System Design for Dada

Table of Contents

[1 Architecture 2](#_Toc13606234)

[1.1 User Management Service 2](#_Toc13606235)

[1.2 Position Tracking Service 2](#_Toc13606236)

# Architecture

First, I would break down the application into a set of loosely decoupled, collaborating services as follows:

* User Management
* City Management
* Position Tracking
* Order Management
* Deal Maker Service
* Payment Management
* Distributed Lock Service

## Micro Services

### User Management Service

User management service handles user registration, user login and logout, user role (driver or passenger) and other account related stuff.

It exposes following APIs:

* Register new account
* Update account role as driver or passenger or both
* User login
* User logout
* User authentication
* User authorization

### City Management Service

City management service manage cities on-boarded on Dada.

It exposes following APIs:

* Add new city
* Disable city
* Get all enabled cities

### Position Tracking Service

Position tracking service records the real-time location (latitude, longitude and height) of drivers and passengers whenever Data app is opened on mobile phones.

It exposes following APIs:

* Update real-time location (latitude, longitude and height)
* Find closest available driver

### Order Management Service

Order management service manage orders for users. It tracks passengers and drivers order history and create new order.

It exposes following APIs:

* Create new order for specified passenger and driver
* Get order history
* Get current order (active order)
* Update order status

### Deal Maker Service

Deal maker service’s responsibility is to match passenger with the closest available driver and create a new order for them.

It exposes following APIs:

* Create new order for passenger

### Payment Management Service

Payment management service handles the deal payment.

It exposes following APIs:

* Add new payment method
* Update/remove payment method
* Update order payment status

### Distributed Lock Service

This service provides a distributed lock service for other service.

## API Definitions and Database Schemas

All micro services expose RESTful APIs.

The code field returned from APIs is global unique.

### User Management Service

#### Database Schemas

Use mongo db.

#### API Definitions

* Register new account

|  |  |  |
| --- | --- | --- |
| Request | | |
| URL | https://{domain}/api/v1/user |  |
| Method | Post |  |
| Body |  |  |
| Response | | |
| Body | {  "code": 0,  "message": "succeeded"  } |  |

* Update account role as driver or passenger or both
* User login
* User logout
* User authentication
* User authorization

## Diagrams