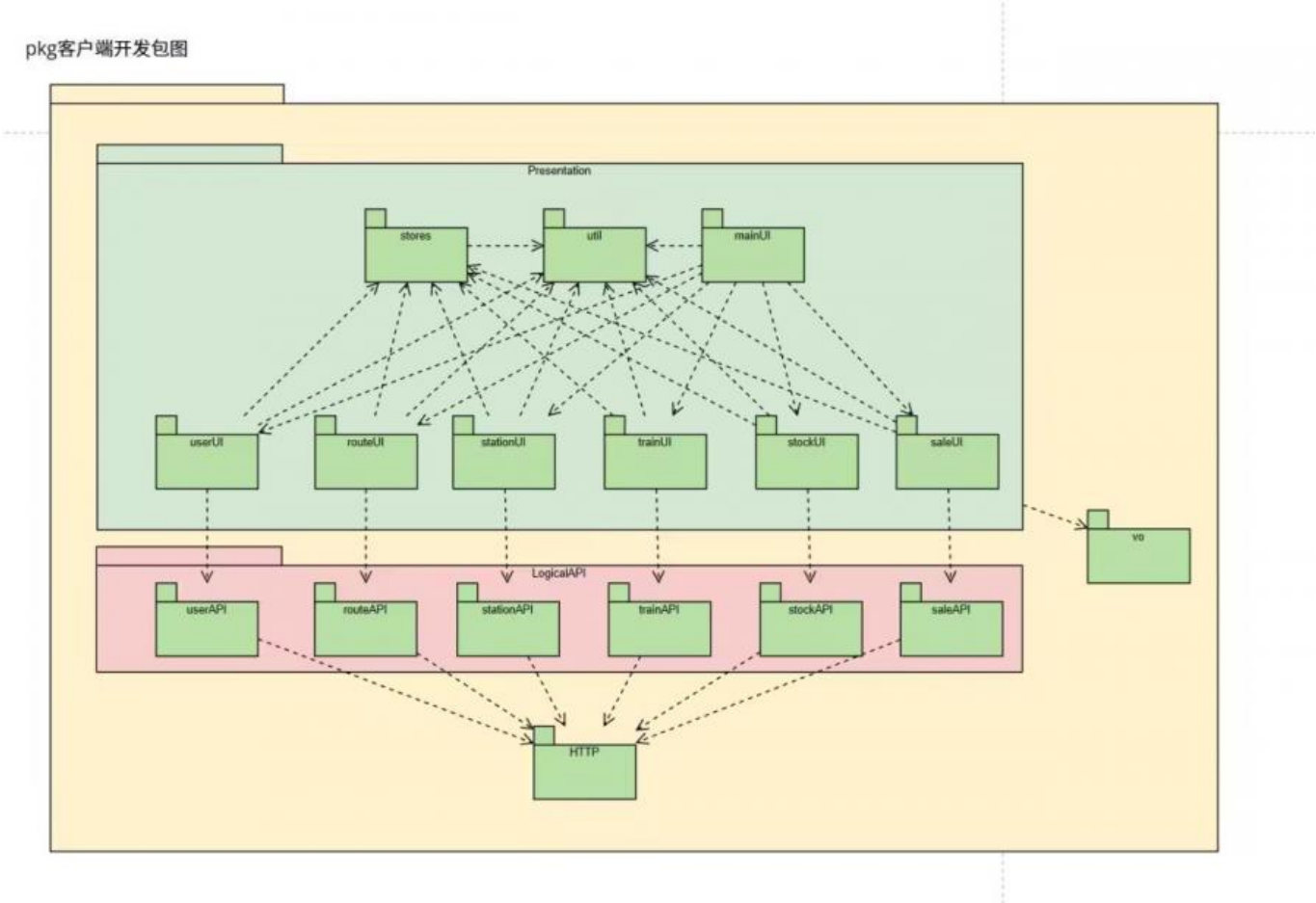
开发包	依赖的其他开发包
mainUi	userUi,routeUi,stationUi,trainUi,stockUi,orderUi,util,vo
userUi	store,util,userAPI,vo
routeUi	store，util,routeAPI,vo
stationUi	store,util,stationAPI,vo

开发包	依赖的其他开发包
trainUi	store,util,trainAPI,vo
orderUi	store,util,orderAPI,vo
util	
store	util
userAPI	HTTP
routeAPI	HTTP
stationAPI	HTTP
trainAPI	HTTP
orderAPI	HTTP
userController	HTTP,userService
routeController	HTTP,routeService
stationController	HTTP,stationService
trainController	HTTP,trainService
orderController	HTTP,orderService
userService	userMapper
routeService	routeMapper
stationService	stationMapper
trainService	trainMapper
orderService	orderMapper
userMapper	po
routeMapper	po
stationMapper	po
trainMapper	po
orderMapper	po
userData	po,userMapper
routeData	po,routeMapper
stationData	po,stationMapper
trainData	po,trainMapper
orderData	po,orderMapper
databaseutility	

开发包	依赖的其他开发包
vo	

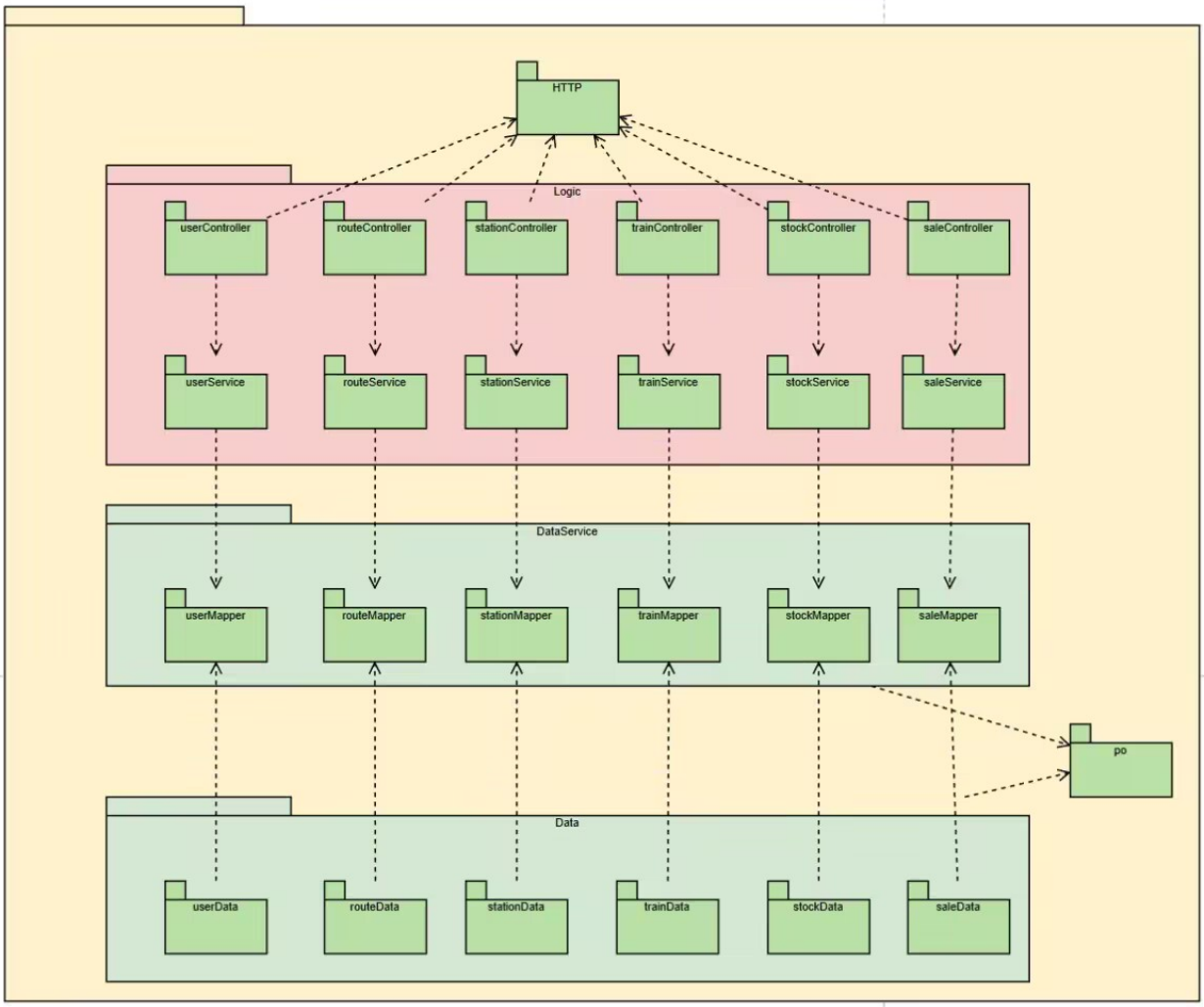
po

- 客户端开发包图



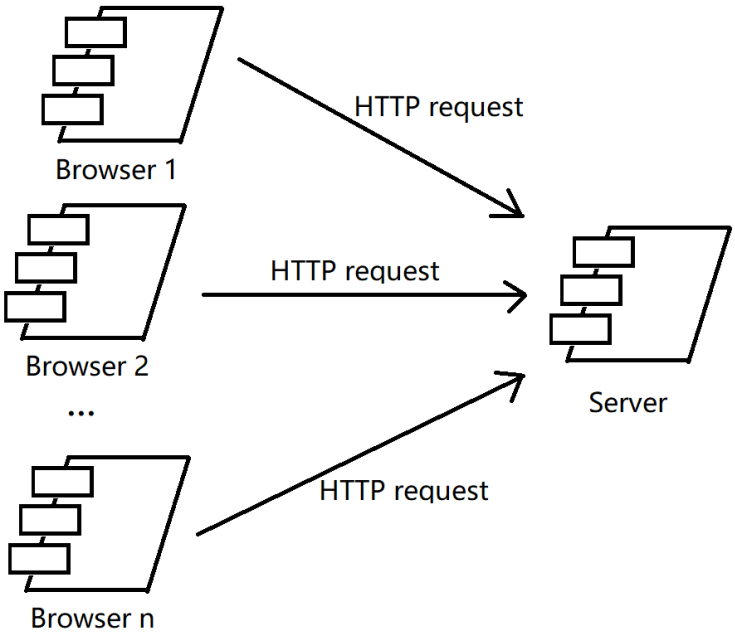
- 服务器端开发包图

pkg服务器端开发包图



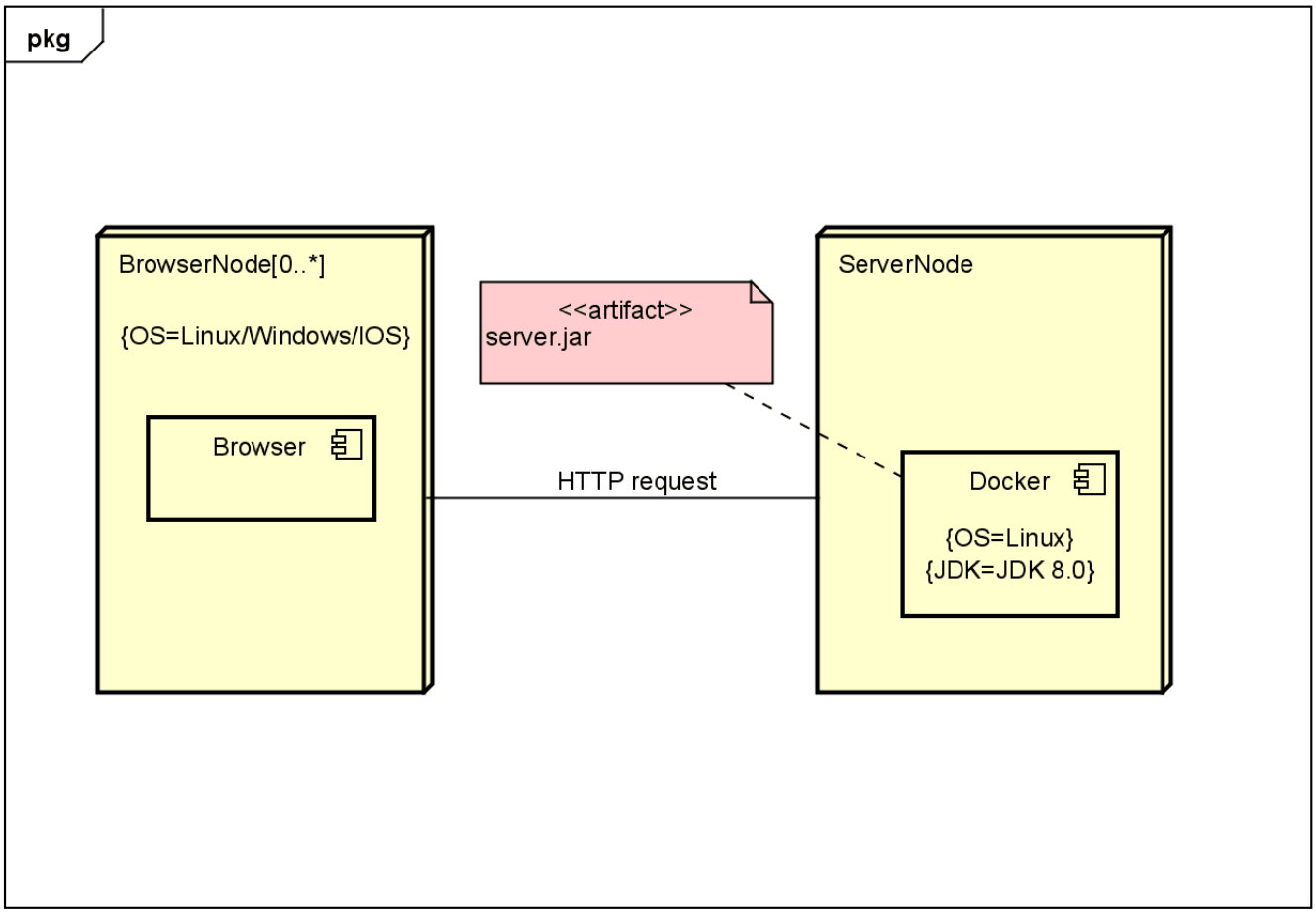
4.2. 运行时进程

- 在123o6中，会有多个网页端进程和一个服务器端进程，其进程图如下图。结合部署图，网页端进程是在浏览器上运行，服务器端进行再服务器端机器上运行。



4.3. 物理部署

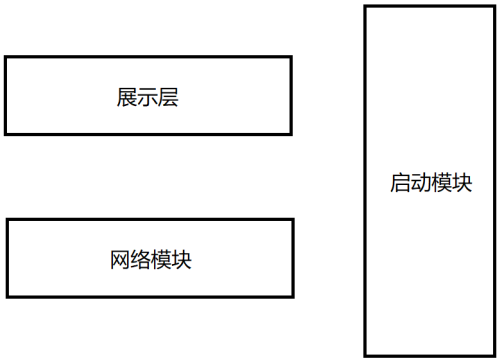
- 123o6中的网页端组件放在客户端机器上，服务器端组件都是放在服务器端机器上。由于Java RMI构建属于JDK的一部分，所以，在系统JDK环境已经设置好的情况下，不需要再单独进行部署。



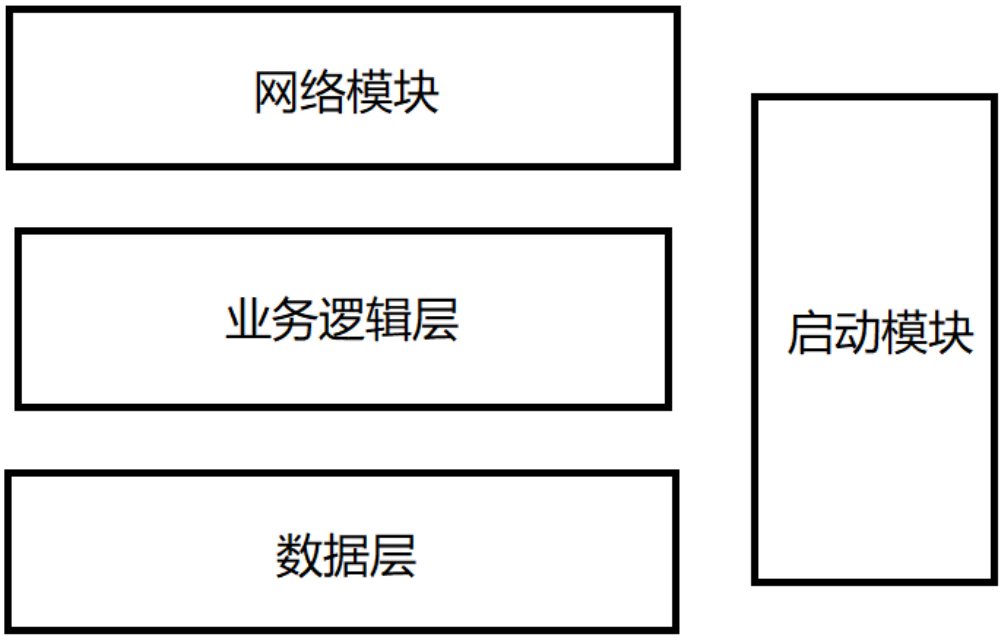
5. 架构设计

5.1. 模块职责

- 网页端模块和服务器端模块视图分别如下两张图所示（依次对应上下两张）。
  - 网页端模块视图



- 服务器端模块视图



- 客户端各层职责

层	职责

层	职责
启动模块	负责初始化网络通信机制，启动用户界面
用户界面层	基于Web界面的123o6的界面
客户端网络模块	利用 Rest API 机制发起HTTP请求

- 服务器端各层职责

层	职责
启动模块	负责初始化网络通信机制，启动用户界面
业务逻辑层	对于用户界面输入的进行响应并进行业务处理逻辑
数据层	负责数据的持久化和数据访问接口
服务端网络模块	接受客户端的HTTP请求

- 层之间调用接口

接口	服务调用方	服务提供方
trainController userController orderController stationController routeController	客户端展示层	客户端业务逻辑层
trainService userService orderService stationService routeService	客户端业务逻辑层	服务端数据层

5.2. 用户界面层分解

5.2.1. 职责

- 用户界面跳转图

模块	职责
train	负责展示车票信息、车票管理界面
user	负责展示登陆、注册、个人信息界面
station	负责展示车站管理界面
route	负责展示线路管理界面
order	负责展示订单界面

5.2.2. 接口规范

- route 模块

接口名称	内容
RouteVue.addRouteAPI	添加路线
RouteVue.delRouteAPI	删除路线
RouteVue.addRouteAPI	添加路线
RouteVue.changeRouteAPI	修改路线
RouteVue.refreshDataAPI	刷新数据
RouteDetailFormVue.addStationAPI	添加站点
RouteDetailFormVue.editStationAPI	编辑站点
RouteDetailFormVue.deleteStationAPI	删除站点

- train 模块

接口名称	内容
TrainVue.addTrainAPI	添加火车
TrainVue.delTrainAPI	删除火车
TrainVue.changeTrainAPI	更改火车
TrainVue.refreshDataAPI	刷新数据
TrainManageDetailVue.getRouteAPI	获取线路详细信息
TrainManageFormVue.getRoutesAPI	获取线路列表
TrainManageFormVue.getRoute	获取线路
searchVue.submit	查找车次

- user 模块

接口名称	内容
LoginFormVue.submitFormAPI	用户登入
RegisterFormVue.submitFormAPI	用户注册
MenuComponentVue.logoutAPI	用户登出
UserInfoComponentVue.setForm	编辑个人信息
UserInfoComponentVue.submitForm	提交个人信息

- station 模块

接口名称	内容
StationVue.renameStationAPI	修改车站信息
StationVue.addStationAPI	添加车站
StationVue.delStationAPI	删除车站

- order 模块

接口名称	内容
UserOrdersVue.getOrderDetail	订单细节
UserOrdersVue.refund	退款
UserOrdersVue.getOrders	得到所有订单
UserOrdersVue.getTrainName	得到车次名
OrderFormVue.submitOrderForm	提交订单
OrderDetailVue.getOrderDetail	订单细节
OrderDetailVue.getTrain	获取车次
OrderDetailVue.pay	支付订单
OrderDetailVue.changeUsePoints	更新使用积分
OrderDetailVue.cancel	取消订单

5.3. 业务逻辑层分解

5.3.1. 职责

模块	职责
train	负责处理车票信息、车票管理
user	负责登陆、注册、管理个人信息
station	负责车站管理
route	负责线路管理
order	负责订单支付与取消

5.3.2. 接口规范

- train 模块

接口	类型	参数	内容

接口	类型	参数	内容
TrainController.listTrains	CommonResponse<List>	ListTrainRequest request	列出火车信息
TrainController.getTrain	CommonResponse	ListTrainRequest request	获到火车信息
TrainController.addTrain	CommonResponse<?>	AddTrainRequest request	添加火车
TrainController.listTrainsAdmin	CommonResponse<List>		管理员界面列出火车信息
TrainController.getTrainAdmin	CommonResponse	Long trainId	管理员得到目标火车
TrainController.changeTrain	CommonResponse<?>	Long trainId, AddTrainRequest request	更改火车
TrainController.deleteTrain	CommonResponse<?>	Long trainId	删除火车
TrainService.getTrain	TrainDetailVO	Long trainId	得到目标火车
TrainService.listTrains	List	Long startStationId, Long endStationId, String date	列出火车信息
TrainService.listTrainsAdmin	List		管理员界面列出火车信息
TrainService. addTrain	void	String name, Long routeId, TrainType type, String date, List arrivalTimes, List departureTimes	添加火车
TrainService. changeTrain	void	Long id, String name, Long routeId, TrainType type, String date, List arrivalTimes, List departureTimes	更改火车
TrainService. deleteTrain	void	Long id	删除火车

- user 模块

接口	类型	参数	内容

接口	类型	参数	内容
TrainController.listTrains	CommonResponse<List>	ListTrainRequest request	列出火车信息
UserController.login	CommonResponse<?>	LoginRequest request	用户登入
UserController.register	CommonResponse<?>	RegisterRequest request	用户注册
UserController.logout	CommonResponse<?>		用户登出
UserController.userInfo	CommonResponse		获取用户信息，转换为视图，并返回成功响应
UserController.editInfo	CommonResponse<?>	EditUserInfoRequest request	修改个人信息
UserService.login	void	String username, String password	用户登入
UserService.editInfo	void	String username, String name, String idn, String phone, String type, String identity, double mileagePoints, double aliBalance, double wechatBalance	用户修改信息
UserService.findByUserName	UserEntity	username	通过用户名查找用户

- station 模块

接口	类型	参数	内容
StationController.listStations	CommonResponse<List>		列出车站信息
StationController.getStation	CommonResponse	Long stationId	获得目标车站
StationController.addStation	CommonResponse<?>	AddStationRequest request	添加车站
StationController.editStation	CommonResponse<?>	Long stationId,AddStationRequest request	编辑车站
StationController.deleteRoute	CommonResponse<?>	Long stationId	删除路线
StationService.getStation	StationVO	Long stationId	获得目标车站
StationService.listStations	List		列出车站信息
StationService.addStation	void	String name	添加车站

接口	类型	参数	内容
StationService.editStation	void	Long id, String name	修改车站信息
StationService.deleteStation	void	Long stationId	删除车站
StationService. validateStation	void	String name	检查车站名规范
StationService.stationIsUsed	void	Long id	处理已有订单包含车次
StationService. stationExisted	void	String name	处理车站已存在

- route 模块

接口	类型	参数	内容
routeController.addRoute	CommonResponse<?>	@Valid @RequestBody AddRouteRequest request	添加路线
routeController.getRoutes	CommonResponse<List>		找到多个目标路线
routeController.getRoute	CommonResponse	@PathVariable("routeId") Long routeId	找到一个目标路线
routeController.editRoute	CommonResponse<?>	@PathVariable("routeId") Long routeId, @Valid @RequestBody AddRouteRequest request	修改路线
routeController.deleteRoute	CommonResponse<?>	@PathVariable Long routeId	删除路线
routeService.listRoutes	List		列出路线

- order 模块

接口	类型	参数	内容
OrderController.createOrder	CommonResponse	CreateOrderRequest request	创建订单
OrderController.listOrders	CommonResponse<List>		列出所有订单

接口	类型	参数	内容
OrderController.getOrder	CommonResponse	Long orderId	根据Id得到订单
OrderController.patchOrder	CommonResponse<?>	Long orderId, PatchOrderRequest request	根据状态做出相应的行为
OrderService.createOrder	void	String username, Long trainId, Long fromStationId, Long toStationId, String seatType, Long seatNumber	创建订单
OrderService.listOrders	List	String username	列出所有订单
OrderService.getOrder	OrderVO	Long id	根据id得到订单
OrderService.cancelOrder	void	Long id	取消订单
payOrder	void	Long id	支付订单
OrderService.setUsePoints	void	Long orderId, boolean usePoints	设置是否使用积分
OrderService.setPaymentStrategy	void	Long orderId, String strategyText	设置支付策略
OrderService.refundOrder	void	Long id	退款

5.4. 数据层分解

5.4.1. 职责

模块	职责
train	负责展示车票信息、车票管理界面

模块	职责
user	负责展示登陆、注册、个人信息界面
station	负责展示车站管理界面
route	负责展示线路管理界面
order	负责展示订单界面

5.4.2. 接口规范

接口名称	内容
CrudRepository.save	保存实体对象
CrudRepository.saveAll	保存多个实体对象
CrudRepository.findById	根据id查找实体对象
CrudRepository.existsById	检查具有给定id的实体对象是否存在
CrudRepository.findAll	返回所有类型为T的实体对象
CrudRepository.findAllById	根据给定的id集合返回类型为T的实体对象集合
CrudRepository.count	返回可用的实体对象数量
CrudRepository.deleteById	根据给定的id删除实体对象
CrudRepository.delete	删除给定的实体对象
CrudRepository.deleteAllById	根据给定的id集合删除类型为T的实体对象
CrudRepository.deleteAll	除给定的实体对象集合

5.5. 信息视角

- 系统的PO类包含用户的对应的相关的实体类

5.5.1. User类

- 包含用户的ID、用户名、密码、名字、电话号码、类别、身份证号、创建时间、修改时间。
- User类的属性如下

```
public class User {
  @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotNull
    @Column(unique = true)
    private String username;
    @NotNull
    private String password;
```

```
    private String name;
    private String phone;
    private String type;
    private String idn;
    @Column
    private String identity;
    @Column
    private double mileagePoints;
    @Column
    private double aliBalance;
    @Column
    private double wechatBalance;

    @CreationTimestamp
    private Date createdAt;

    @UpdateTimestamp
    private Date updatedAt;
}
```

5.5.2. Train类

- 包含车辆编号、名字、路线号、座位售卖情况、车辆类型、日期、到达时间、出发时间、额外信息、创建日期、更新日期。
- Train类如下

```
public class Train {
   @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotNull
    private String name;

    @NotNull
    private Long routeId;

    @NotNull
    @Type(BooleanArrayType.class)
    @Column(name = "seats", columnDefinition = "boolean[][]")
    private boolean[][] seats;

    @NotNull
    private TrainType trainType;

    @NotNull
    private String date;

    @NotNull
    private List<Date> arrivalTimes;
```

```
    @NotNull
    private List<Date> departureTimes;

    @NotNull
    private List<String> extraInfos;

    @CreationTimestamp
    private Date createdAt;

    @UpdateTimestamp
    private Date updatedAt;
}
```

5.5.3. Station类

- 包含车站编号、名字、创建时间、更新日期。
- Station类如下。

```
public class Station {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotNull
    @Column(unique = true)
    private String name;

    @CreationTimestamp
    private Date createdAt;

    @UpdateTimestamp
    private Date updatedAt;
}
```

5.5.4. Route类

- 包含线路编号、名字、车站编号、创建日期、更新日期。
- Route类如下

```
public class Route{
     @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotNull
    @Column
    private String name;
```

```
    private List<Long> stationIds;

    @CreationTimestamp
    private Date createdAt;

    @UpdateTimestamp
    private Date updatedAt;
}
```

5.5.5. Order类

- 包含订单号、用户编号、车辆编号、始发站终点站编号、价格、使用积分、奖励积分、消耗积分、订单状态、座位信息、创建时间、修改时间等。
- Order类

```
public class Order{
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotNull
    private Long userId;

    @NotNull
    private Long trainId;

    @NotNull
    private Long departureStationId;

    @NotNull
    private Long arrivalStationId;

    @Column
    private double price;
    @Column
    private boolean usePoints;
    @Column
    private double bonusPoints; //奖励积分
    @Column
    private double consumptionPoints; //消耗积分
    @Column
    private PaymentStrategy.strategy paymentStrategy;

    @NotNull
    private OrderStatus status;

    @NotNull
    private String seat;

    @CreationTimestamp
    private Date createdAt;
```

```
    @UpdateTimestamp
    private Date updatedAt;
}
```