**Digital Airstrike Coding Assessment**

At a minimum we’d like to see a web page that interacts with an API to deliver the functionality described below using a modern framework appropriate to your stack. Feel free to go beyond those requirements and really show off your talent but we don’t expect you to spend a lot of time on it.

You can submit your source code by granting us access to a private git repository. Free, private repositories are available on GitHub or BitBucket

We’d also like to be able to run your solution. Please include a note on how to run it. Feel free to deploy it (or not) and include a link to the running site.

**Some recommended technologies:**NET - C#/.NET Core, Java, ReactJs, Angular2, Node.js

**The Challenge**

Create a web page that will display the optimal coin combination (fewest number of coins) that fulfills a given dollar value. The web page should take a dollar value as input and interact with an API that computes and returns the optimal coin combination. The results of the API call should be displayed on the web page.

Assume the following coins: - Silver Dollar ($1.00) - Half Dollar (0.50) - Quarter (0.25) - Dime (0.10) - Nickel (0.05) - Penny (0.01)

Input: ​A dollar value (e.g. $1.56 or $12.85).

Output: ​A format of your choosing that indicates which coins and how many of each are required. An ideal output for the API is an object in JSON representation of the coins and a count for each coin.

For example, your solution should return something like the following for the indicated values:

$0.99​ => {

"silver-dollar": 0,

"half-dollar": 1,

"quarter": 1,

"dime": 2,

"nickel": 0,  
"penny": 4 }

$1.56​ => {

"silver-dollar": 1,

"half-dollar": 1,

"quarter": 0,

"dime": 0,

"nickel": 1,  
"penny": 1 }

$12.85​ => {

"silver-dollar": 12,

"half-dollar": 1,

"quarter": 1,

"dime": 1,

"nickel": 0,

"penny": 0 }

At a minimum the above examples should work.

Submissions will be judged on readability, code style, creativity and pragmatism. Bonus points for including unit tests, demonstrating client-side skills, and/or making a robust solution.