

The Challenge

We would like for you to build a basic web API that determines the winner of a round of poker. The app does not need to conform to any existing Quicken Loans/Rocket brand styles. Please follow the guidelines below.

Guidelines

A poker deck contains 52 cards. Each card has a suit of either clubs, diamonds, hearts, or spades. Each card also has a value of either 2 through 10, jack, queen, king, or ace. For scoring purposes card values are ordered as above, with 2 having the lowest and ace the highest value. The suit has no impact on value.

A poker hand belonging to a player (which need only be tracked as a name; names can be assumed to be unique) consists of five cards dealt from the deck. Poker hands are ranked by the following partial order from lowest to highest:

- **High Card:** Hands which do not fit any higher category are ranked by the value of their highest card. If the highest cards have the same value, the hands are ranked by the next highest, and so on.
- **Pair:** Two of the five cards in the hand have the same value. Hands which both contain a pair are ranked by the value of the cards forming the pair. If these values are the same, the hands are ranked by the values of the cards not forming the pair, in decreasing order.
- **Two Pairs:** The hand contains two different pairs. Hands which both contain two pairs are ranked by the value of their highest pair. Hands with the same highest pair are ranked by the value of their other pair. If these values are the same the hands are ranked by the value of the remaining card.
- **Three of a Kind:** Three of the cards in the hand have the same value. Hands which both contain three of a kind are ranked by the value of the three cards.
- **Straight:** Hand contains five cards with consecutive values. Hands which both contain a straight are ranked by their highest card.
- **Flush:** Hand contains five cards of the same suit. Hands which are both flushes are ranked using the rules for High Card.
- **Full House:** Three cards of the same value, with the remaining two cards forming a pair. Ranked by the value of the three cards.
- **Four of a Kind:** Four cards with the same value. Ranked by the value of the four cards.
- **Straight Flush:** Five cards of the same suit with consecutive values. Ranked by the highest card in the hand.

The program needs to evaluate a pair of poker hands and to indicate which, if either, has a higher rank. You should include the winner's name as part of the resulting output.

For example, if the hand belonging to "Angel" contains "2H", "3D", "5S", "9C", "KD", and the hand belonging to "Bodhi" contains "2C", "3H", "4S", "8C", "AH", then Angel has a High Card, and Bodhi also has a High Card, and Bodhi wins because their High Card is of higher rank than Angel's.

As another example, if the hand belonging to "Cacey" contains "2H", "3H", "4H", "5H", "6H", and "Dominique" contains the cards "2C", "2H", "3H", "3C", "KH", then Cacey has a Straight Flush, 6-high, and Dominique has Two Pairs, 3-high, and Cacey has the winning hand.

For simplicity's sake, we will assume that there are two players, and that each player is simply a name (String) that can be assumed to be guaranteed elsewhere to be unique. (You need not verify the names are unique, and if you want to use the names as unique identifiers that is fine.)

Requirements

- The endpoint should be a .NET Core Web API using C# and following RESTful principles. You may use any .NET libraries (Entity Framework, open source libraries, commercial libraries such as those from Infragistics or Telerik) so long as they are included with your final project or are installed as part of the build process.
- The code for this should be saved to GitHub or similar service (GitLab, BitBucket, etc).

Evaluation

Once you're happy with your work here, send the link to your repo to your recruiter. They will, in turn, pass that link on to the engineering team for review. From there, the team will evaluate the code based on the following:

- Feature Completeness: Did the submission meet or exceed the listed requirements.
- Future proofing: How well will submitted code grow as needs change or evolve?
- Maintainability: How clean or well organized is the submitted code.
- Code Quality: Does the code show a strong understanding of the technology?
- Creativity: Does the submission show novel solutions or unique design decisions?