

Lab 4 Part 2

Providing APIs

Overview

In this lab you will:

- Create a simple API and access the API with curl
- Complete and tidy up your bookshop dynamic web application and make it ready for assessment

You will continue working on the Berties Books application from the last lab.

Note that the extension tasks in this lab are worth a total of 40 marks!

Tasks

Task 1: Add an api route

Let's now add an API to our bookshop so other developers can get a list of our books in machine-readable format.

1. Add the following piece of code to your main.js file:

```
app.get('/api', function (req,res) {  
  
    // Query database to get all the books  
    let sqlquery = "SELECT * FROM books";  
  
    // Execute the sql query  
    db.query(sqlquery, (err, result) => {  
        if (err) {  
            res.redirect('/');  
        }  
        // Return results as a JSON object  
        res.json(result);  
    });  
});
```

2. Test the new added route in your browser. You should see your list of books as a JSON result set.
3. This route is almost identical to the list books route. Compare the two and note the differences.

Task 2: Use your friend's api

This task won't be assessed (in case your friend lets you down!).

4. Find a friend who has created an api for the bookshop. Ask them for their books api. Then create a new menu item "My friend's books" that lists the books on their site.

Task 3: Extension: Add a search term to your API

This is an optional extension task, which will give you additional marks if you attempt it.

5. Add a feature to your API to allow a parameter to add a search term. For example, this URL will search for books that contain the word 'universe':

```
https://www.doc.gold.ac.uk/usr/123/api?keyword=universe
```

Your app should still return a list of *all* books if no keyword is added:

```
https://www.doc.gold.ac.uk/usr/123/api
```

Task 4: Extension: TV shows

This is an optional extension task, which will give you additional marks if you attempt it.

Berties Books is about to get even weirder!

6. Take a look at this website.

<https://www.tvmaze.com/api>

It provides a free API for you to retrieve information about TV shows.

Try this URL in your browser as an example:

<https://api.tvmaze.com/search/shows?q=rick%20morty>

You should see a list of shows. Pick the id from the first one and plug it into this new URL:

<https://api.tvmaze.com/shows/216/episodes>

You should see a list of episodes for the show.

7. Add a new menu item "TV Shows". This should present a form that allows the user to enter some search criteria to search for a TV show (e.g "rick and morty"). Your site should then display a nicely formatted page showing the results. For example, you could list the episodes for a show, or shows matching a search term, or the cast for a show. Explore the API to see what you can find!

- 8.** Make sure you add error handling in case the API fails in some way. If your app crashes we won't be able to complete the marking and you will lose marks.

Task 5: Make your web application ready for submission

- 9.** Push your code to GitLab or GitHub and give Developers permission to me and the lab assistants (see the submission page for details)
- 10.** Make sure your code is nicely formatted and commented.
- 11.** Make sure your web app looks neat. There is no need for fancy CSS styling, but the layout should not be messy, there should not be spelling mistakes, etc.
- 12.** Deploy your application to the live server (virtual server). And test it again!

Task 6: Make your submission

After successful completion of all tasks in lab 4 part 1 and 2, your app is ready to be marked.

Remember to give permission to me and lab assistants to access your GitLab or GitHub repo.

Then run your web application with **'forever'** and submit TWO URLs on the lab 4 assessment page:

1. URL of your git repo,
example: <https://gitlab.doc.gold.ac.uk/username/projectname>
2. URL of your web server: www.doc.gold.ac.uk/usr/ID/

Test your application again, including using erroneous inputs. If your app crashes we won't be able to complete the marking and you will lose marks!

YOU WILL ONLY BE MARKED IF BOTH URLs ARE SUBMITTED and working.

The rubric for this lab submission is as follows:

• Weather page	Lab 4.1. Task 2,3	10 marks
• Weather form	Lab 4.1 Task 4	10 marks
• Other weather info	Lab 4.1 Task 5	10 marks
• Weather API error handling	Lab 4.1 Task 6	10 marks
• Books API	Lab 4.2 Task 1	10 marks
• Extension: Books API search term	Lab 4.2 Task 3	20 marks
• Extension: TV shows	Lab 4.2 Task 4	20 marks
• Nice, tidy code with comments		5 marks
• Clean and tidy user interface		5 marks

END