Backend Coding Challenge Accepted

This technical challenge will help us get an idea of how you would use your technical skills, knowledge and experience to provide backend infrastructure for our end users.

We really value your time and for us is very important, so this challenge has been designed with that in mind. There is no specific amount of time that this challenge should take you to complete, as the level of effort will depend on how familiar you are with backend development and whether or not you tackle the optional functionality. Please feel free to spend as much time as you would like building your preferred solution and we hope you have fun building it!

## Mission

First a little bit of context:

*Stolen bikes are a typical problem in big cities. The Police want to be more efficient in resolving stolen bike cases. They decided to build a software that can automate their processes — the software that you're going to develop.*

Your mission is to create a backend infrastructure that provides APIs for a frontend application that satisfies all requirements below.

## Requirements

1. Bike owners can report a stolen bike.
2. A bike can have multiple characteristics: license number, color, type, full name of the owner, date, description of the theft, address where the bike was stolen, status of the case.
3. Police have multiple departments that are responsible for stolen bikes.
4. A department can have some amount of police officers who can work on stolen bike cases.
5. The Police can scale their number of departments, and can increase the number of police officers per department.
6. Each police officer should be able to search bikes by different characteristics in a database and see which department is responsible for a stolen bike case.
7. New stolen bike cases should be automatically assigned to any free police officer in any department.
8. A police officer can only handle one stolen bike case at a time.
9. When the Police find a bike, the case is marked as resolved and the responsible police officer becomes available to take a new stolen bike case.
10. The system should be able to assign unassigned stolen bike cases automatically when a police officer becomes available.
11. Include instructions for building and testing the backend and any other relevant information it needs to have in consideration in a README.md file.
12. Please upload your code to Github, Bitbucket or Gitlab.

## Ideas

The ideas listed below are all optional and you should feel free to implement either any/all of these or other from your preferences as time permits.

Don’t be shy and let us wow with your application!

1. When a stolen bike is registered, save latitude and longitude coordinates based on the address information.
2. Implementing sign in/sign up process and [RBAC](https://en.wikipedia.org/wiki/Role-based_access_control) (Role Based Access Control) for: Bike Owners, Police Officers, Police Directors Departments. Assign the rules permission based on your own criteria.
3. Bike owners can check the status by introducing the record id associated to the case.
4. Send a notification to the bike owner via email when their status case changes.

Use any library, framework, database engine or tool that fits your technical needs.

## Bonus

1. A simple React or Vue frontend application, and/or Express-powered webpage, offering a visualization of all or part of the data utilizing the API you have built as a backend.
2. Unit and integration tests
3. APIs documentation with a visual tool leitehr [Slate](https://github.com/slatedocs/slate), [Swagger](https://swagger.io/), [ApiDoc](https://apidocjs.com/) or any other of your preference,

## 

## Resources

To get latitude and longitude coordinates, you may use any solution you prefer. We suggest, for convenience, the [Google Geocoding API](https://developers.google.com/maps/documentation/geocoding/overview). It has acceptable free tier limits and is well-documented.

For sending emails notifications, you may also use any solution you prefer. We suggest [SendGrid](https://sendgrid.com/docs/api-reference/), since they also have an acceptable free tier limit and is well-documented.

## Considerations

Among other criterias, your submission will be evaluated on:

1. Implementation of the stated requirements.
2. Backend Architecture.
3. The general quality of the code and it’s resistance to crashing.
4. Your use of Javascript coding conventions.
5. Knowledge and usage of Javascript/NodeJS libraries and SDKs.
6. Clarity of communications in comments and other documentation.
7. Efficient DB queries
8. This isn't just about handling the happy path. Think about concurrency, security, scalability.
9. Make your API public. Deploy it using the service of your choice (i.e. AWS, Heroku, Digital Ocean)
10. Coding challenge reviewers should also be able to load your project and run it easily in a local environment. If the reviewer needs to do any project configuration (i.e. add his/her own Google / SendGrid API keys, install local db engines) and can’t just do *npm install* or similar, that is enough reason to reject your solution.
11. We appreciate there's a lot that is asked in this exercise. If you need more time, feel free to ask. If you need to de-prioritize something, apply the same judgement you would on a professional project, argument your decision.

Oh, and don’t worry if your solution has personality or a sense of humor. We want you to have fun!

Good Luck.