

ASSESSMENT 6 STUDY EXERCISE

Task: Create an Express server which will be a simple planets API with two endpoints.

Build Specifications:

- In `server.js`, construct a simple Express server that will be hosted on port 3000.
 - `server.js` must require the routes module.
- In `routes.js`, create this array of student objects.

```
let students = [  
  { id:1, firstName: "Obi-Wan", lastName: "Kenobi" },  
  { id:2, firstName: "Boba", lastName: "Fett" },  
  { id:3, firstName: "Luke", lastName: "Skywalker" },  
  { id:4, firstName: "Anakin", lastName: "Skywalker" },  
  { id:5, firstName: "Jango", lastName: "Fett" }  
];
```
- In `routes.js`, construct a Router object named `routes` which has the following endpoints:
 - **GET: /students** - As a response, send back the entire students array as JSON.
 - **GET: /students/:id** - As a response, send back only one of the student objects as JSON.
 - i. It must respond with the student that has the id specified in the URI parameter.
 - ii. If there is no such student in the array, respond with status code 404 (Not Found.).
 - **GET: /students/last-name/:last-name** - As a response, send back a JSON array of students that have that last name.
 - **PUT: /students/:id** - Update the student with the id in the URL with the data from the body of the request.
 - i. If a field is not present or is undefined/null it keeps current value. Id cannot be updated
 - ii. It must respond with the new student data that has the id specified in the URI parameter.
 - iii. If there is no such student in the array, respond with status code 404 (Not Found.).
 - **POST: /students** - Add a student to the students array with the data provided in the body of the request
 - i. It responds with 201 if it succeeds
 - ii. If a field is not present, respond with status code 400 (Bad Request) and what field is missing.
 - iii. If an id is sent with the data and that id exists in the array return with status code 409 (Conflict)



- iv. Bonus: have the id be generated in the route with the next id being kept track of and not generated by just adding 1 the length. Ex if you were to delete id: 5 then insert a new one that would have id: 6
- **DELETE: /students/:id** - Remove the student with the id in the URL
 - i. It responds with 200 and the record that was removed if it succeeds
 - ii. If a field is not present, respond with status code 400 and what field is missing (Bad Request).

