

Backend Tech Challenge - II

Underwater volcano creates new island

Gaurav

www.nydailynews.com/news/world/underwater-volcano-reservation-a-rest-api-away

TRENDING: CITIZENSHIP NOW CONGRESSIONAL BASEBALL SHOOTING GOLDEN STATE WARRIORS BILL COSBY OPIOID NATION

SEARCH SECTIONS

DAILY NEWS | NEWS

f t i v

SUBSCRIBE

Crime U.S. World Politics

Underwater volcanic eruption forms island in Pacific Ocean; Booking just a REST API away

f t i



Pleiades Satellite Image - Image: Tonga-Mungu-Harison, Tonga © 2015 CNES / Distribution: Airbus DS

A new island formed from an underwater volcano in the Pacific Ocean. (PLEIADES © CNES 2015 -DISTRIBUTION: AIRBUS DEFENCE AND)

BY
GORDON FREEMAN

NEW YORK DAILY NEWS
Monday, June 12, 2017, 3:10 PM

An underwater volcano formed a new island in the Pacific Ocean last month.

Satellite footage from Jun. 15 shows an island formed due to an eruption from the Lorem Ipsum underwater volcano, [reports the Earth Observation Portal](#).

The conditions on the island seems just perfect and travellers will be delighted to know that it will be opened up for the general public soon. You will be now able to book a night or two on the pristine uncharted island by making simple REST calls. The API is currently being developed and more details will be made available soon.

An underwater volcano formed a new small island in the Pacific Ocean last month. All the conditions on the island seems perfect and it was decided to open it up for the general public to experience the pristine uncharted territory. The island is big enough to host a single campsite so everybody is very excited to visit. In order to regulate the number of people on the island, it was decided to come up with an online web application to manage the reservations. You are responsible for design and development of a REST API service that will manage the campsite reservations.

To streamline the reservations a few constraints need to be in place -

- The campsite will be free for all.
- The campsite can be reserved for max 3 days.
- The campsite can be reserved minimum 1 day(s) ahead of arrival and up to 1 month in advance.
- Reservations can be cancelled anytime.
- For sake of simplicity assume the check-in & check-out time is 12:00 AM

System Requirements

- The users will need to find out when the campsite is available. So the system should expose an API to provide information of the availability of the campsite for a given date range with the default being 1 month.
- Provide an end point for reserving the campsite. The user will provide his/her email & full name at the time of reserving the campsite along with intended arrival date and departure date. Return a unique booking identifier back to the caller if the reservation is successful.
- The unique booking identifier can be used to modify or cancel the reservation later on. Provide appropriate end point(s) to allow modification/cancellation of an existing reservation
- Due to the popularity of the island, there is a high likelihood of multiple users attempting to reserve the campsite for the same/overlapping date(s). Demonstrate with appropriate test cases that the system can gracefully handle concurrent requests to reserve the campsite.
- Provide appropriate error messages to the caller to indicate the error cases.
- In general, the system should be able to handle large volume of requests for getting the campsite availability.
- There are no restrictions on how reservations are stored as long as system constraints are not violated.

Things to look out for in the solution

- How is the RESTful service designed and what technologies are used to solve the problem?
- Any design patterns used?
- How is the schema designed? Can it be support future use cases e.g. multiple campsites?
- How are the APIs designed? Do they follow best practices?
- How are the CRUD operations designed and handled in the above problem statement?
- How are the concurrent operations handled?
- Type of database used - Relational vs NoSQL?
 - In case of Relational DB - Optimistic Locking vs Pessimistic Locking?
- Transaction management
- Unit/Integration testing
- Business validations
- Caching considerations?