

Requirements

Modify a solution (either yours or the instructors) to Assignment 06 to enhance the Jeopardy Questions Web API to authenticate the user by using Token Based Authentication, and to allow new questions to be inserted into the database tables.

1. To implement a simple Token Based Authentication:
 - a. Make use of two new columns which have been added to the Users table:
 - i. AuthToken, text datatype
 - ii. AuthTokenIssued, date and time
 - b. Change the *signin* endpoint so that when a user successfully signs in:
 - i. Generate a new unique authorization token using a strategy of your own choosing.
 - ii. Update the user's record in the Users table with this new AuthToken and set AuthTokenIssued to the current system date and time.
 - iii. Return this new AuthToken value to the client. E.g.

```
{
  message: "success",
  authToken: "xxxxxxxxxx"
}
```
 - c. For all endpoints (excluding the *signin* endpoint), require a valid AuthToken to be included in the request.
 - i. Pass the token in the query string as "auth=xxxxxxxxxx", where xxxxxxxxxxxx is the authorization token returned from the *signin* endpoint.
 - ii. If the token is not present in the query string, return an "unauthorized access" error using the format specified below
 - iii. If the token is present:
 1. Query the Users table to see if any User records have this AuthToken value. If none are found, return an "unauthorized access" error. Otherwise...
 2. Compare the AuthTokenIssued value from the Users record with the current system date/time. If the AuthTokenIssued indicates it was issued > one hour ago, return an "auth token expired" error.
 3. If the token is valid, proceed with the request as normal.
2. Create a new *questionadd* endpoint that allows new questions to be added to the database.
 - a. If any errors are found with the inputs to the endpoint, return a descriptive error message. Validate question values as:
 - i. Air Date is not empty. (For this assignment you can assume we will always pass valid dates using the same format as the database).
 - ii. Show Number is a valid integer > zero.
 - iii. Dollar Value is a valid integer, an even multiple of 100, and in range 100 through 2000.
 - iv. Question Text is not empty.
 - v. Answer Text is not empty.
 - vi. Category Code is not empty.
 - vii. Category Title is not empty.
 - b. Compare the input CategoryCode and CategoryTitle with what is currently in the Categories table in the

database:

- i. If the input CategoryCode is already in the table, and the input CategoryTitle match the CategoryTitle in the database, do nothing.
 - ii. If the input CategoryCode is already in the table, and the input CategoryTitle does not match the CategoryTitle in the database, return an “Invalid Category Title” error.
 - iii. If the input CategoryCode is not in the table, but the input CategoryTitle is already in the table for a different CategoryCode, return an “Invalid Category Code” error.
 - iv. If the input CategoryCode and input CategoryTitle have no matches in the database, add a new record in the Categories table for the inputs.
- c. If all inputs are valid, add a new record in the Questions table, and to the Categories table if needed.
 - d. Use transaction processing when inserting records into the Questions and Categories table to ensure that either both inserts work, or neither insert works. You can take advantage of the `sqlite3-transactions` package to make this process easier.
3. All endpoints should use JSON for sending and receiving data.
 4. Your web API should expose the endpoints as described below.
 5. Your server should run on the port specified in the PORT environment variable (`process.env.PORT` in node). If the environment variable is not set, default to port 8000.
 6. Continue to implement your Web API using Node.js, SQLite, and the following npm packages: [Express](#), [body-parser](#), [sqlite3](#), and [sqlite3-transactions](#).
 7. For this assignment, *you are allowed* to use 3rd-party JavaScript libraries to help with dates, times, and generating unique identifiers.

Testing

For this assignment, you do not have to build and submit any user interface since it is a back-end only assignment. Of course, you should test your Web API and can do so by creating your own testing UI, by using a web API testing app (e.g. Postman), or even by using a web browser for testing GET requests.

As previously announced, bonus point Assignment 7.5 will allow you to use 3rd-party code to build a UI for this API.

Web API

Endpoint	Get/Post	Description
Token Middleware (Apply to all endpoints except sign in)	--	<p>Token Format</p> <p>Authorization tokens should be passed using the “auth” query string paramater (for example <code>?auth=myTokenHere</code>). If the token is valid, complete the request as normal.</p> <p>If the token is invalid or expired, return a 400 status code with the following format:</p> <pre>{ "message": "unauthorized access", }</pre> <p>If the token is valid but has expired, return a 400 status code with the following format:</p> <pre>{ "message": "auth token expired", }</pre>

		}
/questionadd	POST	<p>Request Format</p> <p>Example JSON for a new entry:</p> <pre>{ "categoryCode": "3LW", "categoryTitle": "3-LETTER WORDS", "airDate": "2004-12-31", "questionText": "'In the title of an Aesop fable, this insect shared billing with a grasshopper'", "dollarValue": 200, "answerText": "the ant", "showNumber": 4680 }</pre> <p>For success, use the following format with a 200 status code:</p> <pre>{ message: "success" }</pre> <p>For unsuccessful execution, use the following format with a 400 status code:</p> <pre>{ message: "descriptive error message" }</pre>

Due Date

Submit your assignment using Vocareum before **Thursday, November 9th, 11:00 pm**. Please make sure to include a valid package.json file, and do not submit the node modules folder or the database file.

Scoring

Max Points	Requirement
5	Correct file submission, coding style and readability, etc.
10	Change signin endpoint to generate new AuthToken and set AuthTokenIssued
15	For all endpoints (excluding the <i>signin</i> endpoint), require a valid AuthToken to be included in the request, and ensure it is < one hour old.
5	questionadd: Validating input values, excluding CategoryCode and CategoryTitle
10	questionadd: Validating CategoryCode and CategoryTitle, and inserting when needed.
5	questionadd: Inserting new record into the Questions table.
5	questionadd: Correct use of transaction begin, commit, and rollback.
55	Total