Lab 6
Node.js and Express

CSCI2720 Building Web Applications

Agenda

- Getting started with Node.js
  - Local installation
  - Online Node.js playgrounds
- Hello World from Express
- Parsing URL parameters
- Obtaining POST parameters

Using Node.js

- You are not able to use Node.js on the lab computers
  - For a web server to be able to serve contents, it has to ***listen*** on a TCP port, e.g. port 8000
  - You won't be able to access that without administrative rights
- Pick one of these ways (or try both if you wish)
  - Install Node.js onto your laptop
  - Use online Node.js playgrounds

Getting Node.js

1. Access https://nodejs.org and follow the link to download the latest ***Current*** version for the OS of your laptop
2. Follow the screen instructions and install
3. Start ***Command Prompt*** (Windows) or ***Terminal*** (Mac/Linux) and issue this command: **node –v** which shows the version number

Local installation ▲

Online playgrounds ▼

1. Pick a flavor (or find any other playgrounds)
   - runkit.com
     - The remaining of this lab will be based on this
   - repl.it
2. You may need to create an account to use the service
3. Don't worry, the free service is good enough for our work

Setting up the first js web app

1. Create a new directory somewhere, e.g. *Desktop/lab6*
2. Navigate to this directory, e.g. `cd Desktop/lab6`
3. Type this command: `npm init`
   - Accept default answers for all questions with **Enter**
4. Install Express

```
npm install express --save
```

Local installation ▲

Online playgrounds ▼

1. … assuming you have already set up an account on runkit.com
2. You may create a "new notebook" if you want to
3. That's it, you don't need to install anything

Hello World from Express

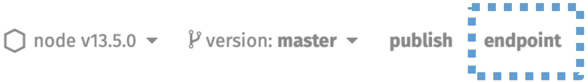
1. Set up a new file in this new directory, e.g. **`server.js`** with the following contents
2. Start the server in the directory by **`node server.js`**
3. Check this out in browser: **`http://localhost:3000`**

```
const express = require('express');
const app = express();

// handle ALL requests
app.all('/*', function (req, res) {
  // send this to client
  res.send("Hello World!");
});

// listen to port 3000
const server = app.listen(3000);
```

Local installation ▲

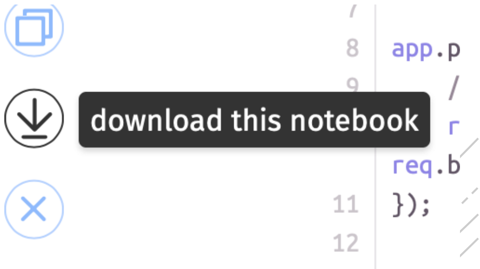
Online playgrounds ▼

1. Put down the above contents into your new notebook
2. You may give this notebook a good name
3. Click ***Run*** (or press ***Shift***+***Enter***)
4. Choose the endpoint link and the page is opened



Which way to use?

- Since we are mainly building very simple web servers at this point, either the local or online Node.js is fine
- The versions v10 to v15 do not matter much too
- From RunKit, if you want you can download the notebook and obtain the js file the ***lib*** directory



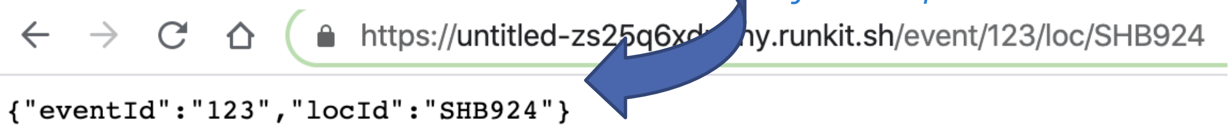
```
app.p
  /
  r
req.b
});
```

Parsing URL parameters

- You can read parameters from URL segments using the **:** operator
- You can try to parse the URL like this by adding it before app.all()

```
app.get('/event/:eventId/loc/:locId', function (req,
res) {
  res.send(req.params);
});
```

Default output



https://untitled-zs25q6xd ny.runkit.sh/event/123/loc/SHB924

```
{"eventId":"123","locId":"SHB924"}
```

- Can you adjust the res.send() contents into this format?

  Event ID: 123
  Loc ID: SHB924

  - You may need a combination of `req.params['eventId']`, text strings, <br>, and so on

Obtaining POST parameters

- GET is usually used for the server to deliver contents
- POST is usually for putting up contents to the server
  - Advantage: contents are put inside the request body
- Using the same URL before with **`eventId`** and **`locId`**, set up a POST rule to accept **`loginId`** from user

Obtaining POST parameters

- You need to set up an extra ***local HTML file*** to make this POST request
  - Inside there should be a form
    - Input box (for `loginId`)
    - Submit button
  - The action should point to the URL on your server
    - *http://localhost:3000/event/…/loc/…*
    - *https://……runkit.sh/event/…/loc/…*
- This should be shown as response

  Event ID: 123
  Loc ID: SHB924
  Login ID: someone

- *Refer to lecture slides on using POST parameters and body parser*

Submission

- No submission is needed for labs
- But what you have done will be useful for your assignment and project
- **Please keep your own code safely**