

Hotelmanager

Der Hotelmanager ist eine Anwendung, die die Verwaltung des Hotelzimmers ermöglicht. Die Daten werden in einer Datenbank gespeichert und über eine API mit verschiedenen Operationen anbinden.

Einsteig

Mit diesen Anweisungen erhalten Sie eine Kopie des Projekts, das zu Entwicklungs- und Testzwecken auf Ihrem lokalen Computer ausgeführt werden kann. Hinweise zum Bereitstellen des Projekts auf einem Live-System finden Sie unter Bereitstellung.

Voraussetzungen

Welche Dinge benötigen Sie, um die Software zu installieren und wie installieren Sie sie?

JDK 11

Maven

H2

Verwendung

- Starten das Maven-Projekt
- Oder mvn spring-boot:run
- Beim Hochfahren der Anwendung werden drei Hotelzimmer mit beliebiger Zimmernummer angelegt
 - a. Eines besitzt eine Minibar und ist ein Doppelzimmer
 - b. Eines besitzt eine Minibar und ist ein Einzelzimmer
 - c. Eines besitzt keine Minibar und ist eine Suite
- Die Daten werden in H2 gespeichert.
 - o Login to H2-DB:
 - <http://localhost:8080/h2-console>
 - JDBC URL : jdbc:h2:file:./data/hotel
 - Username: hotel
 - Password: password

URLs relativ zu <http://localhost:8080/api>. Das Program hat folgende Endpunkte;

Methode	URL	Beschreibung
POST	/rooms	Erstellen ein neues Hotelzimmer. Das neues erstelltes Hotelzimmer wird zurückgegeben. (Die Raumerstellungszeit wird automatisch generiert, wenn die Zeit nicht eingegeben wird.)
GET	/rooms	Gibt eine Liste aller Hotelzimmern zurück.
GET	/rooms/id	Geben Sie ein einzelnes Hotelzimmer mit angegebenem Ausweis zurück. Wenn es kein Hotelzimmer mit übereinstimmender ID gibt, wird nichts(notFoundException) zurückgegeben.
PUT	/rooms/id	Aktualisieren Sie ein Hotelzimmer mit der angegebenen ID. Das neu aktualisierte Hotelzimmer wird zurückgegeben. Wenn es kein Hotelzimmer mit übereinstimmender ID gibt, wird nichts(notFoundException) zurückgegeben.
DELETE	/rooms/id	Löschen Sie ein Hotelzimmer mit der angegebenen ID. Das gelöschte Hotelzimmer wird zurückgegeben. Wenn es kein Hotelzimmer mit übereinstimmender ID gibt, wird nichts(notFoundException) zurückgegeben.

Methode	
POST	<ul style="list-style-type: none"> • URL http://localhost:8080/api/rooms • Data Params : { "size": "SINGLE", "minibar": false } • Success Response: <ul style="list-style-type: none"> ◦ Code: 201 CREATED Content: { "id": 4, "size": "SINGLE", "minibar": false, "created": "2021-03-02T19:50:40.333+00:00" }
GET	<ul style="list-style-type: none"> • URL http://localhost:8080/api/rooms • Data Params • Success Response: <ul style="list-style-type: none"> ◦ Code: 200 OK Content:

	<pre>[{ "id": 1, "size": "DOUBLE", "minibar": true }, { "id": 2, "size": "SINGLE", "minibar": true }, { "id": 3, "size": "SUITE", "minibar": false }]</pre>
GET	<ul style="list-style-type: none"> • URL http://localhost:8080/api/rooms/1 • Data Params • Success Response: Code: 200 OK Content: { "id": 1, "size": "DOUBLE", "minibar": true }
GET	<ul style="list-style-type: none"> • URL http://localhost:8080/api/rooms/100 • Data Params • Error Response: Code: 404 Not Found Content: <pre>{ "timestamp": "2021-03-02T20:05:52.213+00:00", "status": 404, "error": "Not Found", "message": "", "path": "/api/rooms/100" }</pre>
PUT	<ul style="list-style-type: none"> • URL http://localhost:8080/api/rooms/1 • Data Params { "size": "DOUBLE", "minibar": false } • Success Response: Code: 200 OK Content: { "id": 1, "size": "DOUBLE", "minibar": false }
PUT	<ul style="list-style-type: none"> • URL http://localhost:8080/api/rooms/1 • Data Params { "size": "gfgfgf", "minibar": false }

	<ul style="list-style-type: none"> Error Response: <p>Code: 400 Bad Request</p> <p>Content:</p> <pre>{ "timestamp": "2021-03-02T20:16:33.314+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/rooms/1" }</pre>
DELETE	<ul style="list-style-type: none"> URL http://localhost:8080/api/rooms/1 Data Params Success Response: Code: 200 OK Content: { "id": 1, "size": "DOUBLE", "minibar": false }
DELETE	<ul style="list-style-type: none"> URL http://localhost:8080/api/rooms/99 Data Params Success Response: <p>Code: 404 Not Found</p> <p>Content: {</p> <pre> "timestamp": "2021-03-02T20:19:33.062+00:00", "status": 404, "error": "Not Found", "message": "", "path": "/api/rooms/100" }</pre>