app.js

|  |
| --- |
| var createError = require('http-errors');  var express = require('express');  var path = require('path');  var cookieParser = require('cookie-parser');  var logger = require('morgan');  var session = require('express-session');  var app = express();  // view engine setup  // 기본 path를 SPA/views로 지정함  // view의 engine은 ejs로 설정함  app.set('views', path.join(\_\_dirname, 'views'));  app.set('view engine', 'ejs');  app.engine('html', require('ejs').renderFile);  app.use(logger('dev'));  app.use(express.json());  app.use(express.urlencoded({ extended: false }));  app.use(cookieParser('first project SPA')); // 암호화 키 추가  app.use(express.static(path.join(\_\_dirname, 'public')));  app.use(session({  resave: false,  saveUninitialized: false, // session의 내용 변경이 없더라도 저장을 할것인가?  secret: 'first project SPA', // 필수적으로 넣어야하며, 위의 cookieParser의 인자와 같은 값을 넣어야함  cookie: {  httpOnly: true,  secure: false  }  }));  // localhost:3000/'첫번째인자'가 url로 들어온다면  // require안의 인자로 routing을 시켜줌  app.use('/', require('./routes/index'));  app.use('/users', require('./routes/users'));  app.use('/signup', require('./routes/signup'));  app.use('/login', require('./routes/login'));  app.use('/logout', require('./routes/logout'));  app.use('/search', require('./routes/search'));  // catch 404 and forward to error handler  app.use(function(req, res, next) {  next(createError(404));  });  // error handler  app.use(function(err, req, res, next) {  // set locals, only providing error in development  res.locals.message = err.message;  res.locals.error = req.app.get('env') === 'development' ? err : {};  // render the error page  res.status(err.status || 500);  res.render('error');  });  module.exports = app; |

index.ejs

|  |
| --- |
| <!DOCTYPE html>  <html>      <head>          <title>SPA 1st Project Page</title>          <link rel='stylesheet' href='/stylesheets/style.css' /> <!-- css 가져오기 -->          <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script> <!-- jQuery CDN 가져오기 -->          <script src="javascripts/listener.js"></script> <!-- callback 함수로 이루어진 listener.js file 가져오기 -->            <style>              /\* index.ejs에서 쓰일 style이기에 css file에 안넣어 놓았으며, 배경이미지로 가져옴 \*/              body {                  background-image: url('/images/noIndexBackground.PNG');                  background-repeat: no-repeat;                  background-position: center;                  background-position: top;              }          </style>      </head>      <body>  <div>              <!-- 항목을 위에서 볼 수 있도록 하기 위해 설정한 div tag-->              <div id = 'backgroundBlack'>Click the button you want                  <div id = 'container'>                      <a href = '/'><input type = 'button' id = 'home' value = 'HOME'></a>                      <a href = '/signup'><input type = 'button' id = 'signup' value = 'SIGN UP'></a>                      <a href = '/login'><input type = 'button' id = 'login' value = 'LOGIN'></a>                      <a href = '/search'><input type = 'button' id = 'search' value = 'SEARCH'></a>                  </div>              </div>          </div>      </body>  </html> |

signup.ejs

|  |
| --- |
| <!DOCTYPE html>  <html>      <head>          <title>SPA 1st Project Page</title>          <link rel='stylesheet' href='/stylesheets/style.css' />          <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>          <script src="javascripts/listener.js"></script>            <style>              /\* signup.ejs에서 쓰일 style이기에 css file에 안넣어 놓았으며, 배경이미지로 가져옴 \*/              body {                  background-image: url('/images/registerPicture.PNG');                  background-repeat: repeat-y;              }          </style>      </head>      <body>  <div>              <div id = 'backgroundWhite'>Click the button you want                  <div id = 'container'>                      <a href = '/'><input type = 'button' id = 'home' value = 'HOME'></a>                      <a href = '/signup'><input type = 'button' id = 'signup' value = 'SIGN UP'></a>                      <a href = '/login'><input type = 'button' id = 'login' value = 'LOGIN'></a>                      <a href = '/search'><input type = 'button' id = 'search' value = 'SEARCH'></a>                  </div>              </div>          </div>          <div id = "signupDiv">              <!-- signup을 처리하기 위한 div -->              <fieldset>                  <legend><h2>SIGN UP</h2></legend>                  <div id = 'signupAd'>                      <!-- 광고(사진)를 넣기 위한 자리 -->                      <img src = 'images/advertisement.PNG' width = '400px'>                  </div>                  <!-- loginState에 값이 있는 경우이며, 회원 가입 축하와 함께 loginID 출력 -->                  <%                      if(loginState) {                  %>                      <h2>                          회원 가입을 축하드립니다. <%= loginID%>님                      </h2>                  <%                      } else {                  %>                  <!-- loginState에 값이 없는 경우이며, signupContents 부분이 나오게 됨 -->                      <div id = 'signupContents'>                          <div id = 'signupMain'>                              ID<br><input id = 'signupID' placeholder = 'ID'><br> <!-- ID를 입력하는 공간 -->                              Password<br><input id = 'signupPW' type = 'password' placeholder = 'Password'><br> <!-- PWD를 입력하는 공간 -->                              Name<br><input id = 'signupName' placeholder = 'Name'><br> <!-- 이름을 입력하는 공간 -->                              Email<br><input id = 'signupEmail' type = 'email' placeholder = 'abcd@xxx.com'><br> <!-- 이메일을 입력하는 공간 -->                              Phone Number<br><input id = 'signupPhone' placeholder = 'xxx-yyyy-zzzz'><br> <!-- 핸드폰 번호를 입력하는 공간 -->                              <button id = 'signupBtn'>DONE</button> <!-- 모두 입력한 뒤 누르는 버튼 -->                              <button id = 'signupClearBtn'>CLEAR</button> <!-- 입력한 모든 것을 초기화 시키는 버튼 -->                          </div>                      </div>                  <%                      }                  %>              </fieldset>          </div>      </body>  </html> |

login.ejs

|  |
| --- |
| <!DOCTYPE html>  <html>      <head>          <title>SPA 1st Project Page</title>          <link rel='stylesheet' href='/stylesheets/style.css' />          <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  <script src="javascripts/listener.js"></script>  <style>  /\* login.ejs에서 쓰일 style이기에 css file에 안넣어 놓았으며, 배경이미지로 가져옴 \*/  body {  background-image: url('/images/registerPicture.PNG');  background-repeat: repeat-y;  }  </style>      </head>      <body>          <div>  <div id = 'backgroundWhite'>Click the button you want               <div id = 'container'>                   <a href = '/'><input type = 'button' id = 'home' value = 'HOME'></a>                   <a href = '/signup'><input type = 'button' id = 'signup' value = 'SIGN UP'></a>                   <a href = '/login'><input type = 'button' id = 'login' value = 'LOGIN'></a>                  <a href = '/search'><input type = 'button' id = 'search' value = 'SEARCH'></a>               </div>  </div>  </div>  <div id = 'loginDiv'>  <!-- login을 처리하기 위한 div -->  <fieldset>  <legend><h2>LOG IN</h2></legend>  <div id = 'loginAd'>  <img src = 'images/advertisement.PNG' width = '400px'>  </div>  <%  if(loginState) {  %>  <!-- loginState의 값이 있는 경우이며, 이름을 출력하며 Log Out 버튼이 나옴 -->  <h2><%= loginName %>님 환영합니다. </h2>  <button id = 'logoutBtn'>LOG OUT</button>  <%  } else {  %>  <!-- loginState의 값이 없는 경우이며, 로그인에 필요한 ID와 PWD를 입력하도록 함-->  <div id = 'loginContents'>  <div id = 'loginMain'>  ID<br><input id = 'loginID' placeholder = 'ID'><br> <!-- ID를 입력하는 공간 -->  Password<br><input id = 'loginPW' type = 'password' placeholder = 'Password'><br> <!-- PWD를 입력하는 공간 -->  <button id = 'loginBtn'>LOG IN</button> <!-- 모두 입력한 뒤 누르는 버튼 -->  <button id = 'loginClearBtn'>CLEAR</button> <!-- 입력한 모든 것을 초기화 시키는 버튼 -->  </div>  </div>  <%  }  %>  </fieldset>  </div>      </body>  </html> |

search.ejs

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>SPA 1st Project Page</title>  <link rel='stylesheet' href='/stylesheets/style.css' />  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  <script src="javascripts/listener.js"></script>  </head>  <body>  <div>  <div id = 'backgroundWhite'>Click the button you want  <div id = 'container'>  <a href = '/'><input type = 'button' id = 'home' value = 'HOME'></a>  <a href = '/signup'><input type = 'button' id = 'signup' value = 'SIGN UP'></a>  <a href = '/login'><input type = 'button' id = 'login' value = 'LOGIN'></a>  <a href = '/search'><input type = 'button' id = 'search' value = 'SEARCH'></a>  </div>  </div>  </div>  <div id = "searchDiv">  <!-- search를 처리하기 위한 div -->  <fieldset>  <legend><h2>Search</h2></legend>  <div>  <input placeholder = 'Search' id = 'searchValue'> <!-- 검색 기능을 하는 공간-->  <button type = 'submit' id = 'searchBtn'>Go</button> <!-- 검색창에 해당하는 공간-->  <br />  </div>  <%  if(searchValue) {  %>  <!-- searchValue에 값이 있는 경우이며, searchValue에 대한 검색 결과를 출력한다. -->  <h2><%= searchValue %>에 대한 검색 결과입니다. </h2>  <%  } else {  %>  <!-- searchValue에 값이 없는 경우이며, 검색어를 입력하라고 출력함 -->  <h2>검색어를 입력해 주시기 바랍니다. </h2>  <%  }  %>  </fieldset>  </div>  </body>  </html> |

index.js

|  |
| --- |
| var express = require('express');  var router = express.Router();  /\* GET home page. \*/  router.get('/', function(req, res, next) {  // get 방식으로 오는 경우, index.ejs file로 객체의 값들을 넘겨줌  res.render('index', {  loginState : req.session.loginState,  loginID : req.session.userid,  loginPW : req.session.userpw,  loginName : req.session.name,  loginEmail : req.session.email,  loginPhone : req.session.phone,  });  });  module.exports = router; |

signup.js

|  |
| --- |
| var express = require('express');  var router = express.Router();  router.get('/', (req, res, next) => {  // get 방식으로 오는 경우, signup.ejs file로 객체의 값들을 넘겨줌  res.render('signup', {  loginState : req.session.loginState,  loginID : req.session.userid,  loginPW : req.session.userpw,  loginName : req.session.name,  loginEmail : req.session.email,  loginPhone : req.session.phone,  });  });  router.post('/', (req, res, next) => {  // post 방식으로 오는 경우, req.body 안에 담긴 값들을 req.session 에 매칭되는 곳으로 넘겨줌  req.session.userid = req.body.userid;  req.session.userpw = req.body.userpw;  req.session.name = req.body.name;  req.session.email = req.body.email;  req.session.phone = req.body.phone;  const result = {  txt : '회원 가입을 축하드립니다. ' + req.session.name + '님'  };  // result를 JSON 형태로 변경한 뒤 호출된 곳으로 넘겨줌  res.json(JSON.stringify(result));  });  module.exports = router; |

login.js

|  |
| --- |
| var express = require('express');  var router = express.Router();  router.get('/', (req, res, next) => {  // get 방식으로 오는 경우, login.ejs file로 객체의 값들을 넘겨줌  res.render('login', {  loginState : req.session.loginState,  loginID : req.session.userid,  loginPW : req.session.userpw,  loginName : req.session.name,  loginEmail : req.session.email,  loginPhone : req.session.phone,  });  });  router.post('/', (req, res, next) => {  const result = {  txt : ''  };  // session에 담긴 userid와 userpw 2개와 사용자가 입력한 userid와 userpw가 같은지 확인함  // 맞는 경우 loginState 값을 true로 한 뒤, 요청 경로를 /로 재지정함  if((req.session.userid === req.body.userid) && (req.session.userpw === req.body.userpw)) {  req.session.loginState = true;  res.redirect('/');  } else {  // 틀린 경우 result.txt에 2개 중 하나가 틀렸다고 값을 넣어준 뒤,  // result를 JSON 형태로 변경한 뒤 호출된 곳으로 넘겨줌  result.txt = 'ID 혹은 PW가 틀렸습니다. 다시 로그인 해주십시오. ';  res.json(JSON.stringify(result));  }  });  module.exports = router; |

logout.js

|  |
| --- |
| var express = require('express');  var router = express.Router()  router.get('/', (req, res, next) => {  // get 방식으로 오는 경우 수행되며,  // session.email에 값이 있다면 session을 파기하고  // 요청 경로를 /로 재지정함  if(req.session.email) {  req.session.destroy((err) => {  res.redirect('/');  });  }  });  module.exports = router; |

search.js

|  |
| --- |
| var express = require('express');  var router = express.Router();  router.get('/', (req, res, next) => {  // get 방식으로 오는 경우, search.ejs file로 객체의 값들을 넘겨줌  res.render('search', {  loginState : req.session.loginState,  loginID : req.session.userid,  loginPW : req.session.userpw,  loginName : req.session.name,  loginEmail : req.session.email,  loginPhone : req.session.phone,  searchValue : req.session.searchValue,  });  });  router.post('/', (req, res, next) => {  const result = {  txt : ''  };  // 사용자가 검색창에 입력을 한 경우에 해당하며  // session.searchValue에 사용자가 입력한 값을 넣어줌  // 그 후, 요청 경로를 /로 재지정함  if(req.body.searchValue) {  req.session.searchValue = req.body.searchValue;  res.redirect('/');  } else {  // 사용자가 검색창에 입력을 안 한 경우에 해당하며  // session.searchValue에 아무것도 없는 값을 준 뒤,  // 다시 검색을 요청한다는 txt를 넣고,  // result를 JSON 형태로 변경한 뒤 호출된 곳으로 넘겨줌  req.session.searchValue = '';  result.txt = '검색어 입력을 해주십시오. ';  res.json(JSON.stringify(result));  }  });  module.exports = router; |

style.css

|  |
| --- |
| #backgroundWhite {  padding: 15px;  font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;  }  #backgroundBlack {  padding: 15px;  font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;  color: white;  }  div#container a input {  margin-top: 20px;  }  #signupDiv {  margin-left : 425px;  }  div#signupDiv div input {  width : 200px;  }  div#signupAd {  display: inline;  float: right;  }  #loginDiv {  margin-left : 425px;  }  div#loginDiv div input {  width : 200px;  }  div#loginAd {  display: inline;  float: right;  }  div#searchDiv div input {  width : 200px;  }  body input {  background-color: #2F2E2E;  padding: 10px;  font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;  color: white;  border-style: solid;  width : 100px;  text-align: center;  }  :-ms-input-placeholder {  color: #a8a8a8;  }  ::-webkit-input-placeholder {  color: #a8a8a8;  }  :-moz-placeholder {  color: #a8a8a8;  }  body button {  background-color : #2F2E2E;  color : white;  margin-top: 30px;  padding : 10px;  position: center;  width : 150px;  font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;  }  body {  font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;  text-align: center;  }  button#logoutBtn {  color : red;  }  div#inputGroupAddon:hover {  cursor: pointer;  }  legend {  text-align: left;  } |

listener.js

|  |
| --- |
| $(document).ready(() => {  // id = 'signupBtn'인 객체가 눌리면 다음 일을 수행  $('#signupBtn').click(() => {  const userid = $('#signupID').val();  const userpw = $('#signupPW').val();  const name = $('#signupName').val();  const email = $('#signupEmail').val();  const phone = $('#signupPhone').val();  // 하나라도 입력을 안하는 경우, alert창을 띄우고 이벤트 종료  if(userid === '') {  alert('ID를 입력해주세요.');  } else if(userpw === '') {  alert('비밀번호를 입력해주세요. ');  } else if(name === '') {  alert('이름를 입력해주세요. ');  } else if(email === '') {  alert('Email을 입력해주세요. ');  } else if(phone === '') {  alert('핸드폰 번호를 입력해주세요. ');  } else {  // 모두 입력한 경우, 입력한 것을 객체의 key와 value로 만듬  sendParams = {  userid,  userpw,  name,  email,  phone  };  // 해당 객체를 post 방식으로 전송하며, app.js -> routes/signup.js로 가게 되고,  // 해당 결과를 객체 형태로 변환하여 parsedData에 넣음  $.post('/signup', sendParams, (data, status) => {  const parsedData = JSON.parse(data);  $('#signupMain').html(`<h2>${parsedData.txt}<br>`);  });  }  });  // id = 'loginBtn'인 객체가 눌리면 다음 일을 수행  $('#loginBtn').click(() => {  const userid = $('#loginID').val();  const userpw = $('#loginPW').val();    // 하나라도 입력을 안하는 경우, alert창을 띄우고 이벤트 종료  if(userid === '') {  alert('ID를 입력해주세요.');  } else if(userpw === '') {  alert('비밀번호를 입력해주세요. ');  } else {  // 모두 입력한 경우, 입력한 것을 객체의 key와 value로 만듬  sendParams = {  userid,  userpw,  };  // 해당 객체를 post 방식으로 전송하며, app.js -> routes/login.js로 가게 되고,  // 해당 결과를 객체 형태로 변환하여 parsedData에 넣음  $.post('/login', sendParams, (data, status) => {  // try block 안에 해당한다면, catch block을 수행하게 함  try {  userid = '';  userpw = '';  alert(JSON.parse(data).txt);  } catch(err) {  window.location.reload(true);  }  });  }  });  // id = 'searchBtn'인 객체가 눌리면 다음 일을 수행  $('#searchBtn').click(() => {  const searchValue = $('#searchValue').val();  sendParams = {  searchValue  }    // 해당 객체를 post 방식으로 전송하며, app.js -> routes/search.js로 가게 됨  $.post('/search', sendParams, (data, status) => {  // try block 안에 해당한다면, catch block을 수행하게 함  try {  searchValue = '';  } catch(err) {  window.location.reload(true);  }  });  });  // id = 'signupClearBtn'인 객체가 눌리면 다음 일을 수행  $('#signupClearBtn').click(() => {  // 해당하는 id들의 값들을 모두 ''로 만듬  $('#signupID').val('');  $('#signupPW').val('');  $('#signupName').val('');  $('#signupEmail').val('');  $('#signupPhone').val('');  });    // id = 'loginClearBtn'인 객체가 눌리면 다음 일을 수행  $('#loginClearBtn').click(() => {  // 해당하는 id들의 값들을 모두 ''로 만듬  $('#loginID').val('');  $('#loginPW').val('');  });    // id = 'logoutBtn'인 객체가 눌리면 다음 일을 수행  $('#logoutBtn').click(() => {  // 2번째 인자를 get 방식으로 전송하며, app.js -> routes/logout.js로 가게 됨  $.get('/logout', '',(data, status) => {  location.reload(true);  });  });  }); |