Fontys University of Applied Sciences

Data requirements

GetawayGo

Anna Kadurina

01/12/2024

Table of Contents

Introduction 1
Goal 1
User Service 1
Property Service 2
Booking Service 2
Review Service 2
Chat Service 2
Notification Service 3
Analytics Service 3
Conclusion 3

Introduction

This document defines the data requirements for GetawayGo, a web application that facilitates booking of short-term accommodations globally. It outlines the necessary data types, data flows, and storage requirements to ensure the proper functioning of the platform. All databases in Azure are with managed identities for security and in West Europe region.

Goal

The goal of this document is to define the essential data requirements for each microservice in the GetawayGo platform.

User Service

The user service manages user profiles, authentication, and user roles within the platform.

- Data type: User personal information (name, email, phone number, password – hashed, role)
- Storage: SQL database (Azure SQL server and database)
- Justification: SQL databases are ideal for storing structured and relational data, which ensures consistency in user information and authentication

Property Service

The property service handles accommodation listings, property details, pricing, etc.

- Data type**:** Property details (address, name, type, description, host info), pricing, availability dates, media links.
- Storage**:** SQL Database (e.g., Azure SQL server and database)
- Justification**:** An SQL database allows for structured storage of property details with references to users (hosts) and enables efficient querying of availability and pricing

Booking Service

The booking service manages reservations made by users, booking details.

- Data type**:** Booking details (check-in/check-out dates, guests, status), transactional data (amount).
- Storage**:** SQL Database (e.g., Azure SQL server and database)
- Justification**:** SQL storage ensures transactional integrity for bookings, with strong consistency for booking confirmations, and cancellations

Review Service

The review service manages user-generated reviews, ratings, and comments for properties.

- Data type**:** Reviews, ratings, comments, timestamps, user and property associations.
- Storage**:** SQL Database (e.g., Azure SQL server and database)
- Justification**:** SQL databases provide efficient mechanisms for querying reviews by property or user, ensuring consistency in the review process

Chat Service

The chat service handles messaging between users (guests and hosts) within the platform.

- Data type**:** Message content, timestamps, user associations, chat status (read/unread), system notifications.
- Storage**:** NoSQL Database (e.g., MongoDB, Cosmos DB)

- Justification: A NoSQL database is suitable for chat services because it allows for flexible, unstructured data storage, where message content and metadata can vary in size and complexity. NoSQL databases also scale horizontally, making them suitable for real-time messaging and large volumes of user interactions.

Notification Service

The notification service manages notifications sent to users regarding booking updates, etc.

- Data type: Notification content, user associations, delivery status, notification timestamps.
- Storage: SQL Database (e.g., Azure SQL, PostgreSQL)
- Justification: SQL storage is appropriate for notification services because it ensures reliable storage of notification status (e.g., delivered, failed) and enables querying based on user preferences and notification history.

Analytics Service

The analytics service will provide meaningful charts for hosts and administrators.

Storage: No dedicated database; queries data from other microservices' databases (User, Property, Booking, etc.)

Justification: Due to cost considerations and the nature of analytics, the Analytics Service will not have a dedicated database. Instead, it will query data from other microservices' databases. This approach reduces infrastructure costs while allowing the analytics service to access the necessary data for trend analysis and reporting. The service will perform read-heavy operations, so the use of existing relational and NoSQL databases is sufficient for its requirements.

Conclusion

This document outlines the data requirements for GetawayGo’s microservices, detailing the storage needs for user, property, booking, review, chat, analytics, and notification data. By selecting appropriate databases—primarily SQL for structured data and NoSQL for flexible, scalable storage—I ensure efficient data management, performance, and scalability.