

# Candidate Task: Software Engineer - Backend

## Background

In architecture terminology, we have, among others, the following two terms:

- *Building limits*: The areas (polygons) on your site where you are allowed to build
- *Height plateaus*: Areas (polygons) on your site with different elevation. In reality, your building site is a continuous irregular terrain, but before building, you level your terrain into discrete plateaus with constant elevation.

We run a preprocessing step that splits the building limit into polygons corresponding to the height plateaus. These building limit polygons should have the elevation of the corresponding height plateau set as a property.

## Prerequisites

- Experience with backend-development is recommended.

## Task

Your task is to implement an programmatic API with a clean interface that consumes building limits and height plateaus, splits up the building limits according to the height plateaus, and stores these three entities (building limits, height plateaus and split building limits) in a persistent way.

## Requirements

### Validation

Although the height plateaus should completely cover the building limits, inaccuracies in the input may occur. Consider how you can validate the input/output and handle any inaccuracies such as gaps/holes between height plateaus, etc.

### Error handling

The API should give meaningful error messages when errors occur.

### Concurrency

Imagine that two users, Bob and Mary, make modifications to the same project. Bob does some modifications to the building limits while Mary changes the height plateaus. What happens if they call the API at the same time? Make sure the API deals with concurrent updates.

### Testing

Think about ways of testing your application to make sure it works as intended.

### Deployment

The API should be deployed to an optional cloud service provider.

You are free to choose any language and libraries to solve the task. Deliver your solution in a zip-file or a link to a GitHub repository. The solution should include a README-file.

**Example input**

JSON document with the following format:

```
{  
  "building_limits": <GeoJSON>,  
  "height_plateaus": <GeoJSON>,  
}
```