

Zip Instalment Calculator Code Challenge: C# .Net

This URL has the .Net coding exercise

description: <https://github.com/quadpay/software-engineer-interview#the-pair-programming-interview>.

Though above URL mentions it as pair programming exercise, we are ok with you completing the code offline and then walking them thru the code during the discussion.

And the boilerplate code for the exercise is available

here: <https://github.com/quadpay/software-engineer-interview/tree/master/SoftwareEngineerInterview/CSharp>.

<https://github.com/quadpay/software-engineer-interview#background> provides high level expectations on the quality of code. Below section provide more details on the quality expectations.

Definition of Done

This solution should be what you consider to be production ready, so think carefully about what you want to include.

1. Code should be self-documented, use comments liberally. We want to see how well you communicate thru your code.
2. Proper exception handling for REST endpoints with correct standard error / status codes being returned from the API layer.
3. We would like to see how well you separate code and entities at different layers, like, API, Biz Logic and Repository layers. Use a clean folder structure to reflect various layers.
4. We would like to see what design patterns and/or libraries you want to use, where applicable.
5. Should have unit test cases as well as some integration test. Use mocked DB objects for unit tests.

6. Readme.md should provide instructions to run and test your code. Also document your APIs for other developers who are going to use your APIs. We would like to see how well you communicate thru your documents.
7. Follow best practices, like, logging, error handling.

NOTE: Please share your code using your personal Github account.

Points to take care for REST endpoints

1. Use Nouns for Resource Identification. Bad Practice: Do not use verbs
2. Use Plural Nouns to Name a Resource
3. Use Proper HTTP Headers for Serialization Formats
4. Use Sub-Resources for Relations
5. Use Proper HTTP Methods
6. HTTP Response Status Codes
7. Field Name Casing Convention
8. Restful API Versioning
9. Refer : <https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/>