

Hotel Parcel Tracking Service: Backend Usability and Design Overview

Project Status

Please note that this project is a prototype and was created for demonstration purposes as part of a coding challenge. While it illustrates my skills in software development, it is not intended for actual use in a production environment and may require further development to meet industry standards.

The coding challenge requirements

Scenario

Receptionists have to accomplish various tasks throughout the day when operating a hotel. One of those tasks is to accept parcels for guests.

- It happens, that receptionists accept parcels for guests, that was already checked out of the hotel.
- It happens from time to time, that guests forget to pick up their parcels before check-out and even forget it then.

Code challenge: backend part

Requirements

1. The receptionist needs a tracking tool that knows to any given time which guests are checked into the hotel and have not checked out, to allow the receptionist to decide whether to accept the parcel or not.
2. The tracking tool should further allow the receptionist to check for parcels available for pick-up when the guest is checking out.
3. Use JAVA 8 & SpringBoot
4. Don't forget logging
5. Unit tests are a must, isn't it?

Tasks

- Document the usability and the design of your solution in a PDF file.
- Build a RESTful JSON-API that supports the following use cases. Your implementation should be based on the Microservices architecture approach.
- Document your solution with a Swagger file served by the API.

Deliverable

Please provide your solution on GitHub repo or send a zip file named to us that contains: `firstname- lastname.zip` to us that contains:

- The source code (buildable and runnable without changes) with Maven or Gradle build script

- An executable Jar
 - Instructions to run the solution are documented in a readme.md
- When shipping your solution, please let us know how long you took and where you did it. Don't artificially decrease the development time you tell us - we have good statistical measures and taking longer is not a bad thing at all.

In Scope

We're interested to see how you produce production-ready code. By that, we focus on naming conventions, coding style, sensible design, and meaningful commenting. Perfect code ships at a shape, that enables colleagues to quickly understand the domain and codebase.

We understand writing even a trivial application and having it production-ready is a lot of work. Please limit writing a lot of boilerplates - showing the intent and leaving "to-do" comments in your code is okay. However, make sure your solution works.

Out of Scope

- We don't expect you to use every design pattern you've ever heard of - only apply patterns when it makes sense to do so.
- We don't expect you to build a user interface.
- We're not expecting you to have optimized the solution for performance or memory size - readability is more important than performance.

Introduction

This document provides an overview of the backend system for the Hotel Parcel Tracking Service. It outlines the system's design and usability features, ensuring efficient parcel management for hotel guests and staff.

System Overview

The backend system is designed to facilitate the tracking, registration, and management of parcels for hotel guests. It provides a user-friendly interface for hotel staff to update parcel statuses and search parcel for guests.

Architecture Design

The backend is built on a microservices architecture, ensuring scalability and flexibility. It utilizes Spring Boot for rapid development and easy deployment.

API Documentation

The RESTful API offers endpoints for parcel operations (e.g., `/parcels/register``, `/parcels/pickup``) and guest inquiries (e.g., `/guests/{id}/checked-in``). It uses best practices in REST API design for consistency and maintainability. The Swagger API is accessible after starting the service at `/swagger-ui.html` (e.g., `http://localhost:8080/swagger-ui.html`)

Database Design

A relational database schema is used, with tables for ``Guests``, ``Parcels``.

Error Handling

The system provides clear, actionable error messages. All exceptions are logged, and an alert system notifies administrators of critical issues.

Usability Features

- **Intuitive API Endpoints**: Designed for clarity and ease of use, allowing for straightforward integration with front-end systems.
- **Real-time Tracking**: Staff can track parcels in real-time, enhancing transparency and trust.

Usage

Package the application into a JAR file:

```
```
```

```
mvn clean package
```

```
```
```

Run the application using the following command:

```
```
```

```
mvn spring-boot:run
```

```
```
```

Conclusion

The backend system for the Hotel Parcel Tracking Service is designed with usability and robustness in mind. It provides a secure, reliable, and user-friendly platform for managing the parcel delivery process in a hotel environment.