

Global Aerospace Pre-Interview Technical Exercise

Welcome.

IT recruitment at Global Aerospace takes the form of an off-site technical exercise followed by a telephone interview. Successful candidates will then be invited into the London office for another technical interview with an aptitude test and another short technical exercise. One more CV based interview with a member of HR and the IT Manager would follow.

This first step in the process involves a technical exercise. Hopefully it should not take long and will demonstrate your knowledge and familiarity with the following:

- WPF
- XAML
- Dependency Injection
- C#
- MVVM pattern
- RestSharp
- Linq

Please adhere to MVVM principles. No code behind. The exercise can be undertaken in any version of .Net and targeted at a x64 platform. This doesn't need to be polished, just good enough to demonstrate familiarity with the above. No need for a multi-project solution, a single project using folders to separate the Views, ViewModels, Models will suffice.

In terms of Nuget packages, you will need:

- RestSharp nuget,
- a dependency injection package (e.g. Unity),
- Newtonsoft.Json



Resources can be found here: <https://github.com/camos414/GlobalAerospaceTechnicalTest.git>

The Task

Take a look at the mp4 in the repo folder. This is what we are looking for. It's a contrived exercise but there are elements to it that will test a few key skills.

Functionality:

Most of the functionality is shown off in the video.

- The data is drawn from an API using RestSharp. The AirportDatabase.cs file (with interface, IAirportDatabase.cs) in the repo gives you that bit. Call the method GetByCountry with the parameter GB. Not all GB airports are returned in the free version of this API.
We have also provided the Model file (AirportModel.cs) which can be used to house the records deserialised from the API.
- There are two grids (datagrids) – airport records that are ticked in the grid on the left should appear in the grid on the right. Unticking removes them.
- Filter – the text box at the top should restrict the list – matching records are those where the text exists anywhere in the name of the airport. Use linq to achieve this.
- The two grids should have a detail section underneath them showing the details of the selected item.
- Use the Grid Splitter control to allow the user to stretch parts of the screen as shown in the video.
- In the top right is a combo box containing all the KnownColors. Selecting a different colour changes the colour of the textbox borders, the grid splitters, the alternate row on the grid. Attempt this by using styles in conjunction with an attached property.
tip: FrameworkPropertyMetadataOption “Inherits”.
- Use dependency injection to create the view models you use.
- Use DataTemplates to pair the view models and views.

On Completion

Kindly upload your solution to github repository of your own, make it public and email the url of the repository to Chris Amos camos@global-aero.com.

Do not hesitate to contact me should you need further information.

We've made the assumption that this exercise can be undertaken using the free version of Visual Studio. Kindly let us know if this is not the case and we'll attempt to find another way.

Thanks for agreeing to take undertake the exercise and good luck.