

**JUNIOR-STUDY 과제 – WEB study 6**

**제출일자: 2022.05.31**

**학과: 생활과학계열**

**학번: 202121001**

**이름: 김한솔**

**목차**

**Ⅰ 실습 1, 2, 3**

**Ⅱ node.js로 방명록 만들기**

**Ⅲ rest api 조사**

**Ⅰ 실습 1**

|  |
| --- |
|  |
| **dbinit.php** |
| <?php  $db\_Host="localhost";  $db\_Id="casol0219";  $db\_Pass="2022Wnsldj!";  $db\_Name="casol0219";  $conn=mysqli\_connect($db\_Host,$db\_Id,$db\_Pass,$db\_Name);  if(mysqli\_connect\_errno())  die('Connect Error : '.mysqli\_connect\_errno());  mysqli\_query($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;");  ?> |
| **ex1.php** |
| <html>  <head>  <title>실습1</title>  </head>  <body>  <form method="post" action="">  <input type="text" id="name" name="name">  <input type="text" id="age" name="age">  <input type="submit" name="submit" value="입력">  </form>  </body>  </html>  <?php  include('./dbinit.php');  $name=$\_POST['name'];  $age=$\_POST['age'];  $query="insert into test\_table (name,age) values ('$name', '$age');";  $result=mysqli\_query($conn,$query);      $query="select \* from test\_table;";  if($result=mysqli\_query($conn,$query)){  while($row=mysqli\_fetch\_array($result)){  print\_r($row);  echo "<br>";  }  }  ?> |

**Ⅰ 실습 2**

|  |
| --- |
| http://junior.catsecurity.net/~casol0219/web5/ex2.php |

|  |
| --- |
|  |
| **dbinit.php** |
| <?php  $db\_Host="localhost";  $db\_Id="casol0219";  $db\_Pass="2022Wnsldj!";  $db\_Name="casol0219";  $conn=mysqli\_connect($db\_Host,$db\_Id,$db\_Pass,$db\_Name);  if(mysqli\_connect\_errno())  die('Connect Error : '.mysqli\_connect\_errno());  mysqli\_query($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;");  ?> |
| **ex2.php** |
| <?php  include('./dbinit.php');  $query="select \* from Old\_Member union select \* from New\_Member";  if($result=mysqli\_query($conn,$query)){  echo "<table border='1'><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>";  $n=1;  while($row=mysqli\_fetch\_array($result)){  echo "<tr>";  echo "<td>".$row['Name']."</td>";  echo "<td>".$row['School']."</td>";  echo "<td>".$row['Major']."</td>";  echo "<td>".$row['gender']."</td>";  echo "<td>".$row['EnterYear']."</td>";  echo "<td>".$row['Email']."</td>";  echo "<td>".$row['Phone']."</td>";  echo "<td>".$row['Birth']."</td>";  echo "<td>".$row['Grade']."</td>";  echo "<td>".$row['Score']."</td>";  echo "</tr>";  $n++;  }  echo "</table>";  }  ?> |

**Ⅰ 실습 3**

|  |
| --- |
|  |
| **app.js** |
| const express=require("express");  const ejs=require("ejs");  const path=require("path");  const app=express();  const conn=require("./db.js");  app.listen(8001,function(){  console.log("listening on port 8001");  });  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 Old\_Member union select \* from New\_Member";  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{  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>  <thead>  <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>  </thead>  <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>  </tr><% } %></tbody>  </table>  </div>  </body>  </html> |

**Ⅱ node.js로 방명록 만들기**

**(test 디렉토리에서 node app.js)**

|  |
| --- |
|  |
| **App.js** |
| const express=require("express");  const ejs=require("ejs");  const path=require("path");  const app=express();  const conn=require("./db.js");  app.listen(8001,function(){      console.log("listening on port 8001");  });  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 order by seq desc";      conn.query(sql,function(err,rows){          if(err) console.log("error\n"+err);          else res.render("main.ejs",{list:rows});      });  });  app.get("/write",function(req,res){      res.render("write.ejs");  });  app.post("/dbsend",function(req,res){      var title=req.body.title;      var name=req.body.name;      var content=req.body.content;      var sql="insert into guestbook(title,name,content)values(?,?,?)";      conn.query(sql,[title,name,content],function(err,rows){          if(err) console.log("error\n"+err);          else res.redirect("/");      });  });  app.get("/edit/:seq",function(req,res){      const seq=req.params.seq;      res.render("edit.ejs",{content:seq});  });  app.post("/editsend",function(req,res){      var title=req.body.title;      var name=req.body.name;      var content=req.body.content;      var seq=req.body.seq;      console.log(seq);      var sql="update guestbook set title=?,name=?,content=? where seq=?";      conn.query(sql,[title,name,content,seq],function(err,rows){          if(err) console.log("error\n"+err);          else res.redirect("/");      });  });  app.get("/delete/:seq",function(req,res){      const seq=req.params.seq;      var sql="delete from guestbook where seq=?";      var params=seq;      conn.query(sql,params,function(err,rows){      });        var sql="select \* from guestbook order by seq desc";      conn.query(sql,function(err,rows){          if(err) console.log("error\n"+err);          else res.render("main.ejs",{list:rows});      });  }); |
| **Main.ejs** |
| <html>      <head>          <title>GUESTBOOK-hansol</title>          <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@500&display=swap" rel="stylesheet">          <style>              body{                  background:linear-gradient(#fed6e3,#C1B3DA);              }              table{                  color:#65479e;                  background-color:white;                  width:80%;                  text-align:center;                  border-collapse:collapse;                  border:1px solid #C1B3DA;              }              th,td{                  font-size:15px;                  font-family: 'Noto Sans KR', sans-serif;                  padding:5px;                  border-color:white;                  border:1px solid #C1B3DA;              }              th{                  font-size:20px;                  background-color:#C1B3DA;              }              .title{                  font-family: 'Noto Sans KR', sans-serif;                  font-weight:bold;                  font-size:60px;                  color:#5f4298;              }              .write{                  font-size:25px;                  font-family: 'Noto Sans KR', sans-serif;                  color:#5f4298;              }              a{                  text-decoration:none;                  border:1px solid #5f4298;                  background-color: white;              }              a:link{color:#5f4298};              a:visited{color:#5f4298};          </style>      </head>      <body>          <br><div class="title" align="center">GUESTBOOK</div><br>          <div class="write" align="center">              <a href="/write"> 글쓰기 </a>          </div><br>          <div align="center">              <table>                  <thead>                      <th width="5%">번호</th>                      <th width="30%">제목</th>                      <th width="10%">작성자</th>                      <th width="35%">내용</th>                      <th colspan="2" width="20%"></th>                  </thead>                  <tbody><% for(var i of list) { %><tr>                      <td><%= i.seq %></td>                      <td><%= i.title %></td>                      <td><%= i.name %></td>                      <td><%= i.content %></td>                      <td><a href="/edit/<%= i.seq %>"> 수정 </a></td>                      <td><a href="/delete/<%= i.seq %>" role="button"> 삭제 </a></td>                  </tr><% } %></tbody>              </table>          </div>      </body>  </html> |
| **Edit.ejs** |
| <html>      <head>          <title>EDIT</title>      </head>      <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@500&display=swap" rel="stylesheet">      <style>          body{              background:linear-gradient(#fed6e3,#C1B3DA);          }          div,input{              font-size:25px;              font-family: 'Noto Sans KR', sans-serif;              padding:5px;              border-color:white;              color:#65479e;          }          p{              background-color: white;          }          input{              font-size:15px;          }      </style>      <body>          <br><br><br><div align="center">              <p>-- 방명록 수정하기 --</p>              <form action="/editsend" method="post">                  제목<br><input type="text" name="title" /><br>                  작성자<br><input type="text" name="name" /><br>                  내용<br><input type="textarea" name="content" /><br>                  <input type="hidden" name="seq" value="<%= content %>" />                  <br><input type="submit" value="글쓰기" />              </form>          </form>      </body>  </html> |
| **Write.ejs** |
| <html>      <head>          <title>WRITE</title>      </head>      <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@500&display=swap" rel="stylesheet">      <style>          body{              background:linear-gradient(#fed6e3,#C1B3DA);          }          div,input{              font-size:25px;              font-family: 'Noto Sans KR', sans-serif;              padding:5px;              border-color:white;              color:#65479e;          }          input{              font-size:15px;          }          p{              background-color: white;          }      </style>      <body>          <br><br><br><div align="center">              <p>방명록 작성하기</p>              <form action="/dbsend" method="post">                  제목<br><input type="text" name="title" /><br>                  작성자<br><input type="text" name="name" /><br>                  내용<br><input type="textarea" name="content" /><br>                  <br><input type="submit" value="글쓰기" />              </form>          </div>      </body>  </html> |

**Ⅲ rest api 조사**

|  |
| --- |
| **Rest**  **Representational State Transfer** |
| **자원을 이름으로 구분하여 해당 자원의 상태를 주고받는 것**  **(=자원의 표현에 의한 상태 전달)**  **구성요소**  **1. 자원 (URI)**  : 모든 자원에는 고유한 ID인 URI가 존재하고, 이 자원은 서버에 존재한다. 클라이언트는 URI를 이용해 자원을 지정하고 해당 자원의 상태에 대한 조작을 서버에 요청한다.  **2. 행위**  : GET, POST, PUT, DELETE  **3. 표현**  : 클라이언트와 서버가 데이터를 주고받는 형태로 대부분 JSON, XML을 이용한다. |
| **특징**  -네트워크 상에서 클라이언트와 서버 사이의 통신 방식 중 하나이다.  -세션이나 쿠키정보를 별도로 저장하고 관리하지 않기 때문에 API 서버는 들어오는 단순히 들어오는 요청만 처리하면 된다. 따라서 서비스의 자유도가 높아지고 서버에서 불필요한 정보를 관리하지 않음으로써 구현이 단순해진다.  -웹의 기존 기술과 HTTP 프로토콜을 그대로 활용하기 때문에 웹의 장점을 최대한 활용할 수 있다. |

|  |
| --- |
| **Api**  **Application Programming Interface** |
| **클라이언트가 자원을 요청할 수 있도록 서버 측에서 제공된 인터페이스**  **데이터와 기능의 집합을 제공하여 컴퓨터 프로그램간 상호작용을 촉진하며, 서로 정보를 교환할 수 있게한다.** |

|  |
| --- |
| **REST API** |
| **REST를 기반으로 API를 구현하는 것**  **-REST는 HTTP 표준을 기반으로 구현하므로, HTTP를 지원하는 프로그램 언어로 서버와 클라이언트를 구현할 수 있다.**  **-각 요청이 어떤 동작이나 정보를 위한 것인지를 그 요청의 모습 자체로 추론할 수 있다.**  **REST API 설계 규칙**  \*URI는 정보의 자원을 표현해야 한다.  \*자원에 대한 행위는 URI에 포함하지 않는다.  1. 슬래시 구분자는 계층 관계를 나타내는데 사용한다.  2. URI 마지막 문자로 슬래시를 포함하지 않는다.  3. - 은 URI 가독성을 높이는데 사용한다.  4. \_ 은 URI에 사용하지 않는다.  5. URI 경로에는 소문자가 적합하다.  6. 파일 확장자는 URI에 포함하지 않는다. |