

```
<body>
         6
                        <form>
         7
                            <div>
         8
                                <input id="Username" value="Your username">
         9
                                <br/>
                                <label for="credits">Redeem Credits:</label>
        10
                                <input id="credits" value="0">
        11
                            </div>
        12
        13
                            <div>
        14
                                <button id="sendButton">Send
        15
                            </div>
                        </form>
        16
        17
                        <div id="statusText"></div>
        18
                   </body>
        19
                   <script>
                       const urlString = window.location.href;
        20
        21
                        const url = new URL(urlString);
        22
                        const userId = url.searchParams.get("userId") || 0;
        23
        24
                        const sendButton = document.getElementById('sendButton');
        25
                        sendButton.addEventListener('click', (e) => {
        26
                                e.preventDefault();
        27
                                const credits = document.getElementById("credits").value;
        28
                                const username = document.getElementById("Username").value;
        29
                                fetch('/redeem', {
                                    method: 'POST',
        30
        31
                                    headers: {
        32
                                        'Content-Type': 'application/json',
        33
                                    },
        34
                                    body: JSON.stringify({
        35
                                        userId: userId,
                                        credits: credits,
        36
        37
                                        username: username
        38
                                    })
                                })
        39
        40
                                .then(response => response.json())
        41
                                .then(data => {
                                    document.getElementById('statusText').innerHTML = data.status;
        42
        43
                                })
        44
                                .catch((error) => {
        45
                                    document.getElementById('statusText').innerHTML = error.status;
                                });
        46
        47
                       });
        48
                   </script>
        49
               </html>
               (-)

    ✓ 118  redeem-app/server.js [□]

               @@ -0,0 +1,118 @@
       . . .
            + const express = require("express");
         1
         2
             + const app = express();
         3
             + app.use(express.json());
             + const port = 3000;
mgreiler (Pending)
                                                                                            Owner Author
     Flexible configuration is overrated, isn't it?
     (<del>(</del>()
      Reply...
            + app.use(express.static(__dirname + "/public"));
```

```
Problems-Snarky: Initial Commit without DB logic by mgreiler · Pull Request #4 · mgreiler/code-review-examples
         8
             + app.get("/", function (req, res) {
                 res.sendFile(__dirname + "/public/index.html");
         9
        10
             + });
        11
             + app.post("/redeem", function (req, res) {
        12
        13
                const credits = req.body.credits;
        14
                 const userId = req.body.userId;
        15
                 const username = req.body.username;
        16
                 console.log(`redeeming ${credits} credits for user: ${username} (${userId})`);
        17
        18
                 try {
        19
                   const hours = redeemCredits(credits, userId);
        20
                   res.send({
        21
                     status: `User ${username} redeemed ${credits} credits to get ${hours} hours.`,
        22
                   });
        23
                 } catch (err) {
        24
                   res.status(400).send({
        25
                     status: err,
        26
                   });
        27
        28
             + });
        29
             + app.listen(port, () => {
        30
                 console.log(`API listening at http://localhost:${port}`);
        31
        32
             + });
        33
        34
             + /**
        35
             + * The player can purchase gaming hours by redeeming
             + * credits. How many hours one credit is
        37
             + * worth depends on the level of the player.
        38
        39
             + function redeemCredits(credits, playerId) {
                 playerLevel = getPlayerLevel(playerId);
mgreiler (Pending)
                                                                                             Owner Author
     Seems like someone's been skipping the basics of JavaScript scoping. These variables are declared globally.
     (<del>0</del>)
      Reply...
        41
                 hours = convertCreditsToHours(playerLevel, credits);
        42
        43
                 redeemHoursToPlayerProfile(hours, credits, playerId);
        44
                 return hours;
        45
             + }
        46
        47
             + /**
        48
             + * Returns the level of the player.
        49
        50
             + function getPlayerLevel(playerId) {
        51
                 levelQuery = "SELECT playerLevel FROM players WHERE playerId = " + playerId;
    mgreiler (Pending)
                                                                                             Owner (Author)
     Looks like we're giving hackers a free pass with our SQL queries. Direct concatenation with user input is a no-
     go.
     (<u>U</u>
      Reply...
```

52

playerLevel = executeQuery(levelQuery);

```
return playerLevel;
55
    + }
56
57
58
    + * Adds the purchased hours to the players game hours.
59
60
    + function redeemHoursToPlayerProfile(hours, credits, playerId) {
        hourQuery = "SELECT hours FROM players WHERE playerId = " + playerId;
61
62
        oldHours = executeQuery(hourQuery);
63
        updateQuery = "Update players SET hours = " + (oldHours + hours) + " WHERE playerId = " +
64
      playerId;
65
66
        try {
          executeQuery(updateQuery);
67
68
        } catch (err) {
           throw new Error("Could not add hours.");
69
70
71
72
        chargeCreditsFromPlayer(credits, playerId);
73
    + }
74
75
    + /**
76
    + * Charges the player with the credits redeemed.
77
78
    + function chargeCreditsFromPlayer(credits, playerId) {
79
        creditQuery = "SELECT credits FROM players WHERE playerId = " + playerId;
80
        oldCredits = executeQuery(hourQuery);
```



mgreiler (Pending)

Owner Author

Error handling seems to be an afterthought here. Some try-catch blocks are present, but what's the strategy?





Reply...

```
81
82
        updateQuery = "Update players SET credits = " + (oldCredits - credits) + " WHERE playerId = " +
      playerId;
83
84
        try {
85
          executeQuery(updateQuery);
86
        } catch (err) {
87
          throw new Error("Could not charge credits: " + err);
88
    + }
89
90
91
92
    + * This method converts the credits to game hours. Gamers that
93
    + * have a lower gaming level get more game hours for their credits.
    + * For more advanced gamers, buying new credits is more expensive.
94
95
96
       * Players of level less than 3, get 3 times the hours of their credit
       * Players of level less than 8, get 1.5 times the hours of their credit
98
    + * Players of level higher than 8, get just the hours of the credit.
99
    + */
    + function convertCreditsToHours(playerLevel, credits) {
```



mgreiler (Pending)

Owner Author

Not even this functionality works. What about the player at level 3?



Reply...

```
101
          if (playerLevel < 3) {</pre>
102
           return 3 * credits;
103
         } else if (playerLevel > 3 && playerLevel <= 8) {</pre>
104
            return 1.5 * credits;
105
         } else {
106
            return 1 * credits;
107
          }
108
     + }
109
110
     + /** TODO: implement */
111
     + function getRandomInt(max) {
112
         return Math.floor(Math.random() * Math.floor(max));
113
     + }
114
115
     + /** TODO: implement */
116
     + function executeQuery() {
117
         return getRandomInt(13);
118
     + }
```