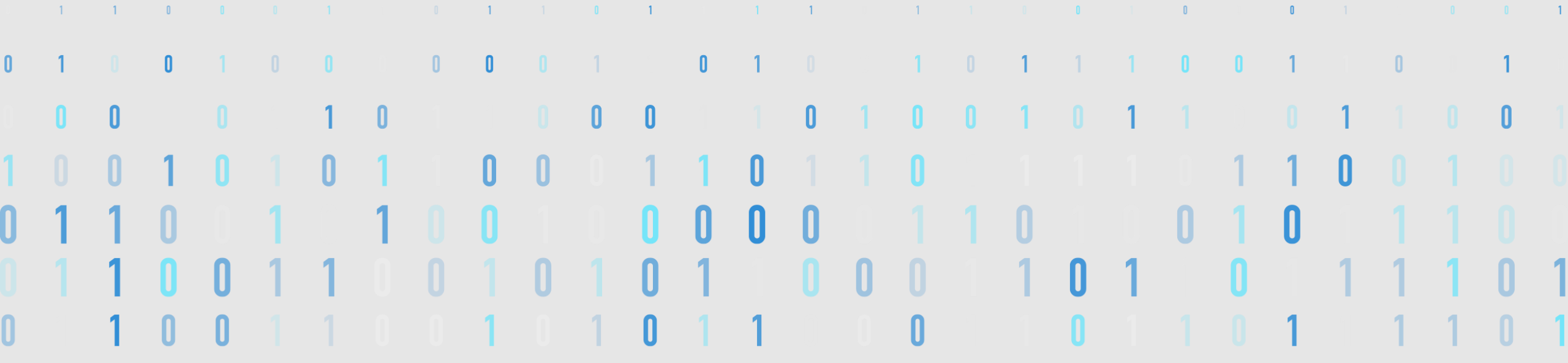
Introduction to Backend Development and Deployment

Via NodeJS and Azure

Session 2 on 11/04/20

Microsoft Student Partners

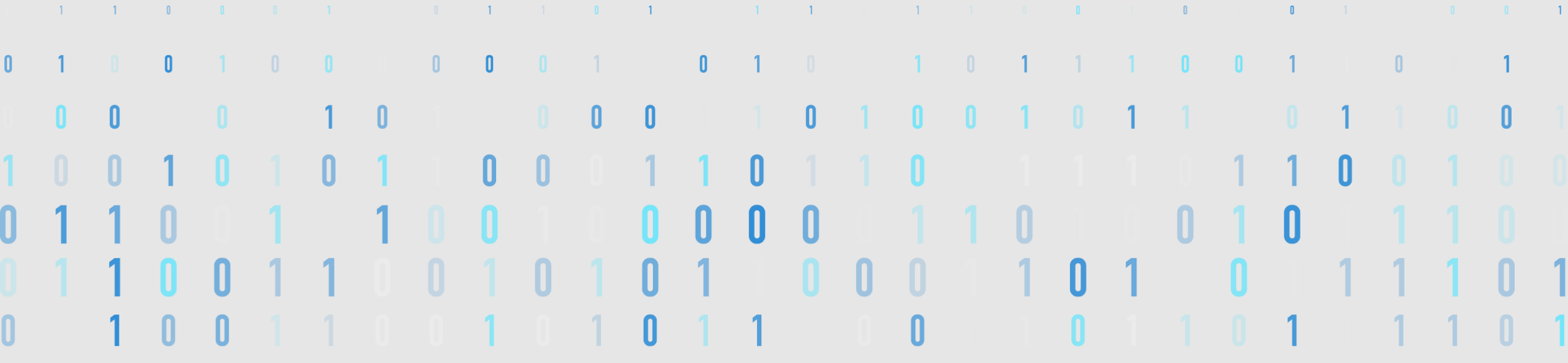
About this Session



Topic to be covered

- Routes in ExpressJS
  - Routings
  - Route Path
  - Route Methods
- Middleware in ExpressJS
- Routers in ExpressJS
- Returning HTML as Response in ExpressJS

Route



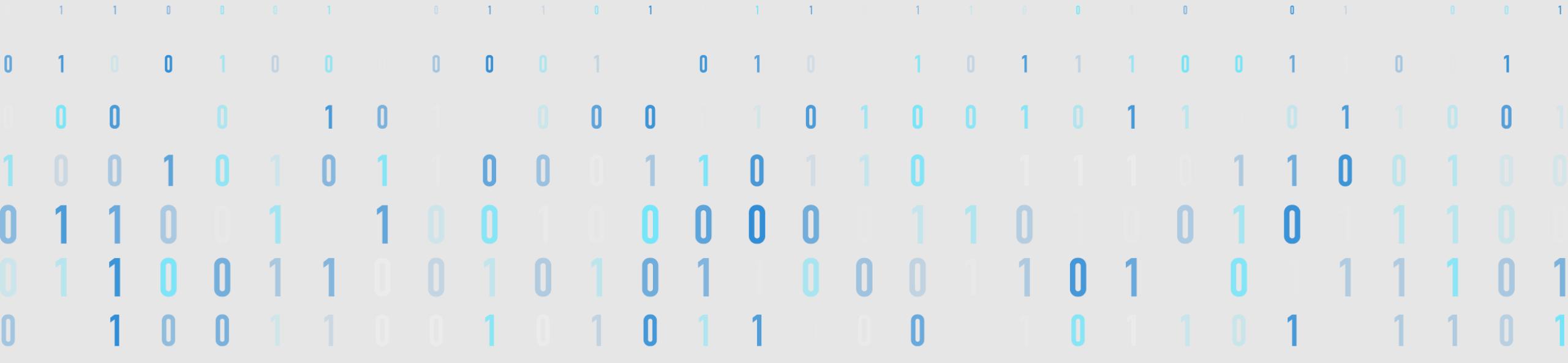
Routing

- An application's endpoints (URIs) respond to client requests.
- Define routing using methods of the Express app object that correspond to HTTP methods;
- for example:
  - app.get() to handle GET requests
  - app.post to handle POST requests.
  - use app.all() to handle all HTTP methods
  - app.use() to specify middleware as the callback function

Route Path

- Route paths can be strings, string patterns, or regular expressions.
- The characters ?, +, *, and () are subsets of their regular expression counterparts.
- The hyphen (-) and the dot (.) are interpreted literally by string-based paths.
- If you need to use the dollar character ($) in a path string, enclose it escaped within ([and]).
- For example, the path string for requests at “/data/$book”, would be “/data/([\$])book”.
- See the path-to-regexp documentation for all the possibilities in defining route paths

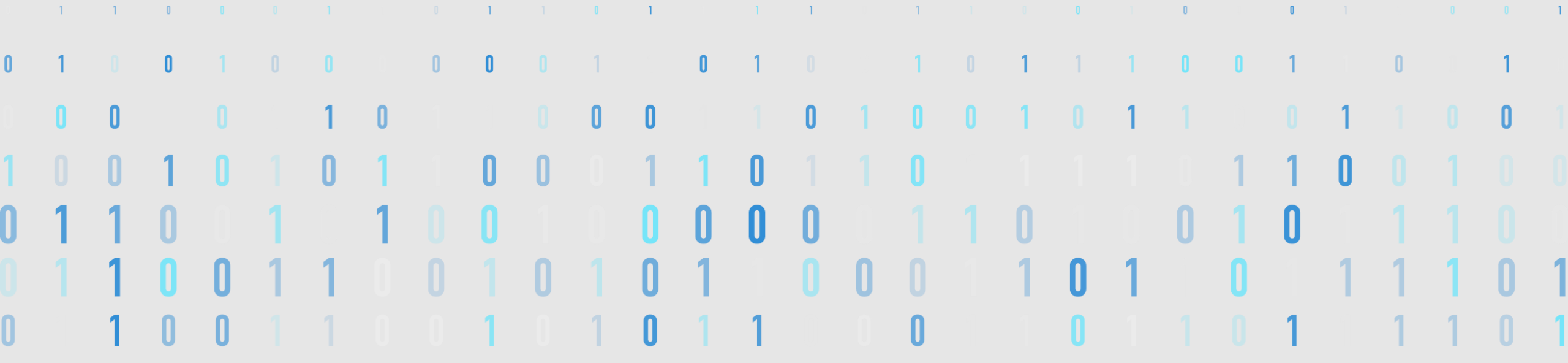
Demo



Routes Basic

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


app.route('/msg/hello')
.get((req, res) => {
  res.send('Hello World! Get');
})
.post((req, res) => {
  res.send('Hello World! Post');
});


app.listen(8080)
```

Middlewares

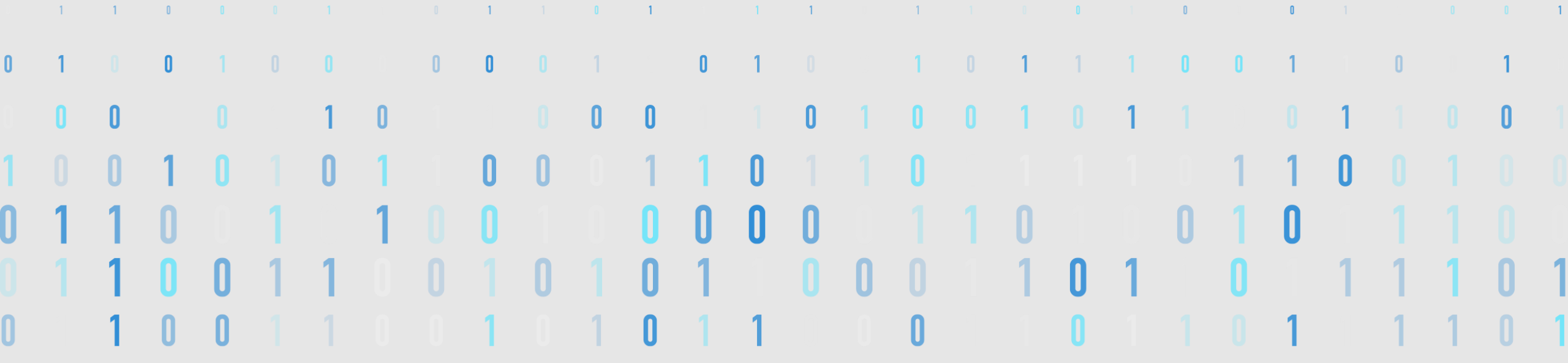


What is a Middleware?

- *Middleware* functions are functions that have access to following object in the application's request-response cycle:
  - the request object (req),
  - the response object (res)
  - the next function .

- The next function is a function in the Express router which, when invoked, executes the middleware succeeding the current middleware.
- If the current middleware function does not end the request-response cycle, it must call next() to pass control to the next middleware function. Otherwise, the request will be left hanging.

Demo



Middleware Basic

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


app.use((req, res, next) => {
  console.log('LOGGED');
  next();
});

app.get('/', function (req, res) {
  res.send('Hello World!');
});

app.listen(8080)
```

Middleware Path Specific

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


app.use('/msg',(req, res, next) => {
  console.log('LOGGED');
  next();
});

app.get('/msg/hello', function (req, res) {
  res.send('Hello World!');
});

app.get('/msg/bye', function (req, res) {
  res.send('Bye World!');
});

app.listen(8080)
```

Middleware Module Import

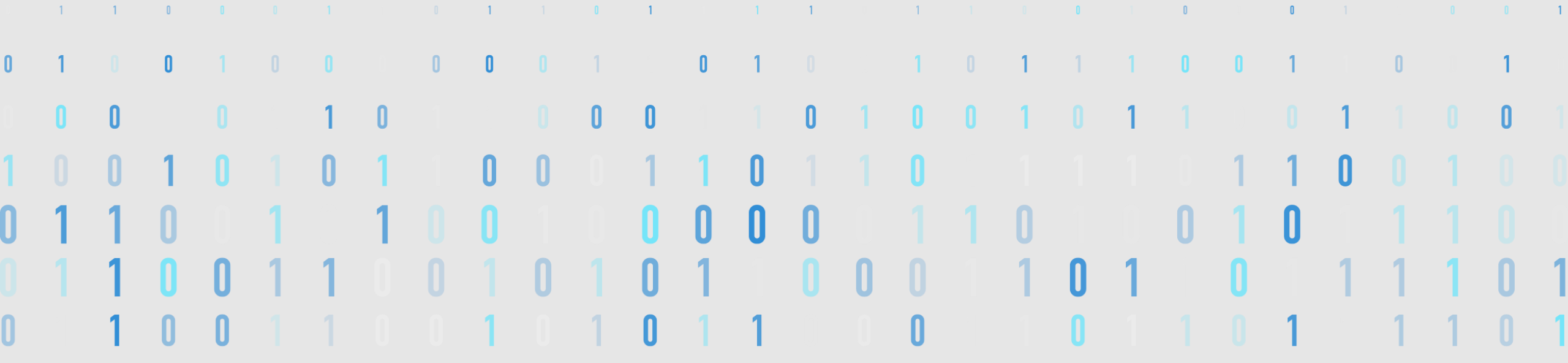
File One

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

const apiCheck = require('./utils/apicheck');

app.use(apiCheck);

app.get('/msg/hello', function (req, res) {
  res.send('Hello World!');
});

app.get('/msg/bye', function (req, res) {
  res.send('Bye World!');
});

app.listen(8080)
```

File Two

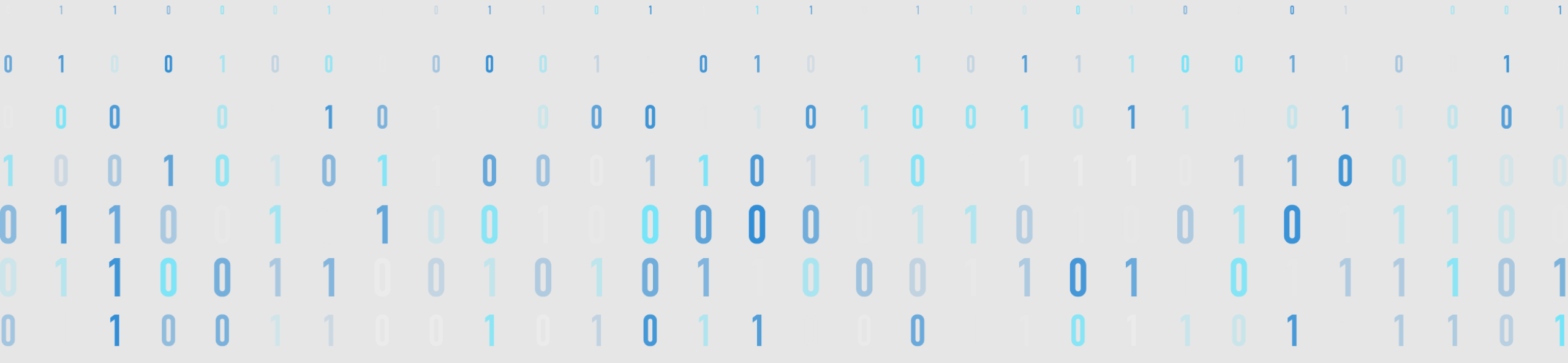
```
const apiKey = 'abcd1234';


function apiCheck(req,res,next) {
    if (req.headers['x-api-key'] !== apiKey) {
        return res.send('Invalid APIKey');
    }
    next();
}


module.exports = apiCheck;
```

Routers



- Router class to create modular, mountable route handlers.
- A Router instance is a complete middleware and routing system
- It is often referred to as a “mini-app”.

Demo



Router Basic

File One

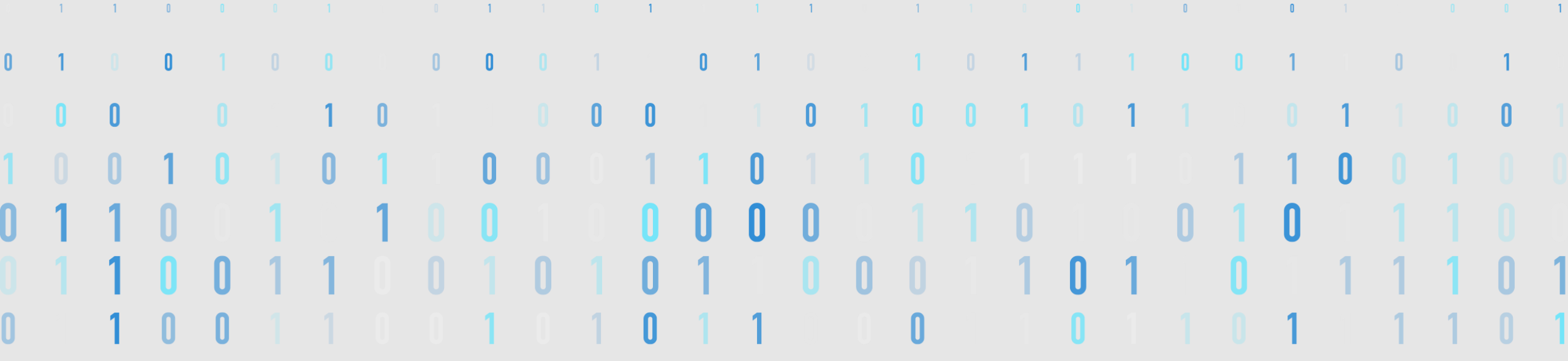
```
const express = require('express');
const app = express();


const msg = require('./controller/message');


app.use('/msg', msg);


app.get('', (req, res) => {
  res.send('Let\'s Go');
});
```

File Two

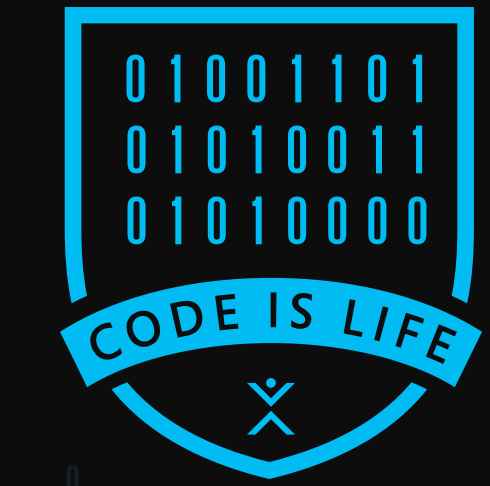
```
const express = require('express');
const router = express.Router()


router.get('', (req, res) => {
  res.send('Hey it\'s message route');
});


router.get('/hello', (req, res) => {
  res.send('Hello World');
});


module.exports = router;
```

Demo: HTML File as Response

Check the repo for the code: https://github.com/ieeeditu/Backend-Dev-and-Deploy/tree/master/Session-2/HTML_Example .



Thank You



About The Speaker

Himanshu Kotnala
Email: kotnala.himanshu@gmail.com
MSP Collaboration: himanshukotnala@studentpartner.com
LinkedIn: https://www.linkedin.com/in/aker99/
Github: https://github.com/aker99/

Microsoft Student Partners