

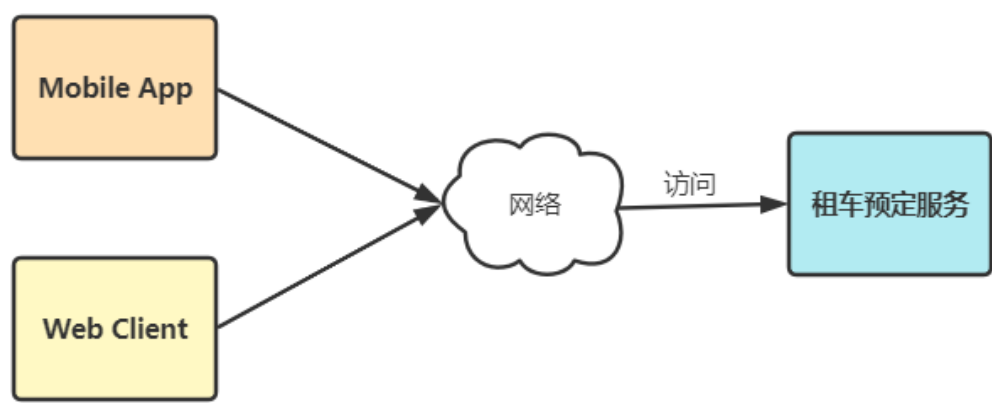
租车预定服务设计(Rental Car Booking Service Design)

一、需求分析概述

为客户实现租车预定服务，以满足客户对租车的查询及预定需求。从客户角度看，客户首先需要知道租车服务提供商有多少车，有哪些型号的车子，客户可以根据喜好选择自己喜欢的车子。因此租车预定服务需要一个提供车辆库存信息的接口，返回每个型号的车子的数量。其次客户需要知道车子在哪些时间段是可以被预定的，以避免预定冲突。所以租车预定服务需要一个返回已有租赁信息的接口，辅助客户选择租赁时间段。最后就是租车预定接口，供客户下单预定。

综上所述，租车预定服务需要提供三个接口，分别是查询车辆库存信息接口，查询车辆租赁信息接口以及租车预定接口。

二、系统设计



三、API接口设计——基于Restful API设计

0.统一错误处理

Responses:

400 Failed Operation

名称	类型	是否必须	默认值	描述
error	object	是		错误信息
└ code	integer	是		业务错误码
└ message	string	是		错误描述

Response body samples:

```
{
  "error": {
    "code": 2103,
    "message": "预定时间段冲突"
  }
}
```

当前系统存在的业务错误码

业务错误码	错误描述
2101	未输入日期或日期格式不正确，正确的格式为（yyyy-MM-dd）
2102	预定时间段冲突
2103	没有输入型号或没有该型号的车辆
2104	该型号的车辆无库存
2105	结束日期必须等于或大于开始日期
2106	无法解析JSON参数

1.查看车辆库存信息接口

Path: GET /v1/cars

QUERY PARAMETERS:

名称	类型	默认值	描述
carModel	string	无	汽车型号，用以查询指定型号的库存信息

Responses:

200 Successful Operation

名称	类型	是否必须	默认值	描述
cars	object []	必须		车辆库存信息
└ carModel	string	必须		车辆型号
└ inStock	integer	必须		剩余数量

Response body samples:

```
{
  "cars": [
    {
      "carModel": "Toyota Camry",
      "inStock": 2
    },
    {
      "carModel": "BMW 650",
      "inStock": 2
    }
  ]
}
```

2.查看车辆租赁信息接口

用来查看车辆的租赁信息

Path: GET /v1/rentals

Responses:

200 Successful Operation

名称	类型	是否必须	默认值	描述
rentals	object []	是		租赁信息
└ carID	integer	是		车辆唯一标识
└ carModel	string	是		车辆型号
└ startDate	string	是		开始日期 (yyyy-MM-dd, UTC)
└ endDate	string	是		结束日期 (yyyy-MM-dd, UTC)

Response body samples:

```
{
  "rentals": [
    {
      carID: 1,
      carModel: "BMW 650",
      startDate: "2022-03-16",
      endDate: "2022-03-18"
    },
    {
      carID: 2,
      carModel: "BMW 650",
      startDate: "2022-04-16",
      endDate: "2022-04-18"
    },
    {
      carID: 3,
```

```
        carModel: "Toyota Camry",
        startDate: "2022-03-16",
        endDate: "2022-03-18"
    },
]
}
```

3.预定车辆接口

Path: POST /v1/rentals

Request Headers

参数名称	参数值	是否必须	备注
Content-Type	application/json	是	

Request Body

名称	类型	是否必须	默认值	描述
carModel	string	是		车辆型号
startDate	string	是		开始日期 (yyyy-MM-dd, UTC)
endDate	string	是		结束日期 (yyyy-MM-dd, UTC)

Request Body samples

```
{
  "carModel": "Toyota Camry",
  "startDate": "2022-03-16",
  "endDate": "2022-03-18"
}
```

Responses:

200 Successful Operation

名称	类型	是否必须	默认值	描述
carID	integer	是		车辆唯一标识
carModel	string	是		车辆型号
startDate	string	是		开始日期 (yyyy-MM-dd, UTC)
endDate	string	是		结束日期 (yyyy-MM-dd, UTC)

Response Body samples

```
{
  carID:1
  carModel: "Toyota Camry",
  startDate: "2022-03-16",
  endDate: "2022-03-18"
}
```

四、数据存储设计

为简单起见，相关业务数据我们使用完全基于内存的设计。

五、测试用例

1.查询车辆库存信息

1) 查询所有车辆库存

Request:

```
GET http://120.78.92.119:8080/v1/cars
```

Response:

```
{
  "cars": [
    {
      "carModel": "Toyota Camry",
      "inStock": 2
    },
    {
      "carModel": "BMW 650",
      "inStock": 2
    }
  ]
}
```

2) 查询指定型号车辆的库存

Request:

```
GET http://120.78.92.119:8080/v1/cars?carModel=BMW 650
```

Response:

```
{
  "cars": [
    {
      "carModel": "BMW 650",
      "inStock": 2
    }
  ]
}
```

3) 如果输入一个不存在的型号Tesla

Request:

```
GET http://120.78.92.119:8080/v1/cars?carModel=Tesla
```

Response:

```
{
  "error": {
    "code": 2103,
    "message": "没有输入型号或没有该型号的车辆"
  }
}
```

2.查询车辆租赁信息

1) 查询所有车辆租赁信息

Request:

```
GET http://120.78.92.119:8080/v1/rentals
```

Response:

```
{
  "rentals": [
    {
      "carID": 1,
      "carModel": "Toyota Camry",
      "startDate": "2020-03-31",
      "endDate": "2020-03-31"
    },
    {
      "carID": 3,
      "carModel": "BMW 650",
      "startDate": "2020-06-01",
      "endDate": "2020-06-21"
    },
    {
      "carID": 4,
      "carModel": "BMW 650",
      "startDate": "2020-06-01",
      "endDate": "2020-06-21"
    }
  ]
}
```

```
}  
]  
}
```

3.预定车辆

1) 预定指定型号的车辆

Request:

POST http://120.78.92.119:8080/v1/rentals

header:

Content-Type application/json

body:

```
{  
  "carModel": "BMW 650",  
  "startDate": "2022-05-01",  
  "endDate": "2022-05-21"  
}
```

Response:

```
{  
  "carID": 3,  
  "carModel": "BMW 650",  
  "startDate": "2022-05-01",  
  "endDate": "2022-05-21"  
}
```

2) 预定不存在的型号的车辆

Request:

POST http://120.78.92.119:8080/v1/rentals

header:

Content-Type application/json

body:

```
{  
  "carModel": "Tesla",  
  "startDate": "2022-05-01",  
  "endDate": "2022-05-21"  
}
```

Response:

```
{
  "error": {
    "code": 2103,
    "message": "没有输入型号或没有该型号的车辆"
  }
}
```

3) 预定时间格式错误

Request:

```
POST http://120.78.92.119:8080/v1/rentals
```

header:

```
Content-Type application/json
```

body:

```
{
  "carModel": "BMW 650",
  "startDate": "2022/05/01",
  "endDate": "2022/05/21"
}
```

Response:

```
{
  "error": {
    "code": 2101,
    "message": "未输入日期或日期格式不正确，正确的格式为（yyyy-MM-dd）"
  }
}
```

4) 预定时间冲突

在查询车辆租赁信息示例中可以看到两辆BMW 650车型的都已在2020-06-01到2020-06-21时间段被预定。

Request:

```
POST http://120.78.92.119:8080/v1/rentals
```

header:

```
Content-Type application/json
```

body:

```
{
  "carModel": "BMW 650",
  "startDate": "2020-06-20",
  "endDate": "2020-06-25"
}
```

Response:


```
{
  "error": {
    "code": 2102,
    "message": "预定时间段冲突"
  }
}
```

六、结尾

这个租车预定服务已在阿里云上部署运行，地址为：

<http://120.78.92.119:8080>

代码已上传至github，地址为：

<https://github.com/Cinple/rentalCarBooking-server>