# 时刻表服务项目

# (2016/10/9编写)

## 项目整体介绍

1. 项目背景

车来了app中当一条线路没有实时数据的时候会降级成一个历史时刻表的展现方式，本项目的目的就是为车来了app后台提供历史时刻表数据服务。

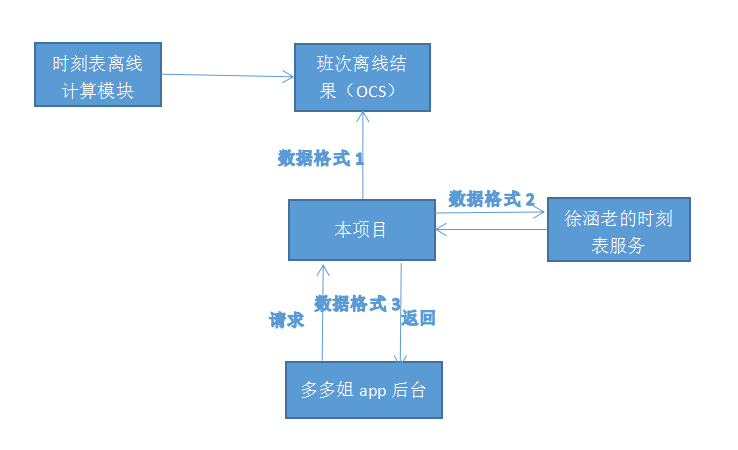
2）项目介绍

时刻表服务项目的功能是提供一个数据接口服务，返回回某一个站点的最近三个历史时刻表班次，通讯协议使用的http,网络框架使用的是springMVC+tomcat，处理逻辑框架使用的rxjava。

输入数据为http请求和离线计算的到站班次数据，输入数据为查询站点最近的三个历史班次。

## 项目与外部模块交互关系

该项目主要外部交互的模块为，时刻表离线计算模块，多多姐app后台模块，以及徐涵的时刻表服务模块（为了区别于本项目在后文中该模块记为老的时刻表服务），整体结构如图所示：



其中涉及到三处数据格式：

1. 数据格式1:本项目从OCS中读取到的离线计算结果。

该数据是存储在OCS中的一个key value 结构：key为cityid#lineNo#direction#order,例如027#100#1#2 表示北京100路1方向第2站的所有班次，value为一个到站班次列表是是一个jSONArray的形式：[{“arriveTime”:”09:00”, “calcTime”:”2016-10-09 08:00”},{“arriveTime”:”10:00”, “calcTime”:”2016-10-09 09:00”}] 里面有两个字段，arriveTime表示该班次到站时间，calcTime表示该班次被计算出来的时间。

线上OCS地址：e7bfd8efc27145da.m.cnhzaliqshpub001.ocs.aliyuncs.com

1. 数据格式2:本项目请求徐涵老的历史时刻表服务的请求url和返回数据格式

详细格式参见wiki：<http://wiki-rd.chelaile.net.cn/index.php?title=%E5%BC%80%E5%8F%91:%E6%97%B6%E5%88%BB%E8%A1%A8%E6%8E%A5%E5%8F%A3#next_cars.E6.8E.A5.E5.8F.A3>

1. 数据格式3:多多姐请求本项目的请求url和返回结果格式

详细格式参见wiki： <http://wiki-rd.chelaile.net.cn/index.php?title=%E5%BC%80%E5%8F%91:%E6%97%B6%E5%88%BB%E8%A1%A8%E6%8E%A5%E5%8F%A3#timetable.E6.8E.A5.E5.8F.A3>

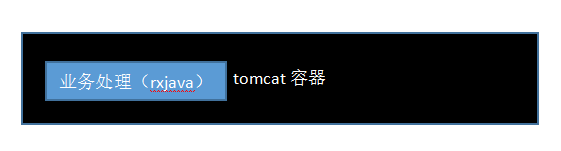
<http://wiki-rd.chelaile.net.cn/index.php?title=%E5%BC%80%E5%8F%91:%E6%97%B6%E5%88%BB%E8%A1%A8%E6%8E%A5%E5%8F%A3#timetable_batch.E6.8E.A5.E5.8F.A3>

这里要重点说明的是：依赖于徐涵老的历史时刻表这个设计是因为当时上线之后因为发现新的线路覆盖率比老的时刻表服务差了太多，为了保证覆盖率不下降而采取的临时方案。该方案中如果一个请求没有数据则这个请求会被转到老的时刻表服务，这个随着新的班次计算覆盖率越来越高最终老的时刻表服务会完全下线。

## 项目系统架构以及关键类的实现

### 3.1 项目系统架构

1）整体web结构



该项目整体架构是一个J2EE的方式，业务逻辑放在一个tomcat容器中提供对外的http服务。

1. 业务处理模块的流程

业务处理模块是通过rxjava形成了一个异步的数据流的处理。如下图所示，主要部分是由两个分支，一个分支是当多多姐的请求包含需要车辆下一站位置的时候需要再异步读一次OCS，另外一个分支是如果当前结果为空则把请求转给徐涵老的时刻表服务并返回他的结果，如下图所示：

下面对每个流程的处理分别进行介绍

1. 过滤出班次

该流程主要是通过多多姐请求参数中的时间戳和班次数过滤出当前请求站点合适的班次信息。过滤的规则是从读取出的班次中过滤出到站时间在给定时间戳之后最近的n个班次。

1. 获取车辆位置

该流程是作为一个保留的流程，主要设计目的是当请求包含需求车辆位置的时候再去OCS中查询出车辆位置，目前还没有实现。

1. 格式化结果

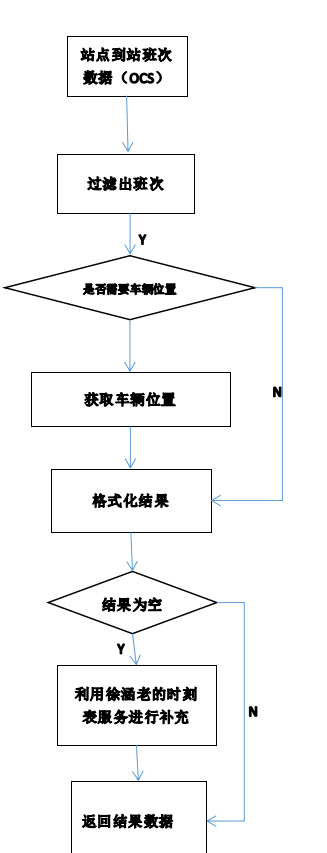
该流程是把一个json对象转化为一个内部的对象，方便数据转化为最终结果。

1. 利用老的时刻表服务进行补充

该流程的作用是当请求结果在当前处理过程为空的时候，转到徐涵老的历史时刻表，获取到返回的结果，通过一个HTTP服务进行通信，具体通信方式参见：第2节中数据格式2的介绍。

1. 返回结果

将结果数据写回给请求者，完成这一次http请求。



N

Y

利用徐涵老的时刻表服务进行补充

N

Y

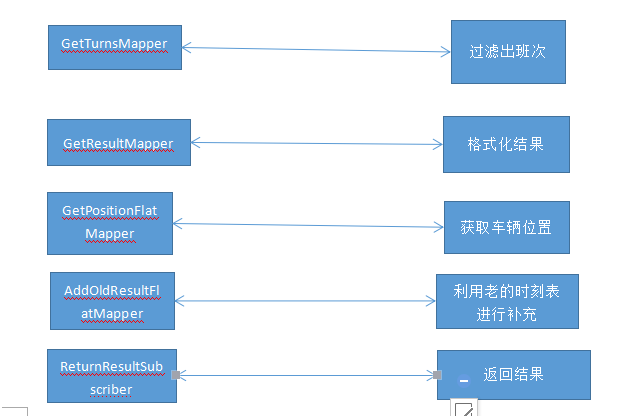
结果为空

结果转化为ReturnResult(GetResultMapper)

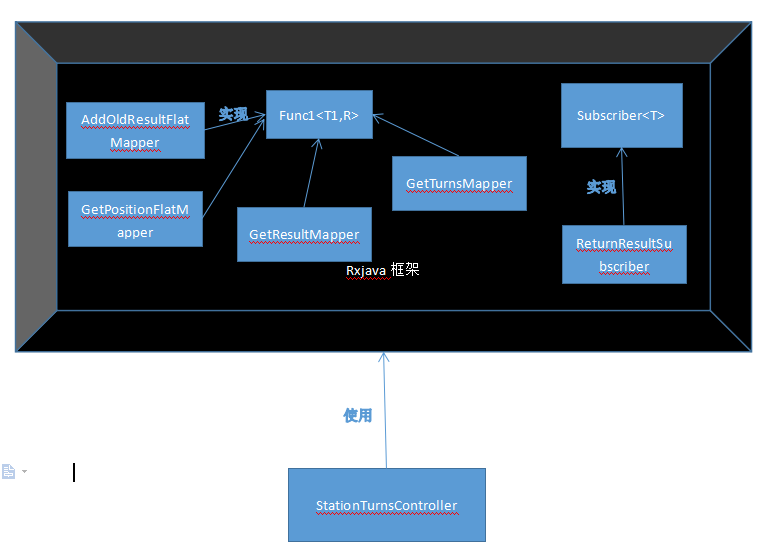
站点到站班次下一站Order（GetPositionFlatMapper）

### 3.2项目类结构

1. 主要类与功能的对应关系



2）类与框架的关系



从上图可以看出在rxjava框架中注册了一些FlatMapper和Mapper类以及Subscribe类，外层的Controller类使用rxjava框架，rxjava框架再去回调内部注册的类进行逻辑处理。有关rxjava框架进一步知识可以参考https://mcxiaoke.gitbooks.io/rxdocs/content/Topics.html

## 项目部署和测试

svn地址：svn://svn.yg84.com/chelaile/da/TimeTableService

数据格式wiki地址：

单条接口：

<http://wiki-rd.chelaile.net.cn/index.php?title=%E5%BC%80%E5%8F%91:%E6%97%B6%E5%88%BB%E8%A1%A8%E6%8E%A5%E5%8F%A3#timetable.E6.8E.A5.E5.8F.A3>

批量接口：

<http://wiki-rd.chelaile.net.cn/index.php?title=%E5%BC%80%E5%8F%91:%E6%97%B6%E5%88%BB%E8%A1%A8%E6%8E%A5%E5%8F%A3#timetable_batch.E6.8E.A5.E5.8F.A3>

启动方式是直接把项目打成war包部署在tomcat的webapps下面,目前路径为/opt/apache-tomcat-8.5.3/webapps/，然后到/opt/apache-tomcat-8.5.3/bin路径下执行./start.sh启动tomcat，tomcat目前监听的是80端口可以通过修改/opt/apache-tomcat-8.5.3/conf/server.xml配置文件来修改监听端口,目前服务部署在TB3上，外层nginx部署在TB1。

本项目目前没有单元测试，目前的测试方式是在测试服务器启动一个实例，手动发送http请求观察返回结果进行测试，也可以用ab等工具对其进行压测。

## 项目代码介绍

1. 项目入口代码：com.yg.timetableservice.controller.StationTurnsController

其中定义了两个接口/timetable(单条接口) /timetable\_batch(批量接口)具体的协议参见wiki。

1. com.yg.timetableservice.rxjava包中定义了数据的主要处理逻辑，其中：

GetTurnsMapper/GetBatchTurnsMapper类主要是根据page和timestamp两个参数过滤读取到的班次数据。

GetPositionFlatMapper/GetBatchPositionFlatMapper类主要是根据timestamp来获取每个班次当前的下一站stoporder.

GetResultMapper/GetBatchResultMapper类是把结果从JSONArray转成一个ReturnResult对象。

ReturnResultSubscriber类是把结果写回到一个DeferredResult中。

1. com.yg.timetableservice.utils包定义了一些辅助的工具，比如操作OCS的和访问http的，这些工具的方式都是异步IO，返回一个rxjava里面的Observable.

返回结果数据（ReturResultSubscribe）