Design description of rental car booking API service

Background

Design a simple rental car booking API service for customers to reserve a car for a period time.

Functional module

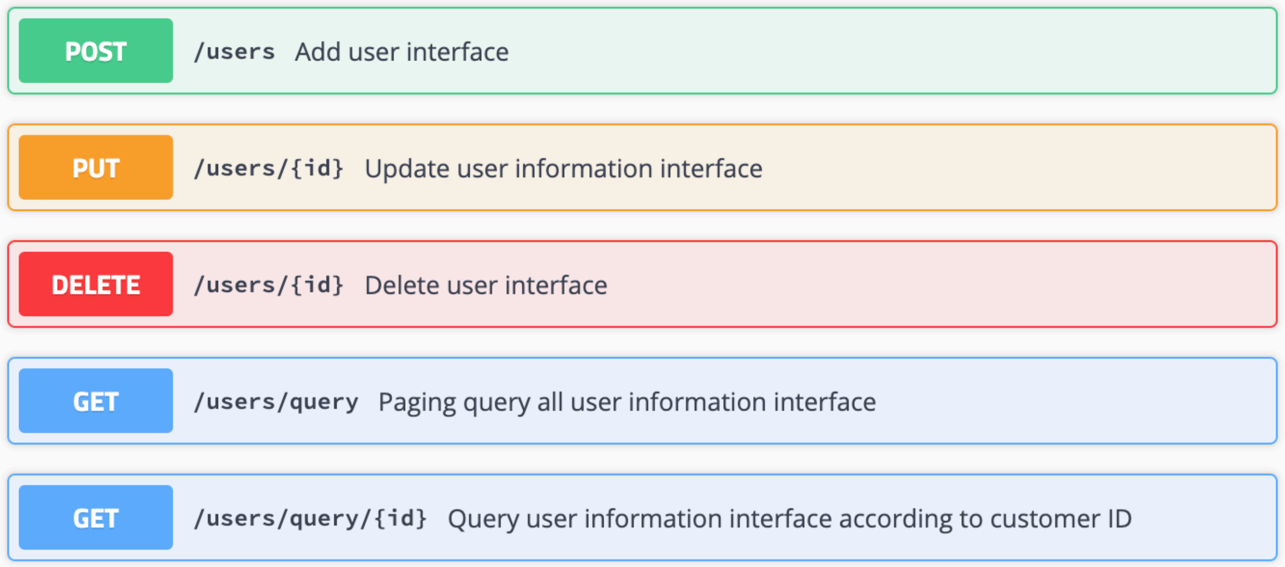
It is divided into three main functional modules: user management module, car management module and reservation management module.

• User management module

As shown below, a simple user table is designed to store user data.

```
create table users(
id int not null primary key comment 'user id',
name char(20) not null default '0' comment 'user name',
phone char(11) not null comment 'user phone',
password char(8) not null comment 'user password'
);
```

The API services provided are shown in the figure below.



POST /users Add user interface

PUT /users/{id} Update user information interface

DELETE /users/{id} Delete user interface

GET /users/query Paging query all user information interface

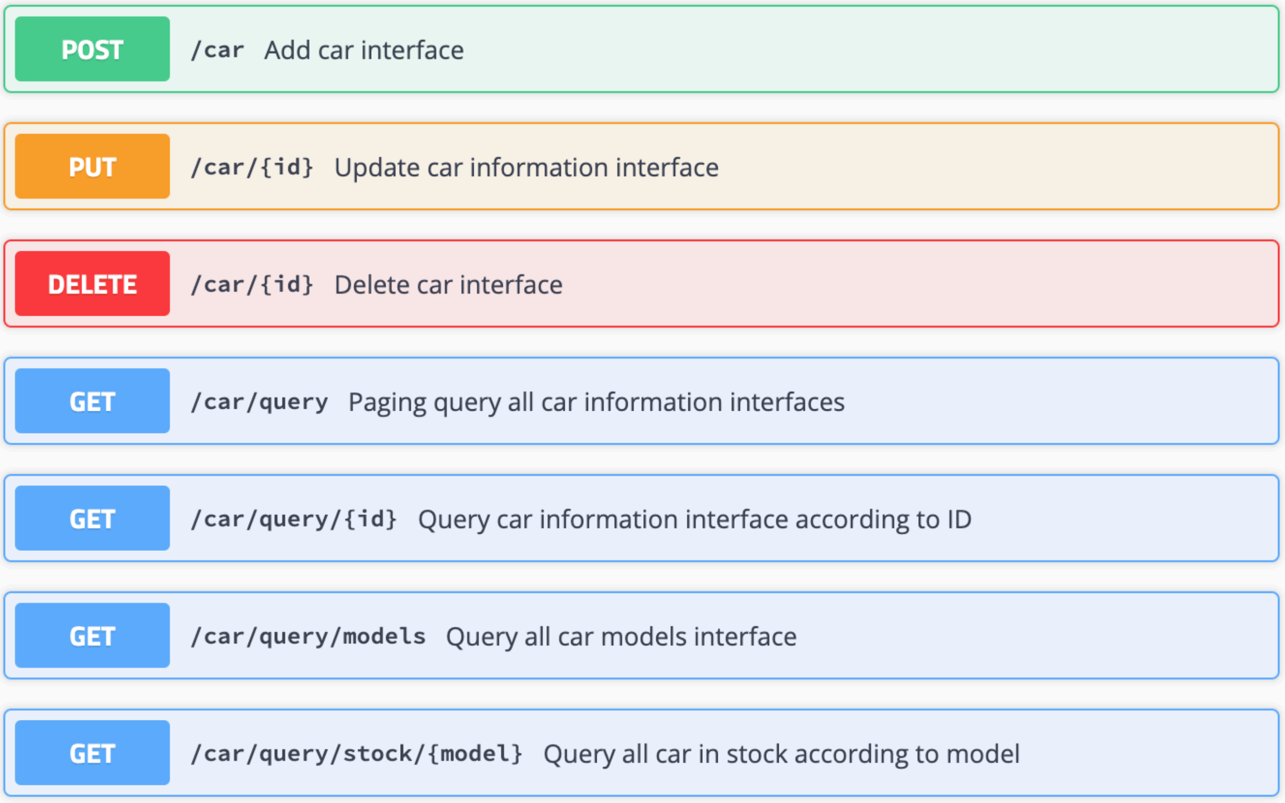
GET /users/query/{id} Query user information interface according to customer ID

• Car management module

As shown below, a simple car table is designed to store car data.

```
create table car(
id char(7) not null primary key comment 'registration
number',
model char(20) not null comment 'car model',
status char(1) not null default '0' comment 'car status',
price decimal not null default 0 comment 'rent per day'
);
```

The API services provided are shown in the figure below.



POST /car Add car interface

PUT /car/{id} Update car information interface

DELETE /car/{id} Delete car interface

GET /car/query Paging query all car information interfaces

GET /car/query/{id} Query car information interface according to ID

GET /car/query/models Query all car models interface

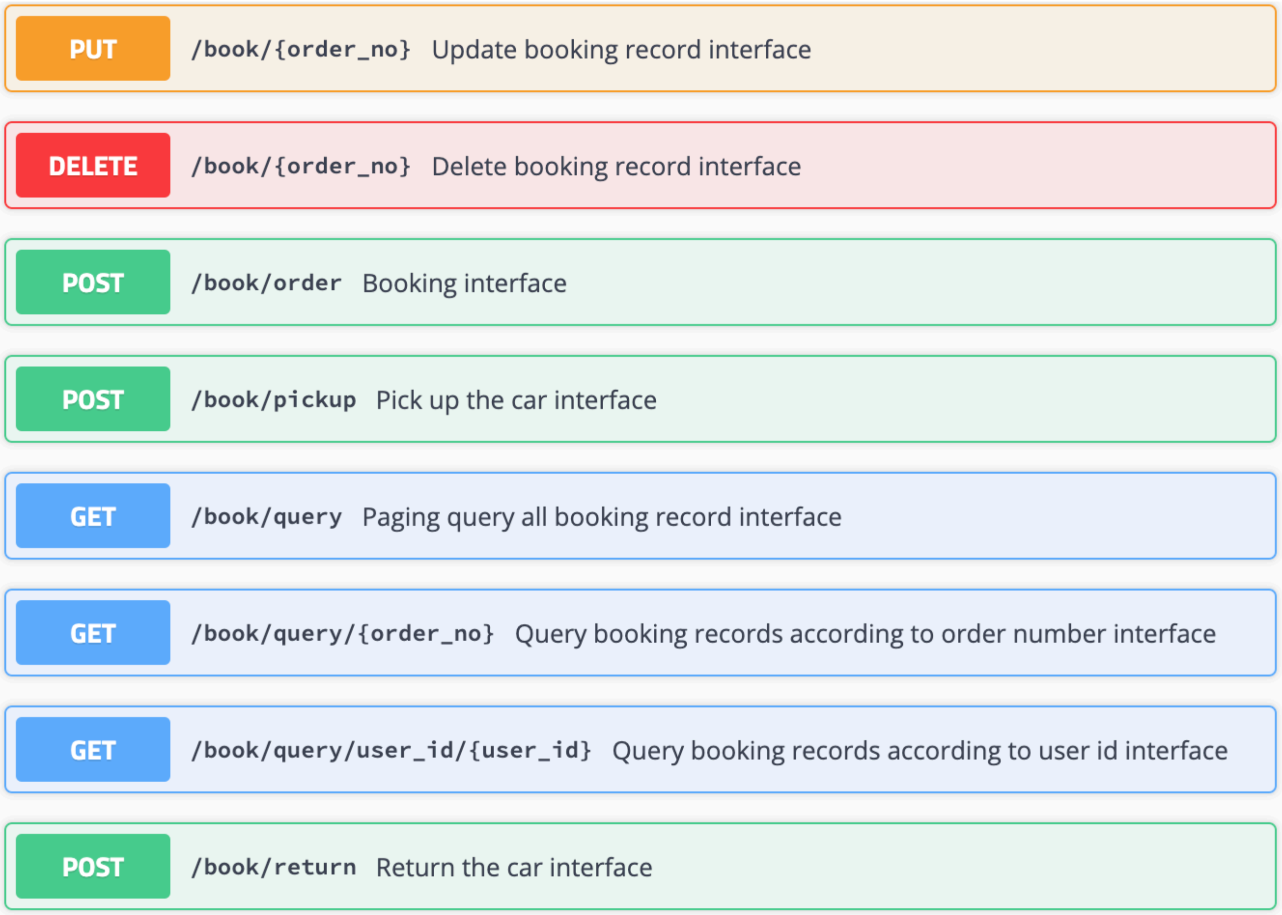
GET /car/query/stock/{model} Query all car in stock according to model

- ## Booking management module

As shown below, a simple booking table is designed to store booking records.

```
create table booking(
order_no char(13) not null primary key comment 'booking order
number',
user_id char(6) not null comment 'user id',
car_id char(7) not null comment 'car id',
status char(1) not null default '0' comment 'booking status',
rent decimal default 0 comment 'total rent',
pickup_date datetime not null comment 'pick up date',
return_date datetime not null comment 'return date',
create_time datetime not null default current_timestamp
comment 'create time'
);
```

The API services provided are shown in the figure below.



PUT `/book/{order_no}` Update booking record interface

DELETE `/book/{order_no}` Delete booking record interface

POST `/book/order` Booking interface

POST `/book/pickup` Pick up the car interface

GET `/book/query` Paging query all booking record interface

GET `/book/query/{order_no}` Query booking records according to order number interface

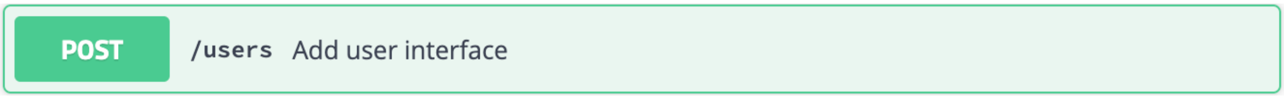
GET `/book/query/user_id/{user_id}` Query booking records according to user id interface

POST `/book/return` Return the car interface

Customer usage scenarios

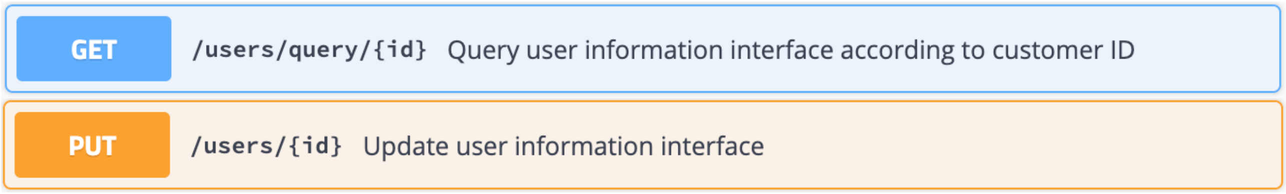
Based on the above API services, introduce the customer usage scenarios.

1、User registration
New users can register through the *Add user interface*.



POST `/users` Add user interface

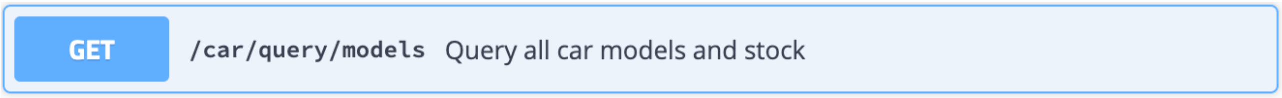
2、User information query and modification
Registered users can query their own user information through user ID, and can also modify their own information.



GET `/users/query/{id}` Query user information interface according to customer ID

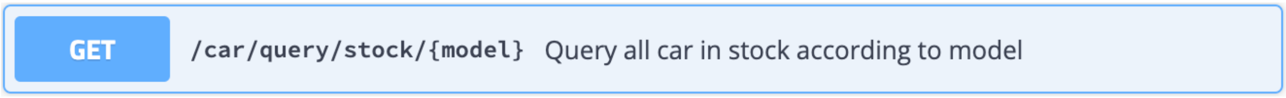
PUT `/users/{id}` Update user information interface

3、Query all car models and stock
Users can query all car models and corresponding stock.

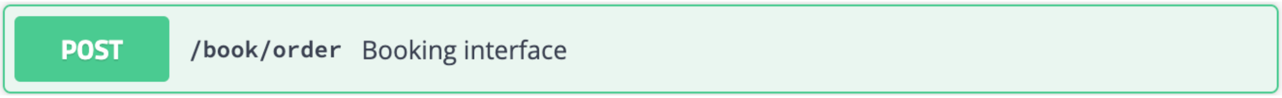


GET `/car/query/models` Query all car models and stock

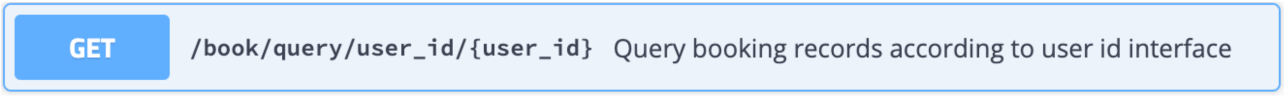
4、Query car details
Users can query all cars details of the specified car model in stock.

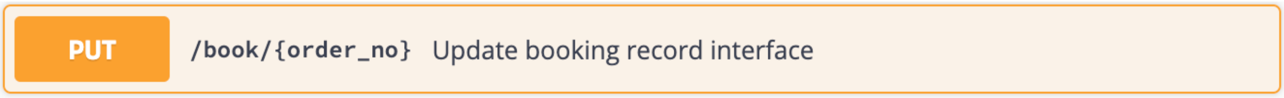

GET /car/query/stock/{model} Query all car in stock according to model

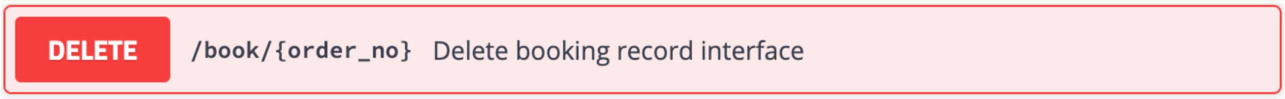
5、Booking a car
Users can select a car from the list of cars that can be reserved for reservation. When the user booked a car, the booking record status and booked car status should be updated.


POST /book/order Booking interface

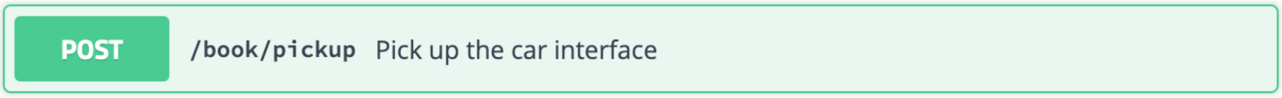
6、Booking records query, modification and cancel
Users can query booking records according to user id, and modify or cancel the booking record. When the user cancel the booking, the booking record should be deleted and the booked car status should be updated.


GET /book/query/user_id/{user_id} Query booking records according to user id interface

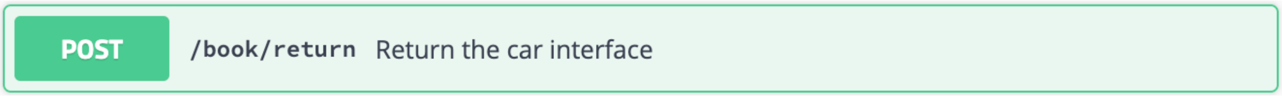

PUT /book/{order_no} Update booking record interface


DELETE /book/{order_no} Delete booking record interface

7、Pick up the car
When the user pick up the car, the booking record status and booked car status should be updated.


POST /book/pickup Pick up the car interface

8、Return the car
When the user return the car, the booking record status and booked car status should be updated.


POST /book/return Return the car interface

Background technology selection

- SpringBoot
- Mybatis

Development environment

- Programming language: Java
- Development tools: IDEA, GIT
- Project construction: Maven
- Deployment server: Alibaba cloud server ECS
- Database: H2 Memory database
- Code hosting platform: GitHub https://github.com/CherBao/carrental
- Swagger address: http://47.103.206.41:8080/swagger-ui.html#/