

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| **WEB STUDY** | | |
| #6 | | |
|  | | |
| 학 부 | 컴퓨터정보공학부 |  |
| 학 번 | 202121243 |  |
| 이 름 | 박이슬 |  |
| 제출일자 | 2022-05-31 |  |

|  |  |  |
| --- | --- | --- |
|  | 목차 |  |
|  |  |
| [1. 조사과제 2](#_Toc104925601)  [1.1. 소주제 1 2](#_Toc104925602)  [2. 실습과제 2](#_Toc104925603)  [2.1. 실습#1 2](#_Toc104925604)  [2.2. 실습#2 3](#_Toc104925605)  [2.3. 실습#3 5](#_Toc104925606)  [3. Node.js 방명록 7](#_Toc104925607)  [3.1. Node.js 방명록 7](#_Toc104925608) | | |

# 조사과제

## Rest.api

REST(Representational State Transfer)

: HTTP URI를 통해 자원을 명시하고 HTTP Method를 통해 해당 자원에 대한 CRUD Operation을 적용하는 것

* CRUD Operation

: 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능인 Create, Read, Update, Delete

* REST에서의 CRUD Operation 동작

1. Create: 데이터 생성(POST)
2. Read: 데이터 조회(GET)
3. Update: 데이터 수정(PUT)
4. Delete: 데이터 삭제(DELETE)

|  |  |
| --- | --- |
| REST API  : REST | |
| 구성 요소 | * 자원(Resource): HTTP URI * 행위(Verb): HTTP Method * 표현(Representations): HTTP Message Pay Load |
| 특징 | * Server-Client(서버-클라이언트 구조) * Stateless(무상태) * Cacheable(캐시 처리 가능) * Layered System(계층화) * Uniform Interface(인터페이스 일관성) * Self-descriptiveness(자체 표현 구조) |
| 장점 | * 별도의 인프라 구축 필요 없음 * HTTP 표준 프로토콜에 따르는 모든 플랫폼에서 사용 가능 * Hypermedia API 기본을 충실히 지키면서 범용성 보장 * 의도하는 바를 쉽게 파악할 수 있음 * 서버와 클라이언트의 역할 명확하게 분리함 |
| 단점 | * 표준이 존재하지 않아 정의가 필요 * 사용 가능한 메소드가 4가지밖에 없음 * HTTP Method 형태가 제한적 |
| 중심 규칙 | 1. URI는 정보의 자원을 표현해야함   GET /example/delete/1   1. 자원에 대한 행위는 HTTP Method로 표현   DELETE /example/1 |
| 설계 예시 | * URI는 동사보다 명사, 대문자보다 소문자를 사용한다   안 좋은 예: http://example.com/running  좋은 예: http://example.com/run |
| * 마지막에 슬래시(/)를 포함하지 않는다.   안 좋은 예: http://example.com/run/  좋은 예: http://example.com/run |
| * 언더바 대신 하이폰을 사용한다.   안 좋은 예: http://example.com/run\_web  좋은 예: http://example.com/run-web |
| * 파일확장자는 URI에 포함하지 않는다.   안 좋은 예: http://example.com/test-image.jpg  좋은 예: http://example.com/test-image |
| * 행위를 포함하지 않는다.   안 좋은 예: http://example.com/delete-content/1  좋은 예: http://example.com/content/1 |

# 실습과제

## 실습#1

[<링 크>](http://junior.catsecurity.net/~snz2262/web6/ex1.php)

<소스코드>

|  |
| --- |
| test1.php |
| <?php  $db\_Host = "localhost";  $db\_Id = "snz2262";  $db\_Pass = "2022Wnsldj!";  $db\_Name = "snz2262";  $conn = mysqli\_connect($db\_Host, $db\_Id, $db\_Pass, $db\_Name);  mysqli\_query($conn, "set session character\_set\_connection=utf8;");  mysqli\_query($conn, "set session character\_set\_results=utf8;");  mysqli\_query($conn, "set session character\_set\_client=utf8;");  //연결 오류시 종료  if (mysqli\_connect\_errno())  die('Connect Error : '.mysqli\_connect\_errno());  else  echo "<script>console.log('Connect Success!');</script>";  ?> |
| ex1.php |
| <html>  <head>  <title>ex1</title>  </head>  <body>  <form method='post' action=''>  이름: <input type='text' name='name'/> <br>  나이: <input type='text' name='age'/> <br>  <input type='submit' name='submit'/>  </form>  <?php  include('./test1.php');  $name = $\_POST['name'];  $age = $\_POST['age'];  $query = "insert into test\_table (name, age) values ('".$name."', '".$age."')";  echo $query;  if ($result = mysqli\_query($conn, $query)) {  while ($row = mysqli\_fetch\_array($result)) {  print\_r($row);  echo "<br>";  }  }  ?>  </body>  </html> |

<출력화면>

|  |
| --- |
| 이름: Opal, 나이: 20 |
|  |

## 실습#2

<소스코드>

|  |
| --- |
| test1.php 코드는 실습#1과 같음 |
| ex2.php |
| <html lang='ko'>  <head>  <meta charset='utf-8'>  <title>ex2</title>  </head>  <body>  <?php include('./test1.php');?>  <table border=1 width=100%>  <tr>  <td>이름</td>  <td>학교</td>  <td>학과</td>  <td>성별</td>  <td>입학년도</td>  <td>이메일</td>  <td>전화번호</td>  <td>생일</td>  <td>학년</td>  <td>학점</td>  <tr>  <?php  $query = "select \* from Old\_Member";  if ($result = mysqli\_query($conn, $query)) {  while ($row = mysqli\_fetch\_array($result)) {  echo "<tr> <td>"  .$row['Name']."</td><td>"  .$row['School']."</td><td>"  .$row['Major']."</td><td>"  .$row['Gender']."</td><td>"  .$row['EnterYear']."</td><td>"  .$row['Email']."</td><td>"  .$row['Phone']."</td><td>"  .$row['Birth']."</td><td>"  .$row['Grade']."</td><td>"  .$row['Score']."</td> </tr>";  }  }  $query = "select \* from New\_Member";  if ($result = mysqli\_query($conn, $query)) {  while ($row = mysqli\_fetch\_array($result)) {  echo "<tr> <td>"  .$row['Name']."</td><td>"  .$row['School']."</td><td>"  .$row['Major']."</td><td>"  .$row['Gender']."</td><td>"  .$row['EnterYear']."</td><td>"  .$row['Email']."</td><td>"  .$row['Phone']."</td><td>"  .$row['Birth']."</td><td>"  .$row['Grade']."</td><td>"  .$row['Score']."</td> </tr>";  }  }  ?>  </table>  </body>  </html> |

<출력화면>

|  |
| --- |
| Old\_Member, New\_Member 전체 출력 |
|  |

## 실습#3

<소스코드>

|  |
| --- |
| db.js |
| const mysql = require("mysql");    const connection = mysql.createConnection( {  host: 'localhost',  user: 'snz2262',  password: '2022Wnsldj!',  port: 3306,  database: 'snz2262'  } );  connection.connect( function ( err ) {  if ( err ) console.log(err);  else console.log( 'Connected!' );  } );  module.exports = connection; |
| app.js |
| const express = require("express");  const ejs = require("ejs");  const path = require("path");  const app = express();  const conn = require("./db.js");  app.listen(8010, function() {  console.log("listening on port 8010");  });  app.use(express.urlencoded({ extended: false}));  app.set("views", path.join(\_\_dirname, "views"));  app.set("view engine", "ejs");  app.get("/", function ( req, res ) {  var sql = "SELECT \* FROM (SELECT \* FROM Old\_Member UNION SELECT \* FROM New\_Member) t";  conn.query(sql, function (err, rows) {  if (err) console.log("query is not excuted. select fail!\n" + err);  else res.render("index.ejs", { list: rows });  });  }); |
| index.ejs |
| <html>  <head>  <title>CAT-Security Node.js & MySQL</title>  <style>  table {  width: 100%;  text-align: center;  border-collapse: collapse;  border-top: 2px solid #ed0606;  }  th,  td {  border-bottom: 1px dashed #001763;  }  </style>  </head>  <body>  <h2>CAT-Security Node.js & MySQL</h2>  <div>  <table>  <thread>  <th> 이름 </th>  <th> 학교 </th>  <th> 전공 </th>  <th> 성별 </th>  <th> 입학년도 </th>  <th> 이메일 </th>  <th> 전화번호 </th>  <th> 생일 </th>  <th> 학년 </th>  <th> 학점 </th>  </thread>  <tbody> <% for(var i of list) { %> <tr>  <td> <%= i.Name %> </td>  <td> <%= i.School %> </td>  <td> <%= i.Major %> </td>  <td> <%= i.Gender %> </td>  <td> <%= i.EnterYear %> </td>  <td> <%= i.Email %> </td>  <td> <%= i.Phone %> </td>  <td> <%= i.Birth %> </td>  <td> <%= i.Grade %> </td>  <td> <%= i.Score %> </td>  </tr> <% } %> </tbody>  </table>  </div>  </body>  </html> |

<출력화면>

|  |
| --- |
| Old\_Member, New\_Member 전체 출력 |
|  |

# Node.js 방명록

## Node.js 방명록

<소스코드>

|  |
| --- |
| **db.js** |
| const mysql = require('mysql');  const connection = mysql.createConnection( {  host: 'localhost',  user: 'snz2262',  password: '2022Wnsldj!',  port: 3306,  database: 'snz2262',  dateStrings: 'date',  multipleStatements: true  });  connection.connect( function(err) {  if (err) console.log(err);  else console.log('Connected!');  });  module.exports = connection; |
| **app.js** |
| const express = require("express");  const ejs = require("ejs");  const path = require("path");  const app = express();  const conn = require('./db.js');  app.listen(8010, function () {  console.log("listening on port 8010");  });  app.use('/Image',express.static(path.join(\_\_dirname,'Image')));  app.use(express.urlencoded({ extended: false }));  app.set("views", path.join(\_\_dirname, "views"));  app.set("view engine", "ejs");  app.get("/", function (req, res) {  var sql = 'SELECT \* FROM guestBook';  conn.query(sql, function (err, rows) {  if (err) console.log("query is not excuted. select fail!\n" + err);  else res.render("index.ejs", { list: rows, form: "" });  });  } );  app.post("/", function (req, res) {  console.log(req.body);  var sql = 'INSERT INTO guestBook (name, nick, content) VALUES (?,?,?);'+'SELECT \* FROM guestBook;';  var params = [req.body.inputName,req.body.inputNick,req.body.inputText];  conn.query(sql, params, function (err, rows) {  if (err) console.log("query is not excuted. insert fail!\n" + err);  else res.render("index.ejs", { list: rows[1], form: "" });  });  } );  app.get("/modify/:no", function (req, res) {  const no = req.params.no;  var sql = 'SELECT \* FROM guestBook;'+'SELECT \* FROM guestBook WHERE no=?;';  conn.query(sql, no, function (err, rows) {  if (err) console.log("query is not excuted. select fail!\n" + err);  else res.render("index.ejs", { list: rows[0], form: rows[1] });  });  });  app.post("/modify/:no", function (req, res) {  const no = req.params.no;  const name = req.body.inputName;  const nick = req.body.inputNick;  const content = req.body.inputText;  var sql = 'UPDATE guestBook SET name=?, nick=?, content=? WHERE no=?;';  var params = [name, nick, content, no];  conn.query(sql, params, function (err, rows) {  if (err) console.log("query is not excuted. modify fail!\n"+err);  else res.redirect("/");  });  });  app.get("/delete/:no", function (req, res) {  const no = req.params.no;  var sql = 'DELETE FROM guestBook WHERE no=?;';  conn.query(sql, no, function (err, rows) {  if (err) console.log("query is not excuted. delete fail!\n" + err);  else res.redirect("/");  });  }); |
| index.ejs |
| <html>  <head>  <title> 방명록 </title>  <link rel="shortcut Icon" href="#">  <style>  \* {  text-align: center;  margin: 0 auto;  font-family: 'GmarketSansMedium';  }  @font-face {  font-family: 'GmarketSansMedium';  src: url('https://cdn.jsdelivr.net/gh/projectnoonnu/noonfonts\_2001@1.1/GmarketSansMedium.woff') format('woff');  font-weight: normal;  font-style: normal;  }  #header {  height: 60px;  line-height: 60px;  background-color: #043F8C;  }  #header\_title {  text-align:center;  color: white;  font-size: 20px;  }  #wrap {  width: 60%;  margin-top: 50px;  padding-top: 70px;  padding-bottom: 70px;  background-image: url('http://20.196.222.67:8010/Image/dot\_patterns.png');  }  #contents {  width: 75%;  background-color: #FFF;  border: 1px solid #0879A6;  }  #contents\_title {  width: 85%;  padding: 10px;  padding-top: 40px;  display: flex;  justify-content: center;  border-bottom: 5px solid #043F8C;  border-radius: 5px;  }  #contents\_title\_square {  width: 10px;  height: 36px;  margin: 0;  background-color: #043F8C;  }  #contents\_title\_text {  margin-left: 10px;  }  #inputField {  width: 85%;  padding: 20px 10px 20px 10px;  margin-bottom: 20px;  border-bottom: 5px solid #043F8C;  border-radius: 5px;  }  input {  text-align: left;  padding: 5 10px 5 10px;  }  td {  vertical-align: top;  padding: 5px 0 5px 0;  font-size: 18px;  }  input[type='text'] {  width: 80%;  color: #333;  border: none;  outline: 2px solid #0869A6;  }  input[type='text']:focus {  outline: 2px solid #F2AC29;  }  textarea {  width: 80%;  padding: 5 10px 5 10px;  text-align: left;  color: #333;  resize: none;  border: none;  outline: 2px solid #0869A6;  }  textarea:focus {  outline: 2px solid #F2AC29;  }  #inputField\_submit {  text-align: center;  width: 100px;  height: 30px;  color: #0869A6;  border: 2px solid #0869A6;  }  #outputField {  width: 85%;  padding: 10px 0 30px 0;  }  .outputField\_box {  width: 100%;  color: #666;  border: 1px solid #d9d9d9;  padding: 10px;  margin-bottom: 10px;  text-align: left;  }  .output\_content {  text-align: left;  margin-top: 8px;  font-size: 14px;  }  .output\_name, .output\_nick, .output\_date, .output\_modify, .output\_delete {  text-decoration: none;  color: #0869A6;  font-size: 14px;  }  </style>  </head>  <body>  <div id="header">  <p id="header\_title">2022 CAT-Security</p>  </div>  <div id='wrap'>  <div id='contents'>  <div id='contents\_title'>  <div id='contents\_title\_square'></div>  <h1 id='contents\_title\_text'> 방명록 </h1>  </div>  <div id='inputField'>  <table width='80%'>  <form method='post' action=''>  <tr><td>이&ensp;&ensp;름:</td><td><input type='text' name='inputName' maxlength='20' value="<% if(form){ %><%= form[0].name %><% } %>"/></td></tr>  <tr><td>닉네임:</td><td><input type='text' name='inputNick' maxlength='20' value="<% if(form){ %><%= form[0].nick %><% } %>"/></td></tr>  <tr><td>내&ensp;&ensp;용:</td><td><textarea name='inputText' cols='30' rows='5' maxlength='255'><% if(form){ %><%= form[0].content %><% } %></textarea></td></tr>  <tr><td colspan="2" style="vertical-align: middle;"><input id="inputField\_submit" type='submit' name='submit' value='제출'/></td></tr>  </form>  </table>  </div>  <div id='outputField'>  <% if (list) { %>    <script>  const outputField = document.getElementById('outputField');  const outputBox = document.getElementsByClassName('outputField\_box');  <% for (var i of list) { %>  outputField.innerHTML = "<div class='outputField\_box'></div>" + outputField.innerHTML;  outputBox[0].innerHTML += "<span class='output\_name'><%= i.name %></span>";  outputBox[0].innerHTML += "<span class='output\_nick'>(<%= i.nick %>) </span>";  outputBox[0].innerHTML += "<span class='output\_date'>| <%= i.date %> |</span>";  outputBox[0].innerHTML += "<a href='/modify/<%= i.no %>' class='output\_modify' role='button'> 수정 |</a>";  outputBox[0].innerHTML += "<a href='/delete/<%= i.no %>' class='output\_delete' role='button'> 삭제</a>";  outputBox[0].innerHTML += "<p class='output\_content'><%= i.content %></p>";  <% } %>  </script>    <% } %>  </div>  </div>  </div>  </body>  </html> |

<출력화면>

|  |  |
| --- | --- |
| 입력 | **수정 입력** |
|  |  |
| 수정 후 | **삭제** |
|  |  |