

FS

- `require`

```
var http = require('http');
var fs = require('fs');
var url = require('url');
var qs = require('querystring');
var template = require('./lib/template.js');
var path = require('path');
var sanitizeHtml = require('sanitize-html');
```

- `http.createServer`

```
var app = http.createServer(function(request, response){
  var _url = request.url;
  var queryData = url.parse(_url, true).query;
  var pathname = url.parse(_url, true).pathname;
  if(pathname === '/'){
    if(queryData.id === undefined){
      fs.readdir('./data', function(error, filelist){
        var title = 'Welcome';
        var description = 'Hello, Node.js';
        var list = template.list(filelist);
        var html = template.HTML(title, list,
          `

## ${title}</h2>${description}`, `\${sanitizedTitle}</h2>\${sanitizedDescription}`, `


```

```

    });
  });
}
}

```

- `create` + `create_process`

```

else if(pathname === '/create'){
  fs.readdir('./data', function(error, filelist){
    var title = 'WEB - create';
    var list = template.list(filelist);
    var html = template.HTML(title, list, `
      <form action="/create_process" method="post">
        <p><input type="text" name="title" placeholder="title">
</p>
        <p>
          <textarea name="description"
placeholder="description"></textarea>
        </p>
        <p>
          <input type="submit">
        </p>
      </form>
    `, '');
    response.writeHead(200);
    response.end(html);
  });
} else if(pathname === '/create_process'){
  var body = '';
  request.on('data', function(data){
    body = body + data;
  });
  request.on('end', function(){
    var post = qs.parse(body);
    var title = post.title;
    var description = post.description;
    fs.writeFile(`data/${title}`, description, 'utf8',
function(err){
    response.writeHead(302, {Location: `/?id=${title}`});
    response.end();
  })
  });
}

```

- `update` + `update_process`

```

else if(pathname === '/update'){
  fs.readdir('./data', function(error, filelist){
    var filteredId = path.parse(queryData.id).base;
    fs.readFile(`data/${filteredId}`, 'utf8', function(err, description)
{
    var title = queryData.id;
    var list = template.list(filelist);
    var html = template.HTML(title, list, `
      <form action="/update_process" method="post">

```

```

        <input type="hidden" name="id" value="${title}">
        <p><input type="text" name="title" placeholder="title"
            value="${title}"></p>

        <p>
            <textarea name="description"
placeholder="description">${description}</textarea>
        </p>
        <p>
            <input type="submit">
        </p>
    </form>
    `
    ,
    `<a href="/create">create</a> <a href="/update?
        id=${title}">update</a>`

    );
    response.writeHead(200);
    response.end(html);
    });
    });
} else if(pathname === '/update_process'){
    var body = '';
    request.on('data', function(data){
        body = body + data;
    });
    request.on('end', function(){
        var post = qs.parse(body);
        var id = post.id;
        var title = post.title;
        var description = post.description;
        fs.rename(`data/${id}`, `data/${title}`, function(error){
            fs.writeFile(`data/${title}`, description, 'utf8',
                function(err){
                    response.writeHead(302, {Location: `/?id=${title}`});
                    response.end();
                })
        });
    });
}
}

```

- delete_process

```

else if(pathname === '/delete_process'){
    var body = '';
    request.on('data', function(data){
        body = body + data;
    });
    request.on('end', function(){
        var post = qs.parse(body);
        var id = post.id;
        var filteredId = path.parse(id).base;
        fs.unlink(`data/${filteredId}`, function(error){
            response.writeHead(302, {Location: `/${}`});
            response.end();
        })
    });
} else {
    response.writeHead(404);
}

```

```
response.end('Not found');
}
```

- `template.js`

```
module.exports = {
  HTML:function(title, list, body, control){
    return `
    <!doctype html>
    <html>
    <head>
      <title>WEB1 - ${title}</title>
      <meta charset="utf-8">
    </head>
    <body>
      <h1><a href="/">WEB</a></h1>
      ${list}
      ${control}
      ${body}
    </body>
    </html>
    `;
  },list:function(filelist){
    var list = '<ul>';
    var i = 0;
    while(i < filelist.length){
      list = list + `<li><a href="/?id=${filelist[i]}">${filelist[i]}</a>
</li>`;
      i = i + 1;
    }
    list = list+'</ul>';
    return list;
  }
}
```

MySQL

- `require` + `createConnection`

```
var http = require('http');
var fs = require('fs');
var url = require('url');
var qs = require('querystring');
var template = require('./lib/template.js');
var path = require('path');
var sanitizeHtml = require('sanitize-html');
var mysql = require('mysql');
var pw = require('./pw.js');

var db = mysql.createConnection({
  host : 'localhost',
  user : 'root',
  password : pw.pw,
  database : 'opentutorials'
});
db.connect();
```

- `http.createServer`

```
var app = http.createServer(function(request,response){
  var _url = request.url;
  var queryData = url.parse(_url, true).query;
  var pathname = url.parse(_url, true).pathname;
  if(pathname === '/'){
    if(queryData.id === undefined){
      var title = 'welcome';
      var description = 'Hello, Node.js';
      db.query('SELECT * FROM topic', function(error, topics) {
        var list = template.list(topics);
        var html = template.HTML(title, list,
          `

## ${title}</h2>${description}`, `\${title}</h2>\${description} <br><br>by \${topic\[0\].name}`, `


```

- `create` + `create_process`

```
else if(pathname === '/create'){
  db.query(`SELECT * FROM topic`, function(error, topics) {
    db.query(`SELECT * FROM author`, function(error2, authors) {
      if (error) {
```

```

        throw error;
    }
    var title = 'Create';
    var list = template.list(topics);
    var html = template.HTML(title, list, `
        <form action="/create_process" method="post">
            <p><input type="text" name="title" placeholder="title"></p>
            <p>
                <textarea name="description" placeholder="description">
</textarea>
                </p>
                <p>
                    ${template.authorSelect(authors)}
                </p>
                <p>
                    <input type="submit">
                </p>
            </form>
        `, '');
    response.writeHead(200);
    response.end(html);
    })
    })
} else if(pathname === '/create_process'){
    var body = '';
    request.on('data', function(data){
        body = body + data;
    });
    request.on('end', function(){
        var post = qs.parse(body);
        db.query(`INSERT INTO topic (title, description, created,
author_id)
                VALUES(?,?, NOW(),?)`, [post.title, post.description,
post.author],
                function(error2, result) {
                    if (error2) {
                        throw error2;
                    }
                    response.writeHead(302, {Location: `/?id=${result.insertId}`});
                    response.end()
                })
    });
}
}

```

- update + update_process

```

else if(pathname === '/update'){
    db.query(`SELECT * FROM topic`, function(error, topics) {
        db.query(`SELECT * FROM topic WHERE id = ${queryData.id}`,
function (error2, topic) {
            var list = template.list(topics);
            var html = template.HTML(topic[0].title, list,
            `
                <form action="/update_process" method="post">
                    <input type="hidden" name="id" value="${topic[0].id}">
                    <p><input type="text" name="title" placeholder="title"
value="${topic[0].title}"></p>

```

```

        <p>
          <textarea name="description"
placeholder="description">${topic[0].description}</textarea>
        </p>
        <p>
          <input type="submit">
        </p>
      </form>
    `
    `<a href="/create">create</a> <a href="/update?
id=${topic[0].id}>update</a>`
    );
    response.writeHead(200);
    response.end(html);
  })

  })
} else if(pathname === '/update_process'){
  var body = '';
  request.on('data', function(data){
    body = body + data;
  });
  request.on('end', function(){
    var post = qs.parse(body);
    db.query(`UPDATE topic SET title = ?, description = ? WHERE id = ?
    `,
      [post.title, post.description, post.id], function(error, result)
    {
      if (error) {
        throw error;
      }
      response.writeHead(302, {Location: `/?id=${post.id}`});
      response.end();
    })
  });
}
}

```

- delete_process

```

else if(pathname === '/delete_process'){
  var body = '';
  request.on('data', function(data){
    body = body + data;
  });
  request.on('end', function(){
    var post = qs.parse(body);
    db.query(`DELETE FROM topic WHERE id = ${post.id}`,
      function(error, result) {
        if (error) {
          throw error;
        }
        response.writeHead(302, {Location: `/\`});
        response.end();
      })
  });
} else {
  response.writeHead(404);
}

```

```

        response.end('Not found');
    }
});

```

- `template.js`

```

module.exports = {
  HTML:function(title, list, body, control){
    return `
    <!doctype html>
    <html>
    <head>
      <title>WEB1 - ${title}</title>
      <meta charset="utf-8">
    </head>
    <body>
      <h1><a href="/">WEB</a></h1>
      ${list}
      ${control}
      ${body}
    </body>
    </html>
    `;
  },
  list:function(filelist){
    var list = '<ul>';
    var i = 0;
    while(i < filelist.length){
      list = list + `<li><a href="/?
id=${filelist[i].id}>${filelist[i].title}</a></li>`;
      i = i + 1;
    }
    list = list+'</ul>';
    return list;
  },
  authorSelect:function(authors) {
    var tag='';
    tag += '<select name="author">'
    for (var i=0;i<authors.length;i++) {
      tag += `<option value="${authors[i].id}">${authors[i].name}</option>`
    }
    tag += '</select>'

    return tag
  }
}

```

Express

- `main.js`

```

var fs = require('fs');
var template = require('./lib/template.js');
var express = require('express');
var bodyParser = require('body-parser');
var compression = require('compression');

```



```

var topicRouter = require('./routes/topic.js');
var indexRouter = require('./routes/index.js');

const app = express()

app.use(express.static('public'));
app.use(bodyParser.urlencoded({extended: false}))
app.use(compression());

app.get('*', function(req, res, next) {
  fs.readdir('./data', function(error, filelist) {
    req.list = filelist;
    next();
  })
})

app.use('/', indexRouter)
app.use('/topic', topicRouter)

app.use(function(req, res, next) {
  res.status(404).send('Sorry cant find that!');
});

app.use(function(err, req, res, next) {
  console.error(err.stack)
  res.status(500).send('Something broke!')
})

app.listen(3000, () => console.log('Example app listening on port 3000!'))

```

- index.js

```

var express = require('express');
var template = require('../lib/template.js');
var fs = require('fs');

var router = express.Router();

router.get('/', function(req, res) {
  var title = 'welcome';
  var description = 'Hello, Node.js';
  var list = template.list(req.list);
  var html = template.HTML(title, list,
    `<h2>${title}</h2>${description}`,
    `<a href="/topic/create">create</a>`
    ``
  );
  res.send(html);
});

module.exports = router

```

- topic.js

```

var express = require('express');

```

```

var fs = require('fs');
var template = require('../lib/template.js');
var sanitizeHtml = require('sanitize-html');
var path = require('path');

var router = express.Router();

router.get('/create', function(req, res) {
  var title = 'WEB - create';
  var list = template.list(req.list);
  var html = template.HTML(title, list, `
    <form action="/topic/create_process" method="post">
      <p><input type="text" name="title" placeholder="title"></p>
      <p>
        <textarea name="description" placeholder="description"></textarea>
      </p>
      <p>
        <input type="submit">
      </p>
    </form>
  `, '');
  res.send(html);
})

router.post('/create_process', function(req, res) {
  var post = req.body;
  var title = post.title;
  var description = post.description;
  fs.writeFile(`./data/${title}`, description, 'utf8', function(err){
    res.redirect(`/topic/${title}`)
  })
})

router.get('/update/:pageId', function(req, res) {
  var filteredId = path.parse(req.params.pageId).base;
  fs.readFile(`./data/${filteredId}`, 'utf8', function(err, description){
    var title = req.params.pageId;
    var list = template.list(req.list);
    var html = template.HTML(title, list, `
      <form action="/topic/update_process" method="post">
        <input type="hidden" name="id" value="${title}">
        <p><input type="text" name="title" placeholder="title"
value="${title}"></p>
        <p>
          <textarea name="description"
placeholder="description">${description}</textarea>
        </p>
        <p>
          <input type="submit">
        </p>
      </form>
    `, `
    <a href="/topic/create">create</a> <a
href="/topic/update/${title}">update</a>`
  );
  res.send(html);
});

```

```

})

router.post('/update_process', function(req, res) {
  var post = req.body;
  var id = post.id;
  var title = post.title;
  var description = post.description;
  fs.rename(`./data/${id}`, `./data/${title}`, function(error){
    fs.writeFile(`./data/${title}`, description, 'utf8', function(err){
      res.redirect(`/topic/${title}`);
    })
  })
});

router.post('/delete_process', function(req, res) {
  var post = req.body;
  var id = post.id;
  var filteredId = path.parse(id).base;
  fs.unlink(`./data/${filteredId}`, function(error){
    res.redirect('/')
  })
});

router.get('/:pageId', function(req, res, next) {
  var filteredId = path.parse(req.params.pageId).base;
  fs.readFile(`./data/${filteredId}`, 'utf8', function(err, description){
    if (err) {
      next(err);
    } else {
      var title = req.params.pageId;
      var sanitizedTitle = sanitizeHtml(title);
      var sanitizedDescription = sanitizeHtml(description, {
        allowedTags: ['h1']
      });
      var list = template.list(req.list);
      var html = template.HTML(sanitizedTitle, list,
        `

## ${sanitizedTitle}</h2>${sanitizedDescription}`, `


```

- `templete.js`

```

module.exports = {
  HTML: function(title, list, body, control){

```

```

return `
<!doctype html>
<html>
<head>
  <title>WEB1 - ${title}</title>
  <meta charset="utf-8">
</head>
<body>
  <h1><a href="/">WEB</a></h1>
  ${list}
  ${control}
  ${body}
</body>
</html>
`;
},list:function(filelist){
  var list = '<ul>';
  var i = 0;
  while(i < filelist.length){
    list = list + `<li><a href="/topic/${filelist[i]}">${filelist[i]}</a>
</li>`;
    i = i + 1;
  }
  list = list+'</ul>';
  return list;
}
}

```

학습평가

```

var mysql = require('mysql')
var http = require('http')
var pw = require('./pw.js')
var url = require('url');
var path = require('path');
var qs = require('querystring');
var express = require('express');
var bodyParser = require('body-parser')

var db = mysql.createConnection({
  host : 'localhost',
  user : 'root',
  password : pw.pw,
  database : 'games'
});
db.connect();

var app = express();

app.use(bodyParser.urlencoded({ extended: false }));

app.get('/', function(req, res) {
  db.query('SELECT * from card2',function (err, cards) {
    var body = `
<!doctype html>
<html>
<head>

```

```

<title>CARD</title>
</head>
<body>
<table border = "1">
  <tr>
    <th>ID</th>
    <th>SUIT</th>
    <th>RANK</th>
  </tr>
  `
for (var i=0;i<cards.length;i++) {
  body += `<tr>
    <td>${cards[i].id}</td>
    <td>${cards[i].suit}</td>
    <td>${cards[i].rank}</td>
  </tr>`
}

  body += '</table> <a href="/create">create</a> <a href ="/delete">delete</a>
</body> </html>'

  res.send(body);
})
})

app.get('/create', function(req, res) {
  var body = `
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>CARD</title>
</head>
<body>
  <form action="create_process" method="post">
    <input type="number" name="id">
    <br>
    <input type="text" name="suit">
    <br>
    <input type="text" name="rank">
    <br>
    INSERT <input type="submit">
  </form>
</body>
</html>
  `;

  res.send(body);
});

app.post('/create_process', function(req, res) {
  var post = req.body;
  db.query(`INSERT INTO card2 VALUES(${post.id},?,?)`, [post.suit, post.rank],
function(err, result) {
  if (err) {
    throw err;
  }
}

```

```

    res.redirect('/');
  })
})

app.get('/delete', function(req, res) {
  var body = `
    <!doctype html>
    <html>
    <head>
    <meta charset="utf-8">
    <title>CARD</title>
    </head>
    <body>
      <form action="delete_process" method="post">
        <input type="number" name="id">
        <br>
        DELETE <input type="submit">
      </form>
    </body>
    </html>
  `;

  res.send(body);
})

app.post('/delete_process', function(req, res) {
  var post = req.body;
  db.query(`DELETE FROM card2 WHERE id = ?`, [post.id], function(err, result) {
    if (err) {
      throw err;
    }
    res.redirect('/');
  })
})

app.listen(3000, () => console.log('Server running at port 3000'));

```