COMP 431/531: Web Development

Lecture 16: Backend - Express

Mack Joyner (mjoyner@rice.edu)

https://www.clear.rice.edu/comp431

Back End Development

PART II

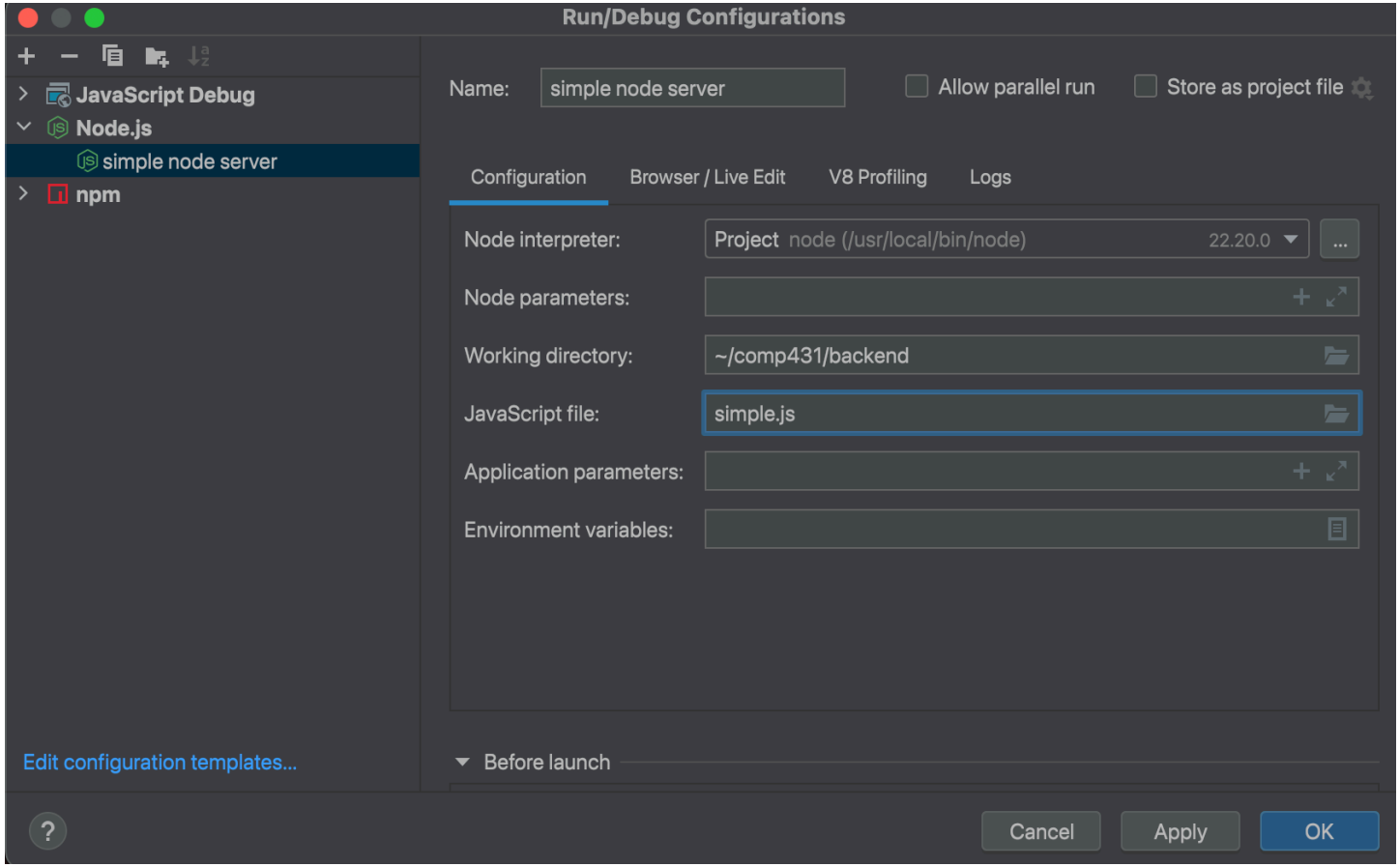
Web Servers
Backend
Architecture
Testing
Web Hosting
Databases

Announcements & Reminders

HW #5 (Front-end) due Thu. Nov. 6th at 11:59pm

classroom hw5 repo: https://classroom.github.com/a/hIFk8cgA

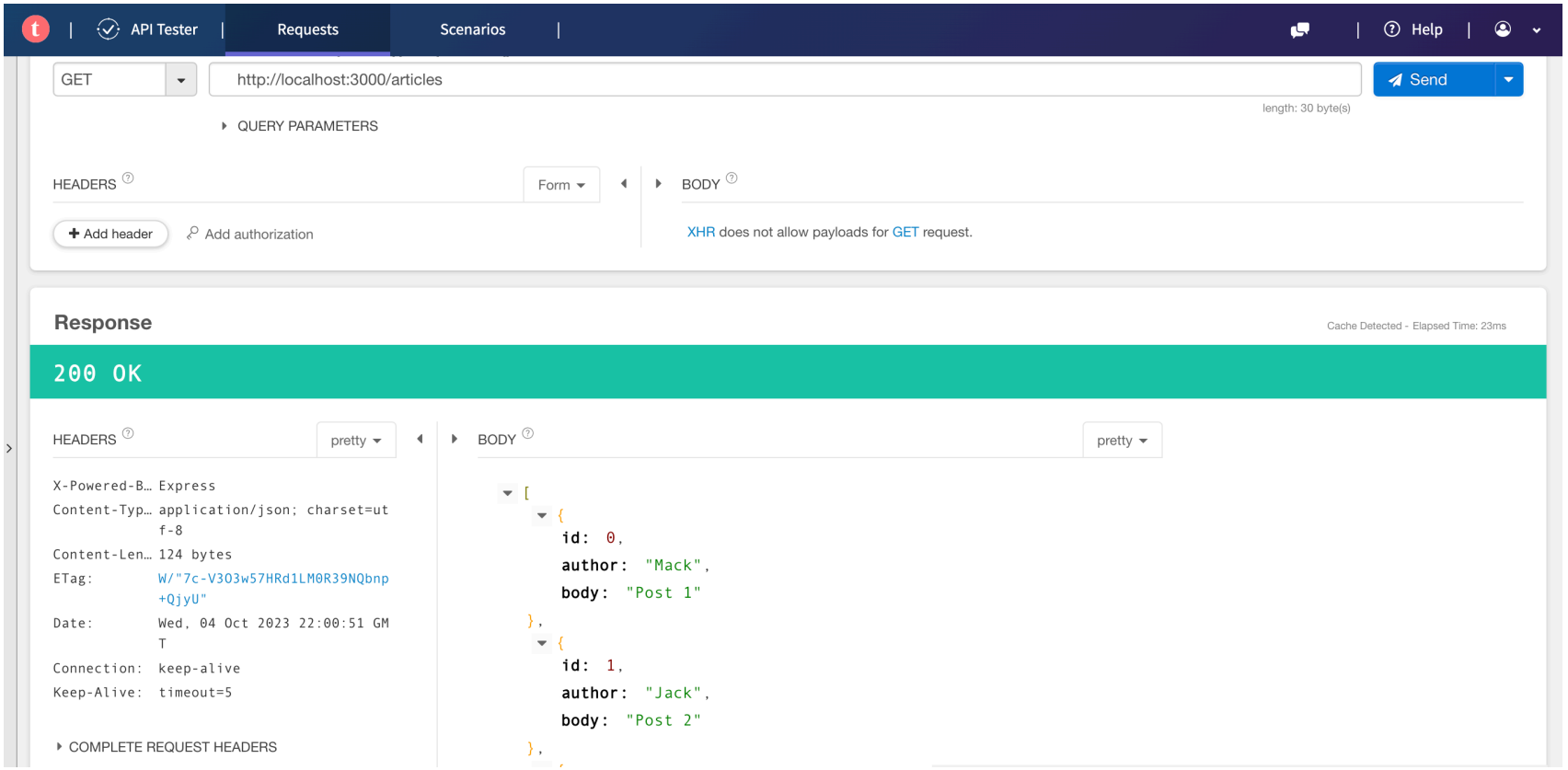
Debugging Node.js



https://www.jetbrains.com/help/idea/running-and-debugging-node-js.html

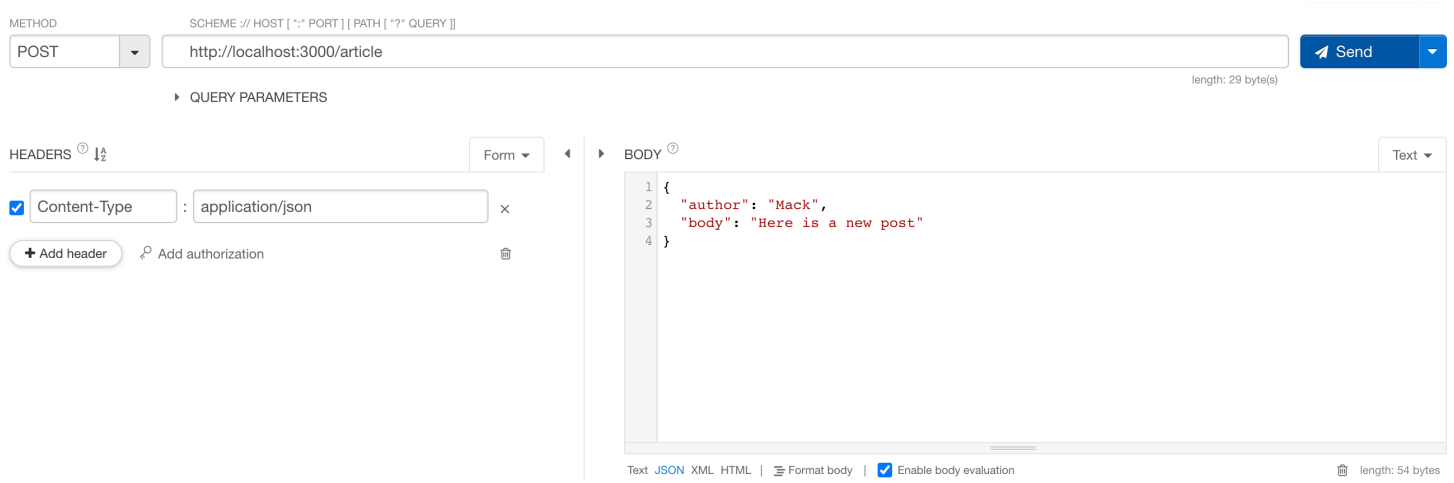
https://plugins.jetbrains.com/plugin/6098-nodejs

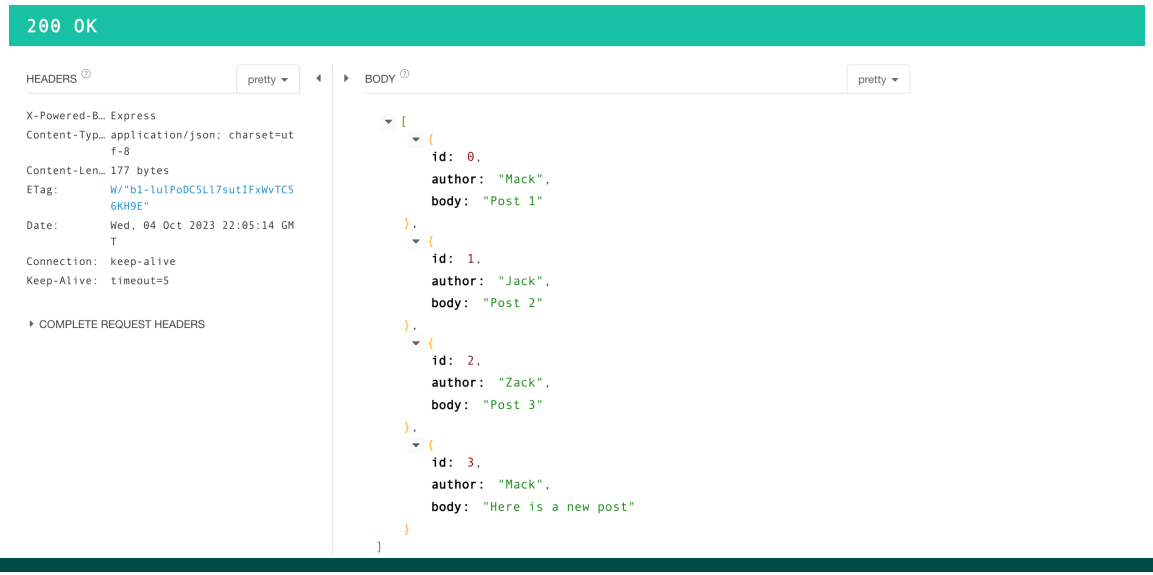
Talend API Tester



Add as a Chrome extension from Chrome web store

Talend API Tester





Talend API Tester Demo

Preprocess Request Body

- Must preprocess req body
  - chunk data
- Would be better to have someone do that for us
  - can use middleware

```
var http = require('http');

var host = '127.0.0.1';
var port = 3333;

http.createServer(preprocess).listen(port, host);
console.log('Server running at http://' + host + ':' + port);

function preprocess(req, res) {
    var body = '';
    req.on('data', function(chunk) {
        body += chunk
    });
    req.on('end', function() {
        req.body = body;
        server(req, res)
    })
}
```


Middleware

- Middleware is anything you put between the server and the final application/framework

- Middleware is compliant
  - accept a request/response and pass it along

- Can modify the request/response
  - check for authentication
  - add/strip headers
  - format or transform content

Express

- Express is part of the MEAN or MERN stack
  - Mongo-Express-Angular-Node
  - Mongo-Express-React-Node

- Express adds to Node a number of helpful libraries
  - minimalist approach
  - middleware

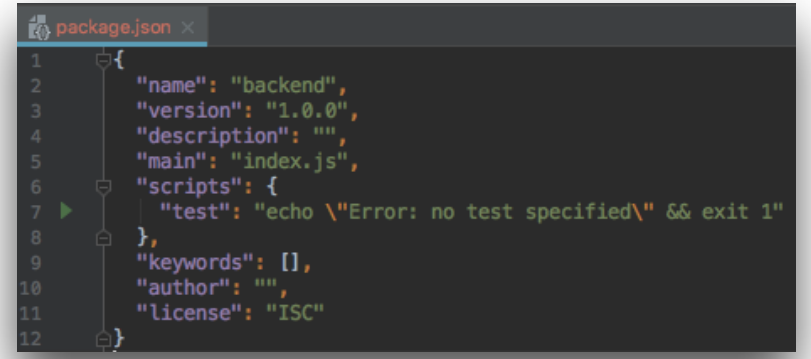
Installing Express

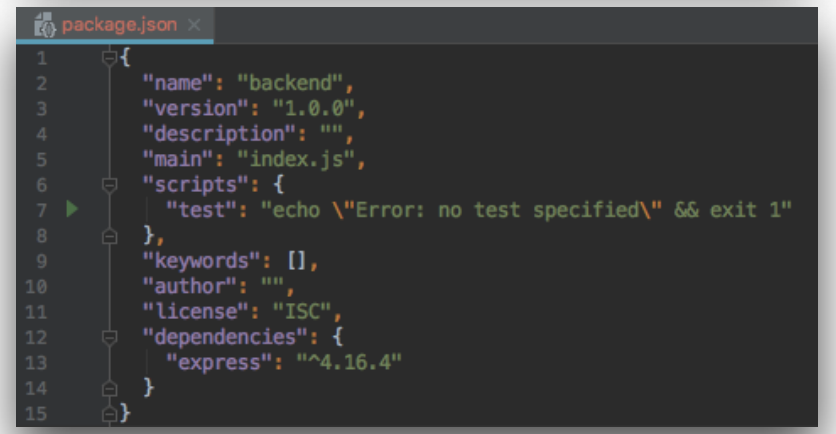
>> mkdir backend

>> cd backend

>> npm init -y (creates package.json)

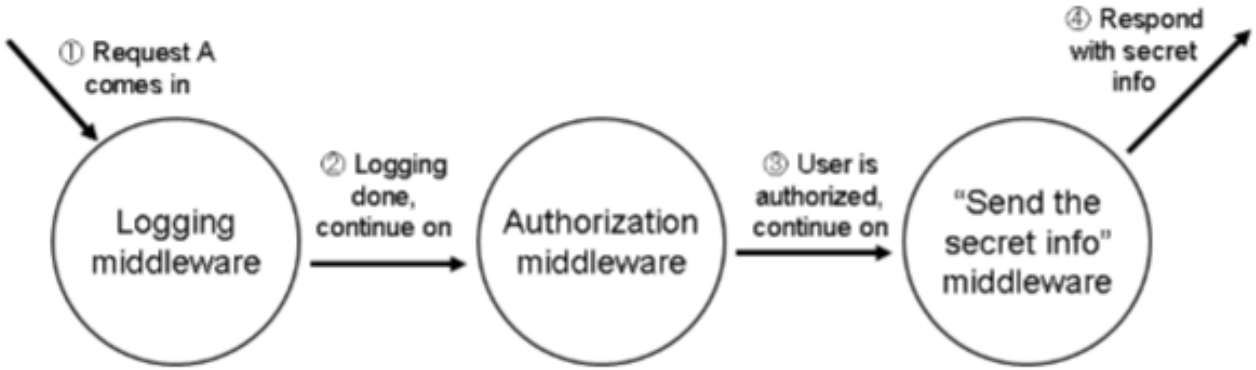
>> npm install express --save

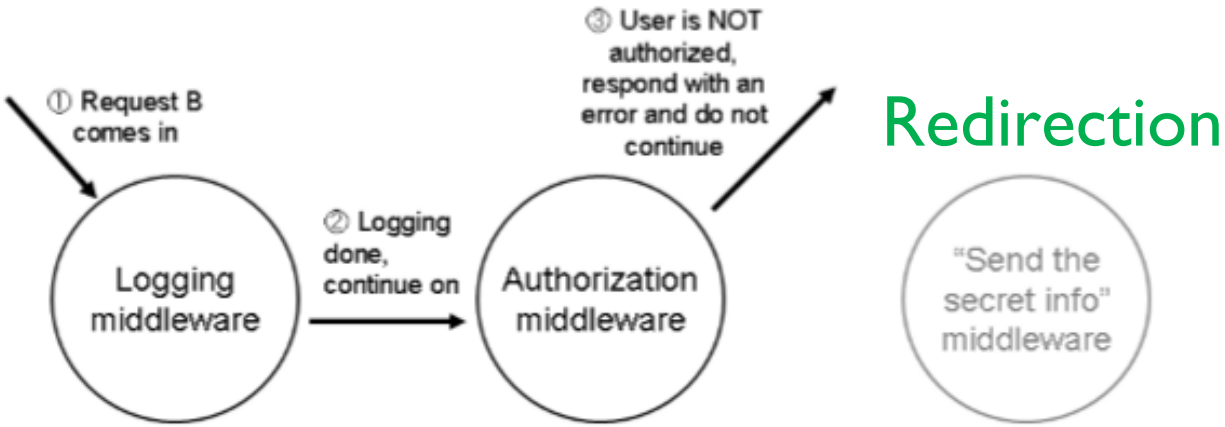




Middleware

```
app.put('/logout', isLoggedIn, logout)
function isLoggedIn(req, res, next)
```





Redirection

Routing with Express

```
var express = require('express')

var app = express()

app.get('/', getIndex)
app.post('/', postIndex)

function getIndex(req, res) {
    res.send('hello world!')
}

function postIndex(req, res) {
    res.send('You POSTed to the homepage')
}
```

```
const port = process.env.PORT || 3000
const server = app.listen(port, () => {
    const addr = server.address()
    console.log(`Server listening at http://${addr.address}:${addr.port}`)
})
```


Preprocess Request Body

- Must preprocess req body
  - chunk data
- Would be better to have someone do that for us
  - can use middleware

```
var http = require('http');

var host = '127.0.0.1';
var port = 3333;

http.createServer(preprocess).listen(port, host);
console.log('Server running at http://' + host + ':' + port);

function preprocess(req, res) {
    var body = '';
    req.on('data', function(chunk) {
        body += chunk
    });
    req.on('end', function() {
        req.body = body;
        server(req, res)
    })
}
```


Install Middleware (body-parser)

>> npm install body-parser --save

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


const addArticle = (req, res) => {
    console.log('Payload received', req.body)
    res.send(req.body)
}

const hello = (req, res) => res.send({ hello: 'world' })

const app = express()

app.post('/article', addArticle)
app.get('/', hello)

const port = process.env.PORT || 3000
const server = app.listen(port, () => {
    const addr = server.address()
    console.log(`Server listening at http://${addr.address}:${addr.port}`)
})
```

```
Server listening at http://:::3000
Payload received undefined
```

```
mjoyner@FVFG21ZAQ6LT ~ % curl -H 'Content-Type: application/json' -d
'{"text":"This is my new article"}' http://127.0.0.1:3000/article
```


Accepting JSON Payloads

>> npm install body-parser --save

```
2  const express = require('express')
3  const bodyParser = require('body-parser')
4
5  const addArticle = (req, res) => {
6      console.log('Payload received', req.body)
7      res.send(req.body)
8  }
9
10 const hello = (req, res) => res.send({ hello: 'world' })
11
12 const app = express()
13 app.use(bodyParser.json())
14 app.post('/article', addArticle)
15 app.get('/', hello)
16
17 const port = process.env.PORT || 3000
18 const server = app.listen(port, () => {
19     const addr = server.address()
20     console.log(`Server listening at http://${addr.address}:${addr.port}`)
21 })
```

```
Server listening at http://:::3000
Payload received { text: 'This is my n
```

```
mjoyner@FVFG21ZAQ6LT ~ % curl -H 'Content-Type: application/json' -d
'{"text":"This is my new article"}' http://127.0.0.1:3000/article
```


Express JSON Payloads

- Node uses require to load modules
- Express use is applied before every endpoint subsequently defined
- No longer need JSON.parse() for req.body
- Passes request, response objects to function

Request Params

- Implement GET /articles/:id, not
  - /articles/1
  - /articles/2
  - /articles/3

```
app.get('/article/:id', getArticle);
```

- The :id is a parameter whose value is determined at runtime

- Get the value from request (e.g. req.params.id)

Create a New Post

Attempt to create a post with a unique id:

```
let post = { posts.length: req.body };
```

Error: Can't perform field access for object key

```
let post = {}; post[posts.length] = req.body;
```

Use dictionary approach to access post by id using length