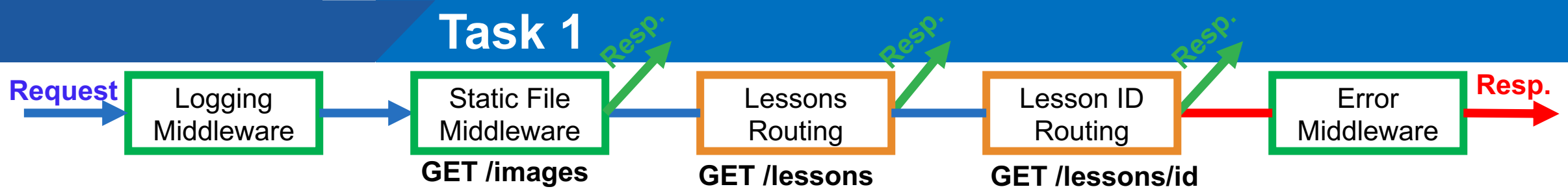


Lab – Routing Functions and Routers



Task 1



Create a **server** using **Express.js**.

The **server** should have the following **Middleware and Routing Functions**.

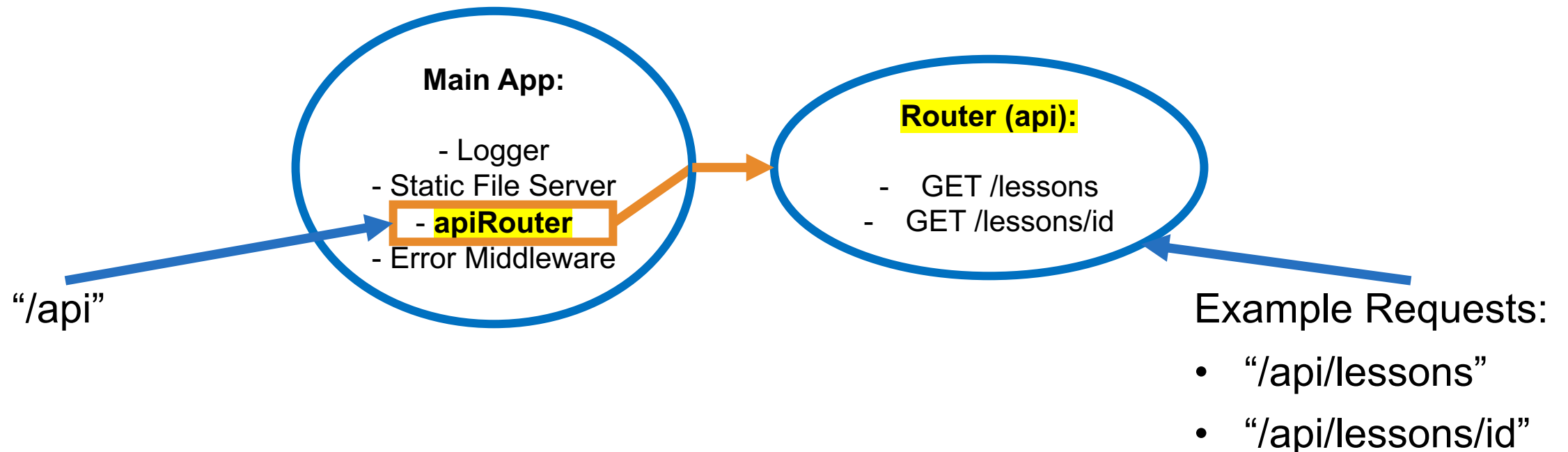
1. **The first middleware** logs all incoming request, and outputs its **url** on the (server) **console**;
2. **The second middleware** responds to **“/images”** by returning the **file (image)** specified by the request url (e.g., **“http://localhost:3000/images/cat.png”**):
 1. it returns the file if it exists;
 2. it goes to the next function if the **file does not exist**, or if there is another error.
3. **The third is a routing function that** responds to **“GET /lessons”** by returning **all the lessons** available in the server as a **JSON**.
4. **The fourth is a routing function that** responds to **“GET /lessons/id”** by returning, as a **JSON**, the lesson with **id** corresponding to the **id parameter** received in the request.
5. **The last middleware** sends back an **error message “Resource not found”** with an **error code '404'**.

Use your **CW1 images** as those to be returned, and such files should be in a **folder called ‘images’** (however, it does need to coincide with the path indicated for the **Static File Routing**, can be also a **different folder name** in the server, such as **“img”**).

Task 2

As the previous task, but this time

- try to organise your project with **SubApplications**,
- where your **routing functions** are within a **SubApplication (Router)**, used by the main application, where
- **routing functions** are exposed as APIs with the path **"/api"**.



Questions?