



Platform Engineering Coding Exercise

Background

This is a coding exercise used by Deloitte Platform Engineering as part of our candidate review process. Our roles are very *hands-on* and we value people with great technical skills and good practices, so we invite you to code-to-impress!

Scenario

An organisation wants to develop a new application that requires users to create their own credentials by providing username and password. You have been tasked to create a **service layer class library** that will be used in this new application.

Task 1 (Required)

1. **Create a service that calculates and returns the password strength.** To calculate the password strength, feel free to suggest the best way to do it and return a value to be used by the application.
2. **Create a console app to interact with your service layer.** Create a console app that allows entering a password and displays the password strength by calling the service described above.
3. **Create a Test project and implement test cases for your service.**

Task 2 (Optional. Not required but you would have extra fun and points if delivered)

4. **Create a service that checks whether the password has appeared in data breaches, and provide the number of times the password has appeared in those breaches.** To check if the password has appeared in data breaches, you need to use the **Have I Been Pwned API v2** described here <https://haveibeenpwned.com/API/v2/> and here <https://www.troyhunt.com/ive-just-launched-pwned-passwords-version-2/>. **IMPORTANT:** You need to **protect the password being searched for**, by using the k-anonymity model provided by the API. So the password **MUST NOT** be sent to the API in plain text or in a hashed string.
5. **Update the console app described in the previous task to interact with your service layer.** Now the console app must also return the number of times the password has appeared in data breaches.
6. **Update the Test project described in the previous task to implement test cases for your service.**

How to deliver your solution

1. Create your solution using **.NET Core**, preferably C#.
2. (Optionally) Add a README.md to your solution if you think it might help.
3. Create a **PRIVATE** repository on GitHub.
4. Invite the [@platformenghireme](#) GitHub handle to collaborate. This will allow us to access your private repository and review your code.
5. You have **1 week** to upload the solution to the repo and invite us to collaborate. However, feel free to invite us to your repo as soon as you have it ready.
6. Only submit your solution when a) the repository can be cloned (`git clone`), b) the solution builds and runs as described above, and c) if you have included tests cases, all of them pass. If any of these requirements are not met, your solution won't be considered.
7. Once you have invited us to your private repo, send an email to dahlhaus@deloitte.com.au including the link to the **private repo** to let us know that you have finished your solution. If you don't send an email with the link, your solution won't be assigned to your application.
8. We will then review your solution and contact you with the next steps.

Notes

- Please **DO NOT** include any (of your) real passwords on the code checked into the repository or your emails.
- The solution **MUST be original and created by the candidate**.
- **DO NOT share** this exercise with anyone else.
- If your submission is successful, we will discuss your solution during the technical interview.