

Adra Developer Test

Instructions

- Complete the development task below for “Accounts balance viewer” scenario.
- UI and presentation can be minimal for application, Focus on the functionality.
- Unit tests can be covered only for couple of main flows and not focused for the overall coverage.
- **Duration for the Test is 1 week**, if completed early you can focus on Reporting, Unit tests coverage/ UI etc.
- Please read the email sent with the Developer Test for any further information.

Accounts balance viewer

Scenario

Account balances are updated every month in the Jondell Corp and the CFO of the company Dr. Sundt likes to have a close look at the overview. Since Dr. Sundt has no experience in software development, he would like to hire you to do his dirty laundry. So, your job is to create a simple web enabled account balance overview tool. There are 5 accounts in Jondell Corp and they are

1. R&D
2. Canteen
3. CEO's car expenses
4. Marketing
5. Parking fines

Balances of these accounts are updated every month and the format is as follows,

Account Balances for January

R&D	5.63
Canteen	50,000
CEO's car	10,000
Marketing	-600
Parking fines	2000

The file can be an excel file or a tab separated text file. The UI should have 2 pages. One to upload these files and one to display the balances in a straightforward view. The straightforward view has to be as follows.

Balances as of March 2017

<u>R&D</u>	<u>Canteen</u>	<u>CEO's car</u>	<u>Marketing</u>	<u>Parkings fines</u>
Rs 10.56	Rs. 98,000/=	Rs. 24,000/=	Rs. -19,112/=	Rs. 11,000/=

There are two types of users in the system. Admin users can only upload balances and view the reports pages (if any) while all the other users can view the current balances of each account.

Suggested technology stack

It is recommended to host the application in Azure. You can use a free 30-day subscription to do so. The code could be in bitbucket. The information must be persisted in a database and must be developed via a ORM framework. The client-server communication must happen via ASP.NET WEB API and the client should be powered by JS framework like Aurelia or Angular. Beyond this you can use any other technology as per your preference.

General guidelines

You are encouraged to use standard development practices when implementing the final product. Knowledge on modularization, decoupling etc. that you have experience in your career may come in handy. Since Dr. Sundt is a grumpy man, he needs this product delivered in 1 week. Couple of more things,

- Don't write code which is not needed
- Product and the unit tests need to be easily runnable
- Build tasks should run all unit tests as well

Some checklists you should double check before sending us your completed solution.

- If you are using a scaffolder/code generator (like Angular-cli or Yeoman generator), please remove all the unnecessary junk that you are not actively using in your solution.
- Make sure all your client side and server-side unit tests/ automation tests are working fine.
- If you are including a README file (We think you should), make sure all the steps you put in there are double checked and working correctly. If you need specific versions of dependencies that we need to install in our developer environment to test your solution (for e.g.: nodejs, bower, etc...) please include them as prerequisites in your README.
- Use as much Azure PaaS services as possible wherever you can.
- If you are using Angular, make sure all the ng* commands are working.
- Please include steps to test the unit tests / automation tests in dev environment.
- Finally, please test the solution in your developer environment and hosted environment for completeness before sending us the final solution.