CSE 154: Web Programming Summer 2019

Week 9 (Optional) QuickCheck: Code Quality with MySQL/Express

Identify the differences and the trade-offs between Examples A and B implementing a `GET /question-status` endpoint in the wpldb. Most differences are related to code quality, but there is also a common (and subtle) bug in one example related to the Response.

Example A:

```
// Returns the status of the given qid, requiring an student uwid to validate.
// Required query parameters: qid and uwid
app.get("/question-status", async (req, res) => {
  res.type(“json”);
  let qid = req.query.qid;
  let uwid = req.query.uwid;
  res.type("text");
  if (qid && uwid) {
    try {
      // Verify that the UWID belongs to a student in the database (users table).
      let check = await veryifyUser(uwid);
      if (check) {
        let stat = await getQuestionStatus(qid);
        res.send(stat);
      }
    } catch (err) { res.status(400).send("Error: " + err); }
  }
  res.status(500).send("bad request");
});

/**
 * Verifies the user.
 */
async function verifyUser(uwid) {
  let db;
  try {
    db = await getDB();
    let query = "SELECT * FROM users WHERE uwid = ‟ + uwid;
    let results = await db.query(query);
    return results.length > 0;
  } catch (err) {
    if (db) {
      db.end();
    }
    throw error;
  }
}

/**
 * Gets the q status.
 */
async function getQuestionStatus(qid) {
  let db;
  try {
    db = await getDB();
    let query = "SELECT status FROM queue WHERE qid = ?";
    let status = await db.query(query, [qid]);
    db.end();
    return status[0].status;
  } catch (error) {
    if (db) {
      db.end();
    }
    throw error;
  }
}
```

Example B

```
// Returns the status of the given qid, requiring a student uwid to validate.
// Required query parameters: qid and uwid
app.get("/question-status", async (req, res) => {
  let qid = req.query.qid;
  let uwid = req.query.uwid;
  res.type("text");

  // Ensure that we have our parameters
  if (qid && uwid) {
    let db;
    try {
      db = await getDB();
      // Verify that the UWID belongs to a student in the database (users table).
      let isVerified = await verifyUser(db, uwid);
      if (isVerified) {
        let status = await getQuestionStatus(db, qid);
        db.end();
        res.send(status);
      }
    } catch (err) {
      if (db) {
        db.end();
      }
      // "Something went wrong on the server. Please try again later."
      res.status(500).send(SERVER_ERROR_MSG);
    }
  } else {
    res.status(400).send("Missing required uwid and/or qid parameters.");
  }
});

/**
 * Queries the database to verify that a given uwid is one belonging to a student
 * @param {String} uwid - A student uwid to verify.
 * @param {Object} db - the database connection to the wpldb database.
 * @return {Boolean} - true if the uwid is one belonging to a student in the users table, false otherwise.
 */
async function verifyUser(db, uwid) {
  let query = "SELECT * FROM users WHERE uwid = ?";
  let results = await db.query(query, [uwid]);
  return results.length > 0;
}

/**
 * Queries the database to get the status of a specific queue member
 * @param {String} qid - The qid of a queue entry.
 * @param {Object} db - the database connection to the wpldb database.
 * @return {String} - The status of the given queue member.
 */
async function getQuestionStatus(db, qid) {
  let query = "SELECT status FROM queue WHERE qid = ?";
  let results = await db.query(query, [qid]);
  return results[0].status;
}
```