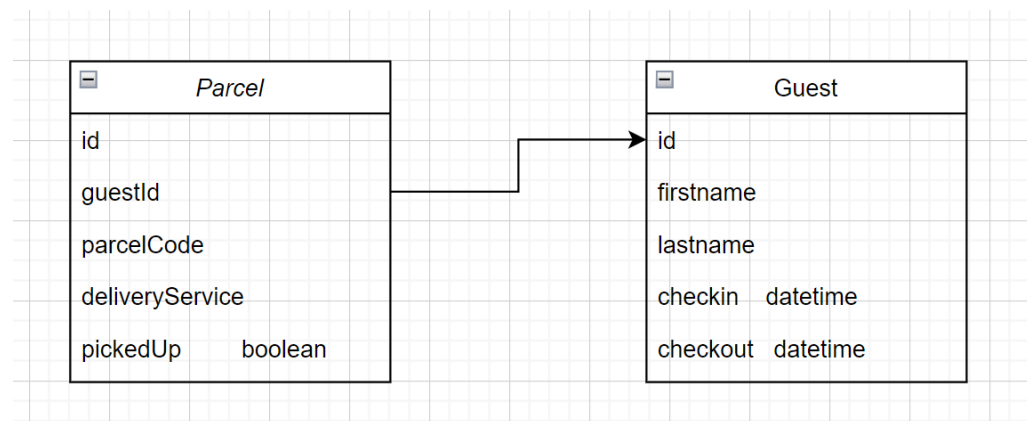


Data Model



Swagger Api

Guest service: ``http://localhost:8080/swagger-ui.html#/guest-controller``

Parcel service: ``http://localhost:8081/swagger-ui.html#/parcel-controller``

```
@PostMapping
public String addParcel(@RequestBody ParcelDto parcel) {
    if (!guestClient.checkIfCanAcceptParcel(parcel.getGuestId())) {
        return "Parcel receiver is not a guest or the guest has already checked out. Can not accept the parcel.";
    }
    parcelService.addParcel(parcel);
    return "Accept.";
}
```

```
no usages  DESKTOP-67ABDM2\win <r>
@PostMapping("/{lastname}/{firstname}")
public String takeParcelForGuest(@Valid @RequestBody ParcelRequest parcelRequest) {
    Guest guest = guestService.getGuestByName(parcelRequest.getLastName(), parcelRequest.getFirstName());
    if (guest == null)
        return "Parcel receiver is not a guest!";
    if (guest.getCheckOut() == null) {
        ParcelDto parcelDto = new ParcelDto(guest.getId(), parcelRequest.getParcelCode(),
            parcelRequest.getDeliveryService());
        parcelClient.storeParcelForGuest(parcelDto);
        return "Guest is still checked in! Parcel was store";
    }
    return "Guest has already checked out on " + guest.getCheckOut();
}
```

For storing parcel, need to check the guest status first. We will accept the parcel when the guest is checked in and have not checked out.

```

@PutMapping("/{id}/checkout")
public String checkoutGuest(@PathVariable long id) {
    Guest guest = guestService.getGuestById(id);
    if (guest == null) {
        return "This guest can not be found in our system.";
    }
    if (guest.getCheckOut() != null) {
        return "Guest has already checked out on " + guest.getCheckOut();
    }
    guestService.checkoutGuest(id);
    List<ParcelDto> parcelDtos = parcelClient.pickUpParcels(id);
    if (parcelDtos.isEmpty())
        return "No parcel for guest! Guest checkout is successful";

    return "Guest has " + parcelDtos.size() + " parcels stored here!";
}

```

For guests checking out, need to check if they have parcels stored.

Guest service API:

The screenshot shows a Swagger UI interface for a REST API. It lists endpoints for three controllers: guest-controller, operation-handler, and parcel-client. Each endpoint is represented by a colored bar indicating the HTTP method (GET, POST, PUT, or DELETE).

Controller	Method	Endpoint	Description
guest-controller Guest Controller	GET	/applicationName/guests	getGuests
	POST	/applicationName/guests	storeGuest
	GET	/applicationName/guests/{id}	getGuestById
	GET	/applicationName/guests/{id}/check	checkIfCanAcceptParcel
	PUT	/applicationName/guests/{id}/checkout	checkoutGuest
	GET	/applicationName/guests/{lastName}/{firstName}	getGuestByName
	POST	/applicationName/guests/{lastName}/{firstName}	takeParcelForGuest
operation-handler Operation Handler			
parcel-client Parcel Client	POST	/parcel-service/parcels	storeParcelForGuest
	GET	/parcel-service/parcels/{guestId}	pickUpParcels

Parcel service API:

localhost:8081/swagger-ui.html#/parcel-controller

翻譯

Gmail

YouTube

地圖

Api Documentation

[Terms of service](#)

[Apache 2.0](#)

basic-error-controller

Basic Error Controller

>

guest-client

Guest Client

>

operation-handler

Operation Handler

>

parcel-controller

Parcel Controller

▼

GET

/applicationName/parcels

getAllParcels

POST

/applicationName/parcels

addParcel

GET

/applicationName/parcels/{guestId}

pickUpParcelsByGuest

service-instance-rest-controller

Service Instance Rest Controller

>

web-mvc-links-handler

Web Mvc Links Handler

>