Purplebricks Code Test

Thanks for taking part in the Purplebricks code test.

Please return the test within 7 days. Purplebricks are a company with a flexible working culture so we understand this may not always be possible. Feel free to talk to us if you want to complete the test on a different schedule.

The problem

For the purposes of this test we have intentionally created a .NET solution that is difficult to maintain and has several issues.

We would like you to begin refactoring the “BookViewing” method in the “ViewingService” class.

You can change anything **except** for the following:

- The signature of the “BookViewing” method in the “ViewingService”.
- Anything in the “LegacyClient” project – this is an existing integration.
- Anything in the “PB.ServiceBus” project – this is a shared dependency.

Business Rules

Note: Definitions can be found at the end of this document.

Customers of Purplebricks can book a viewing for a property online. This is managed by a simple diary system.

If the customer requests a timeslot that **is** available, then they will automatically get booked in.

If the customer requests a timeslot that **is not** available, then their booking will be requested.

- A customer can only book a viewing for an advert that is on the market.
- Advert details are retrieved via a restful web service:
    - If the “FeatureToggle.UseNewAdvertApi” configuration is set to **true**, then use the new Advert API.
    - If the “FeatureToggle.UseNewAdvertApi” configuration is set to **false** or is missing, then use the LegacyAdvertApi.
- If the seller has paid for **Accompanied Viewings,** then the **Accompanied Viewings Local Property Expert (AVLPE)** will conduct/host the viewing.
- If the seller has not paid for **Accompanied Viewings,** then the seller will conduct the viewing themselves.
- When a viewing is booked, then the `ViewingBookedV1` event should be published to the service bus
- When a viewing slot is unavailable, then a `ViewingRequestedV1` event should be published to the service bus

You can use whichever test and/or mocking frameworks you wish. We’re not expecting the work to be ‘finished’, but you should have started the work and be able to discuss the changes you have made and what your next steps would be. Make sure you attempt some refactoring and unit testing to demonstrate your skills in both.

Your solution

Please spend no more than 2 hours improving the solution. We do not anticipate you to have time to fix every issue.

We will be looking for:

- Code that is more maintainable
- Code that is clean, simple and readable
- Use of any engineering principles that you think are relevant. (e.g. SOLID, DRY, YAGNI, etc.)
- We don't want over-engineered code (KISS)
- We aren't looking for lots of comments, but you can add them in if you feel you need to add a narration to something

You are welcome to use the Internet to look anything up.

Please zip up your solution, ensuring that it builds and the unit tests pass.

Next Steps

Several Purplebricks developers will review your solution and provide feedback.

If we feel your solution meets our criteria, will get in touch to arrange a face-to-face interview. As part of the face-to-face we will expect you to explain the changes that you have made and what you would have done next if you had more time.

Definitions

Customer – The person who is looking to purchase a property.

Property – This is a physical building (typically a house). This is represented on the website/app as an advert.

Advert – This is the advertisement for a given property.

Viewing – When a customer views a property..

Seller – this is the person who is selling the property. They will typically do the viewings themselves unless they purchase the “Accompanied Viewings” addon

Accompanied Viewings Local Property Expert (AVLPE) – This is a Purplebricks representative who will conduct a viewing on behalf of a customer if the seller has purchased “Accompanied Viewings” addon.

Diary – Sellers and AVLPEs have diaries. A diary consists of several timeslots which show the seller or AVLPE’s availability. You can think of a diary like a calendar.