Hbnb project documentation

Introduction

The HBNB project is a comprehensive web application designed to manage and display various places, amenities, and user interactions. It follows a structured three-layer architecture, ensuring modularity, scalability, and maintainability. To illustrate the system's design and functionality, three key diagrams have been created:

- Package Diagram: Demonstrates the organization of the system into distinct layers, including the Presentation Layer, Business Layer, and Persistence Layer, each handling specific responsibilities.
- Class Diagram: Defines the core entities of the system, their attributes, relationships, and behaviors, providing a clear representation of the application's data model.
- Sequence Diagram: Showcases the flow of interactions between different components, depicting how various processes and requests are handled within the system.

These diagrams collectively offer a detailed view of HBNB’s architecture, ensuring a well-structured, efficient, and easily maintainable application.

Package diagram

This package diagram represents a three-layered architecture, ensuring a clear separation of concerns between different components of the system. The Presentation Layer contains the User Interface package, which manages user interactions and displays data. The Business Layer handles core application logic and consists of four key packages: User, Place, Review, and Amenity, representing different functional entities within the system. Finally, the Persistence Layer includes Data Access and Service Agents, responsible for database operations and external service interactions. The layers interact sequentially, where the Presentation Layer communicates with the Business Layer, which in turn relies on the Persistence Layer for data management. This structured approach improves maintainability, scalability, and modularity, making it easier to manage and extend the system over time.

API Interaction Flow

This section explains the interactions between various components in the HBnB application for different API calls. It illustrates how the Presentation Layer (Services, API), Business Logic Layer (Models), and Persistence Layer (Database) communicate to handle user requests.

Sequence Diagrams for API Calls

1. **User Registration**:
    - **Purpose**: Visualize user registration process.
    - **Key Components**: User, API, BusinessLogic, Database.
    - **Design Decisions**: Validation before storing data ensures integrity.
2. **Place Creation**:
    - **Purpose**: Illustrate the creation of a place listing.
    - **Key Components**: User, API, BusinessLogic, Database.
    - **Design Decisions**: Validate place data before saving.
3. **Review Submission**:
    - **Purpose**: Show the review submission process.
    - **Key Components**: User, API, BusinessLogic, Database.
    - **Design Decisions**: Validate reviews before storage.
4. **Fetching a List of Places**:
    - **Purpose**: Depict the process of fetching a list of places.
    - **Key Components**: User, API, BusinessLogic, Database.
    - **Design Decisions**: Validate search criteria for accurate results.

Conclusion

This document provides a detailed blueprint of the interaction flow for key API calls within the HBnB application. The sequence diagrams show the step-by-step process of handling user requests, ensuring data integrity, and maintaining consistent communication between layers. This documentation serves as a reference for developers and stakeholders to understand the system's architecture and design, guiding implementation effectively.

Class diagram

A Python class diagram is a visual representation of the structure of an object-oriented program3 . It describes classes, their attributes, methods and relationships13. This type of diagram is part of the Unified Modeling Language (UML).

The user class is a user of the platform. A user can be a host, a guest or an administrator, and can interact with various elements of the system, such as profile information and reservations.

User contains the following Attributes

- firstName :
- lastName :
- email :
- password
- isAdmin

User also contains Methods:

- +register()
- +updateProfile()
- +delete()

The **Place** class represents a place or location that the user can choose in the system. This place is a physical space that can be reserved, described and updated by users (for example, an apartment, a house, a room, etc.). It contains detailed information to describe the place and allows you to manage the creation, updating, deletion and display of these places.

Place contains the following Attributes**:**

- +String title
- +String description
- +Float price
- +Float latitude
- +Float longitude

Place contains Methods :

- create() : crée un nouveau **Place** dans le système.
- update() : met à jour les informations d'un **Place** existant.
- delete() : supprime un P**lace**.
- list() : liste tous les **Place** disponibles.

The **Review** class represents all the comments and ratings that users leave after staying at a place (or Place in the context of Airbnb). This class allows users to share their impressions and experiences, and rate the place according to various criteria

Review contains the following Attributes **:**

Attributs :

- +int rating
- +string comment

Review contains Methods **:**

- create()
- update()
- delete()
- listByPlace()

The **Amenity** class is actually used to describe the equipment and features available in a dwelling or location, such as furniture, appliances, or other amenities important to the user's stay..

Amenity contains the following Attributes**:**

- name : String
- description : String

Amenity contains Methods **:**

- create()
- update()
- delete()
- list()

IN CONCLUSION! In an accommodation rental system, the user chooses a place (or Place) to rent, such as an apartment, house or room. After their stay, users can leave a review of the place, giving it a rating and writing a comment about their experience. Each location also has amenities, which describe

the equipment and services available in the accommodation, such as furniture, appliances (washing machine, Wi-Fi, air conditioning, etc.). These amenities are directly linked to the location, as they form part of it and influence the user's experience.Thus, the user is connected to the location he or she is booking, can leave a review of the location after his or her stay, and the location itself contains amenities that make it attractive for booking.