

# 2nd Project Story Board

## 기본 설치

- `mkdir 2ndproject`
- `cd 2ndproject`
- `express --view=ejs`
- `npm i`
- `npm i --save-dev nodemon`
- `package.json`

```
{
  "name": "1stproject-test",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "nodemon ./bin/www" //이부분 nodemon으로 수정 },
    "dependencies": {
      "cookie-parser": "~1.4.4",
      "debug": "~2.6.9",
      "ejs": "~2.6.1",
      "express": "~4.16.1",
      "http-errors": "~1.6.3",
      "morgan": "~1.9.1"
    },
    "devDependencies": {
      "nodemon": "^1.19.1"
    }
  }
}
```

- `npm start` -> <http://localhost:3000> URL로 Browser에서 확인

## Template Download & 적용

- 기존에 있던 `index.ejs`를 지우고 템플릿에 있던 `index.html`을 `index.ejs`로 파일명만 바꿔서 넣기.
- `index.ejs`

```
...
<head>
```

```

<title>Dimension by HTML5 UP</title>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1,
user-scalable=no" />
<link rel="stylesheet" href="stylesheets/main.css" />
<noscript><link rel="stylesheet" href="stylesheets/noscript.css" />
</noscript>
</head>
...
<!-- Scripts -->
<script src="javascripts/jquery.min.js"></script>
<script src="javascripts/browser.min.js"></script>
<script src="javascripts/breakpoints.min.js"></script>
<script src="javascripts/util.js"></script>
<script src="javascripts/main.js"></script>

</body>
</html>

```

- css파일이 들어있는 폴더경로를 stylesheets/— .css 로 바꿔주고 js파일이 들어있는 폴더경로를 javascripts 로 바꿔준다.
- 100번째 라인 부터(id=elements) 340번째 라인까지는 지워준다. ( 템플릿에 대한 설명임 )

## 회원가입

- index.ejs파일 수정
- id=about

```

...
<nav> <!-- nav에서 세번째부분 수정 -->
<ul>
<li><a href="#intro">Intro</a></li>
<li><a href="#work">Work</a></li>
<li><a href="#signup">Signup</a></li>
<li><a href="#contact">Contact</a></li>
</ul>
</nav>
...
<article id="signup"> <!--article의 id값을 signup으로 수정-->
<h2 class="major">Sign up</h2>
<div id="sign_up_div">
<div class="form">
<div class="fields">
<div class="field">

```

```

        <label for="name">ID</label>
        <input type="text" name="id" id="id" />
    </div>
    <div class="field">
        <label for="password">Password</label>
        <input type="password" name="password" id="password" />
    </div>
    <div class="field half">
        <label for="name">Name</label>
        <input type="text" name="name" id="name" />
    </div>
    <div class="field half">
        <label for="phone">Phone</label>
        <input type="tel" name="phone" id="phone"/>
    </div>
    <div class="field half">
        <label for="birth">Identity num</label>
        <input type="text" name="id_num" id="id_num" />
    </div>
    <div class="field half">
        <label for="email">Email</label>
        <input type="text" name="email" id="email" />
    </div>
    <div class="field">
        <label for="address">Address</label>
        <textarea name="address" id="address" rows="2"></textarea>
    </div>
</div>
<div style="float:right">
    <ul class="actions">
        <li><input id="signup_btn" type="submit" value="Sign Up"
class="primary" /></li>
    </ul>
</div>
</div>
</div>
</article>

```

- css가 깨지므로 form태그에 적용된 css를 class="form"에 맞게 수정

```

...
.form {
    margin: 0 0 2rem 0;
}

```

```
.form > :last-child {
  margin-bottom: 0;
}

.form > .fields {
  display: -moz-flex;
  display: -webkit-flex;
  display: -ms-flex;
  display: flex;
  -moz-flex-wrap: wrap;
  -webkit-flex-wrap: wrap;
  -ms-flex-wrap: wrap;
  flex-wrap: wrap;
  width: calc(100% + 3rem);
  margin: -1.5rem 0 2rem -1.5rem;
}

.form > .fields > .field {
  -moz-flex-grow: 0;
  -webkit-flex-grow: 0;
  -ms-flex-grow: 0;
  flex-grow: 0;
  -moz-flex-shrink: 0;
  -webkit-flex-shrink: 0;
  -ms-flex-shrink: 0;
  flex-shrink: 0;
  padding: 1.5rem 0 0 1.5rem;
  width: calc(100% - 1.5rem);
}

.form > .fields > .field.half {
  width: calc(50% - 0.75rem);
}

.form > .fields > .field.third {
  width: calc(100%/3 - 0.5rem);
}

.form > .fields > .field.quarter {
  width: calc(25% - 0.375rem);
}

@media screen and (max-width: 480px) {
```

```

.form > .fields {
  width: calc(100% + 3rem);
  margin: -1.5rem 0 2rem -1.5rem;
}

.form > .fields > .field {
  padding: 1.5rem 0 0 1.5rem;
  width: calc(100% - 1.5rem);
}

.form > .fields > .field.half {
  width: calc(100% - 1.5rem);
}

.form > .fields > .field.third {
  width: calc(100% - 1.5rem);
}

.form > .fields > .field.quarter {
  width: calc(100% - 1.5rem);
}

}
...

```

- App.js

```

...
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/member_insert', require('./routes/member_insert')); //추가
...

```

- npm i mysql 을 입력해서 mysql 설치
- Routes > member\_insert.js 만들기

```

var express = require('express');
var router = express.Router();

const mysql = require('mysql');

/* POST member_insert */
router.post('/', function(req, res, next) {
  const result = {msg: 'Sign up Error'};

```

```

let conn = mysql.createConnection({
  host : 'localhost',
  user : 'root',
  password : 'mysql',
  database : '2ndproject'
});

conn.connect((err)=>{
  if(err){
    return console.error(err.message);
  }
  console.log("Connected to the MySQL Server");
  const sql = `insert into
member(ID,NAME,PASSWORD,IDENTITY_NUM,PHONE,ADDRESS,EMAIL) values
('${req.body.id}','${req.body.name}','${req.body.password}','${req.body
.id_num}','${req.body.phone}','${req.body.address}','${req.body.email}'
)`;

  console.log(sql);
  conn.query(sql,(err,results,fields)=>{
    if(err){
      console.log(err.message);
      res.json(JSON.stringify(result));
    }else{
      console.log(results,fields);
      result.msg = req.body.name+"님, 가입되었습니다.<br><br>창을 닫고 Login
하세요.";
      req.session.name = req.body.name;
      res.json(JSON.stringify(result));
      console.log(result.msg);
    }
  });
  conn.end((err)=>{
    if(err){
      return console.error(err.message);
    }
  });
});

});

module.exports = router;

```

- signup을 하기 위해서 버튼에 이벤트를 넣어야 한다. 따라서 index.ejs에 my.js를 불러올 수 있는 script 태그 추가

```
...
<!-- Scripts -->
<script src="javascripts/jquery.min.js"></script>
<script src="javascripts/browser.min.js"></script>
<script src="javascripts/breakpoints.min.js"></script>
<script src="javascripts/util.js"></script>
<script src="javascripts/main.js"></script>
<script src="javascripts/my.js"></script>    <!--추가-->
...
```

- javascripts > my.js

```
$(document).ready(function(){
  $("#signup_btn").click(function(){
    const id = $("#id").val();
    const name = $("#name").val();
    const password = $("#password").val();
    const id_num = $("#id_num").val();
    const phone = $("#phone").val();
    const address = $("#address").val();
    const email = $("#email").val();
    const send_params = {
      id,
      name,
      password,
      id_num,
      phone,
      address,
      email
    };
    $.post('/member_insert', send_params, function(data, status){
      const parsed_data = JSON.parse(data);
      $("#sign_up_div").html("<h1>" + parsed_data.msg + "</h1>");

    }); //post
  }); //signup_btn
});
```

- npm i express-session 세션값을 넣기 위해 설치

- app.js

```
...
app.use(logger('dev'));
app.use(express.static(path.join(__dirname, 'public')));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser('secret code'));
app.use(require('express-session')({
  resave:false,
  saveUninitialized:true,
  secret: 'secret code',
  cookie:{
    httpOnly:true,
    secure:false
  }
}));
...
```

## 로그인

- app.js

```
...
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/member_insert',require('./routes/member_insert'));
app.use('/login',require('./routes/login')); //추가
...
```

- index.ejs

```
...
<nav>
  <ul>
    <li><a href="#intro">We are</a></li>
    <li><a href="#login">Login</a></li> <!--수정-->
    <li><a href="#signup">Sign Up</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</nav>
...
```



```

<!-- Login -->
<article id="login">
  <h2 class="major">Login</h2>
  <div id="login_div" class="form" action="#">
    <div class="fields">
      <div class="field">
        <label for="ID">ID</label>
        <input type="text" name="login_id" id="login_id" />
      </div>
      <div class="field">
        <label for="email">Password</label>
        <input type="password" name="login_pwd" id="login_pwd" />
      </div>
    </div>
    <div style="float:right">
      <ul class="actions">
        <!-- <li><input type="reset" value="Reset" /></li> -->
        <li><input id="login_btn" type="submit" value="Login"
class="primary" /></li>
      </ul>
    </div>
  </div>
</article>

```

■ routes > login.js

```

var express = require('express');
var router = express.Router();
const mysql = require('mysql'); //1. 드라이버 등록

/* GET users listing. */
router.post('/', function(req, res, next) {
  const result = {msg:""};

  let conn = mysql.createConnection({ //2. 연결
    host : 'localhost',
    user : 'root',
    password : 'mysql',
    database : '2ndproject',
  });

  conn.connect((err)=>{
    if(err){
      return console.error(err.message);
    }
  });

```

```

}
console.log("Connected to the MySQL Server : ",req.body.login_id);
const sql = `select * from member where id = '${req.body.login_id}'
and password = '${req.body.login_pwd}'`;

console.log(sql);
conn.query(sql,(err,rs,fields)=>{
  if(err){
    console.log(err.message);
    result.msg='SSORRY';
    res.json(JSON.stringify(result));
  }else{
    if(rs[0]){      //로그인 OK //rs[0]이 존재하면서 &&
      console.log(rs[0].ID,rs[0].PASSWORD);
      req.session.login_id=req.body.login_id;
      req.session.name = rs[0].NAME;

      const s_num = rs[0].IDENTITY_NUM.split("");
      let id_num="";
      for(i=0; i<8; i++)
      {
        id_num += s_num[i];
      }
      let id_num1 = rs[0].IDENTITY_NUM;
      let date;
      let yy = id_num1.substr(0,2);
      let mm = id_num1.substr(2,2);
      let dd = id_num1.substr(4,2);
      date=yy+"."+mm+"."+dd;

      req.session.id_num = id_num+"*****";
      req.session.id_num1 = date;
      req.session.password = rs[0].PASSWORD;
      req.session.phone = rs[0].PHONE;
      req.session.address = rs[0].ADDRESS;
      req.session.email = rs[0].EMAIL;
      req.session.loginState = true;
      req.session.photo_path = rs[0].PHOTO_PATH;

      result.msg=req.session.name+"님, 반갑습니다.";
      res.json(JSON.stringify(result));
    }else{ //로그인 실패
      result.msg="Try Again";
      res.json(JSON.stringify(result));
    }
  }
}

```

```

    }
    conn.end((err)=>{
      if(err){
        return console.error(err.message);
      }
      console.log("conn close");
    });
  });
});
});

module.exports = router;

```

- Login\_btn 클릭했을때 이벤트 my.js에 설정

```

$('#login_btn').click(function(){
  const login_id = $('#login_id').val();
  const login_pwd = $('#login_pwd').val();

  const send_params = {
    login_id,
    login_pwd
  }

  $.post('/login', send_params, function(data, status){
    try{
      const parsed_data = JSON.parse(data);

      alert(parsed_data.msg);
      window.location.assign("/");
    } catch(err){
      window.location.reload(true);
    }
  }); // post
}); // login_btn

```

- 로그인 했을때 index.ejs 메인페이지로 돌아가는데 메인 페이지에 이름이랑 주민번호 뜨게 설정하기 index.esj

```

...
<div class="content">
  <div class="inner">
    <%
      if(loginState){
        %>

```

```

    <h1><%= name%></h1>
    <p><%= id_num %></p>
    <%
    }else{
        %>
        <h1>TIFFANY</h1>
        <p>Ticket Is Fair For Anyone aNd You</p>
        <%
        }
        %>
    </div>
</div>

...

```

#### ■ index.js

```

var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { loginState:req.session.loginState,
    id:req.session.login_id,
    name:req.session.name,
    password:req.session.password,
    id_num:req.session.id_num,
    id_num1:req.session.id_num1,
    phone:req.session.phone,
    address:req.session.address,
    email:req.session.email,
    photo_path:req.session.photo_path,
    newpath:req.session.newpath });
});

module.exports = router;

```

로그인 했을 때 메뉴 다르게 보이기

#### ■ index.ejs

```

...
<nav>
  <ul>
    <%

```



```

        <tr><td>EMAIL</td><td><%= email%></td></tr>
        <tr><td>ADDRESS</td><td><%= address%></td></tr>
    </table>
    <ul>
        <button style="float:right;"id="modify_btn"><a
href="#modify">Modify</a></button>
    </ul>
</nav>
</article>
<%
}else{
    %>
<article id="intro">
    <h2 class="major">WE ARE</h2>
    <span class="image sub"></span>
    <table>
        <th colspan="2">TIFFANY</th>
        <tr><td class="rowS" rowspan="2">ID CHECK PAGE</td><td>김민재</td>
    </tr>
        <tr><td>최재성</td></tr>
        <tr><td class="rowS" rowspan="2">TICKET PAGE</td><td>윤영욱</td></tr>
        <tr><td>신영찬</td></tr>
    </table>

</article>
<%
}
%>

```

## 로그아웃

### ■ App.js

```

...
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/member_insert',require('./routes/member_insert'));
app.use('/login',require('./routes/login'));
app.use('/logout',require('./routes/logout')); //추가
...

```

### ■ Routes > logout.js

```

var express = require('express');
var router = express.Router();

/* GET users listing. */
router.get('/', function(req, res, next) {
  const result = {msg:""};
  console.log("Session ID login.js => "+req.sessionID);
  if(req.session.login_id){
    req.session.destroy(function(err){
      res.redirect('/');
    });
  }
});

module.exports = router;

```

## 회원 정보 수정 페이지

### ■ app.js

```

...
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/member_insert',require('./routes/member_insert'));
app.use('/login',require('./routes/login'));
app.use('/logout',require('./routes/logout'));
app.use('/modify',require('./routes/modify'));
...

```

### ■ routes > modify.js

```

var express = require('express');
var router = express.Router();

const mysql = require('mysql');

/* GET users listing. */
router.post('/', function(req, res, next) {
  const result = {msg:'Modify Error'};

  let conn = mysql.createConnection({
    host : 'localhost',
    user : 'root',
    password : 'mysql',

```

```

        database : '2ndproject'
    });

    conn.connect((err)=>{
        if(err){
            return console.error(err.message);
        }
        console.log("Connected to the MySQL Server");
        const sql = `update member set PASSWORD = '${req.body.password}',
NAME = '${req.body.name}', PHONE = '${req.body.phone}', EMAIL =
'${req.body.email}', ADDRESS = '${req.body.address}' where ID =
'${req.session.login_id}'`;

        console.log(sql);
        conn.query(sql,(err,result,fields)=>{
            if(err){
                console.log(err.message);
                res.json(JSON.stringify(result));
            }else{
                console.log(result,fields);
                req.session.loginState=false;
                res.json(JSON.stringify(result));
                result.msg = "수정 완료";
                console.log(result.msg);
            }
        });
        conn.end((err)=>{
            if(err){
                return console.error(err.message);
            }
        });
    });
});

module.exports = router;

```

■ javascripts > my.js

```

...
$("#modify_submit_btn").click(function(){
    const password = $("#m_password").val();
    const name = $("#m_name").val();
    const phone = $("#m_phone").val();

```



```

const email = $("#m_email").val();
const address = $("#m_address").val();

const send_params = {
  password,
  name,
  phone,
  email,
  address
};
$.post('/modify', send_params, function(data, status){
  const parsed_data = JSON.parse(data);

  window.location.assign("/");

}); //post
}); //modify_submit_btn
...

```

## 삭제(탈퇴)하기

- app.js

```

...
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/member_insert', require('./routes/member_insert'));
app.use('/login', require('./routes/login'));
app.use('/logout', require('./routes/logout'));
app.use('/modify', require('./routes/modify'));
app.use('/member_delete', require('./routes/member_delete'));
...

```

- routes > member\_delete.js

```

var express = require('express');
var router = express.Router();

const mysql = require('mysql');

/* GET users listing. */
router.get('/', function(req, res, next) {
  const result = {msg: 'Modify Error'};

```

```

let conn = mysql.createConnection({
  host : 'localhost',
  user : 'root',
  password : 'mysql',
  database : '2ndproject'
});

conn.connect((err)=>{
  if(err){
    return console.error(err.message);
  }
  console.log("Connected to the MySQL Server");
  const sql = `delete from member where ID =
'${req.session.login_id}'`;
  console.log(sql);
  conn.query(sql,(err,result,fields)=>{
    if(err){
      console.log(err.message);
      res.json(JSON.stringify(result));
    }else{
      req.session.destroy();
      result.msg = "삭제 완료";
      res.json(JSON.stringify(result));

      console.log(result.msg);
    }
  });
  conn.end((err)=>{
    if(err){
      return console.error(err.message);
    }
  });
});
});
module.exports = router;

```

■ Javascripts > my.js

```

let withdrawYn1 = false;
let withdrawYn2 = false;
let withdrawYn3 = false;
$("#delete_btn").click(function(){
  withdrawYn1 = confirm("탈퇴하실건가요?");

```

```

if (withdrawYn1) {
    withdrawYn2 = confirm("진지하게?");

    if (withdrawYn2) {
        withdrawYn3 = confirm("아 진짜로?");

        if (withdrawYn3) {
            $.get("/member_delete", (data, status) => {
                const parsedData = JSON.parse(data);

                alert(parsedData.msg);

                window.location.href = "/";
            });
        }
    }
} else {
    window.location.href = "/#modify";
}
});

```

## 사진 업로드

- app.js

```

...
app.post('/upload', function (req, res) {
    console.log("=====");
    console.log(req.session.photo_path);

    var form = new formidable.IncomingForm();
    form.parse(req, function (err, fields, files) {
        var oldpath = files.id_photo_name.path;
        var newpath = 'public/upload/' + req.session.login_id + '_'
+files.id_photo_name.name;
        req.session.newpath=newpath;
        req.session.loginState=false;
        fs.rename(oldpath, newpath, function (err) {
            if (err) throw err;
            res.redirect("/");
            res.end();
        });
    });
});

```

```
...
```

- index.ejs에 추가

```
...
<!-- photo_update -->
<article id="id_update">
  <h2 class="major"><%=name%>'s Photo Update</h2>
  <form action="/upload" method='post' enctype="multipart/form-data"
class="form">
    <div class="field form">
      <label for="photo">ID Photo Upload</label>
      <input type="file" name="id_photo_name" id="id_photo" />
    </div>

    <button type="submit" id="photo_btn" class="primary" />Upload!
  </form>
</article>
...
```

- photo\_btn을 클릭했을때 이벤트 정의 my.js

```
...
$("#photo_btn").click(function(){
  const origin_filepath =
$("#input[name=id_photo_name]").val().split("\\");
  const new_filepath = origin_filepath[origin_filepath.length-1];
  //alert(origin_filepath);
  const send_params = {
    new_filepath
  }
  $.post("/photo_insert",send_params,function(data,status){

    alert(JSON.parse(data).msg);

    window.location.assign("/");
  });
});//photo_btn
...
```

- routes > photo\_insert.js

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

```

const mysql = require('mysql'); //1. 드라이버 등록

/* GET users listing. */
router.post('/', function(req, res, next) {

    let conn = mysql.createConnection({ //2.연결
        host : 'localhost',
        user : 'root',
        password : 'mysql',
        database : '2ndproject'
        // dateStrings: 'date'
    });

    conn.connect((err)=>{
        if(err){
            return console.error(err.message);
        }
        console.log("Connected to the MySQL Server :
",req.session.login_id);
        const filepath = "upload/" + req.session.login_id + "_"
+req.body.new_filepath;
        const sql = `update member set PHOTO_PATH = '${filepath}' where
ID = '${req.session.login_id}'`;

        //req.session.loginState=false;
        conn.query(sql,(err,rs,fields)=>{
            if(err){
                console.log(err.message);

            }else{

                const msg = {msg:"Completed"};
                res.json(JSON.stringify(msg));
            }
        })
        conn.end((err)=>{
            if(err){
                return console.error(err.message);
            }
            console.log("conn close");
        });
    });
});
});
});

```

