2022 CAT-CERT 스터디 보고서

컴퓨터정보공학부 202221168 김지혜

2022.05.18#WEB\_study-6

실습#1

소스코드

<?php

include('./dbinit.php');

$name = $\_POST['name'];

$age = $\_POST['age'];

$query="insert into test\_table(name, age) values ('$name', $age);";

if($result = mysqli\_query($conn,$query)){

while($row =mysqli\_fetch\_array($result)){

print\_r($row);

echo "<br>";

}

}

?>

<html>

<head>

<title> #실습1 </title>

</head>

<body>

<form action="" method="post">

<input type="text" name="name"/>

<input type="text" name="age"/>

<input type="submit" value="제출"/>

</form>

</body>

</html>

출력 결과



텍스트이(가) 표시된 사진

자동 생성된 설명

<http://junior.catsecurity.net/~wlgp9324/web6/web1.php>

실습#2

소스코드

|  |
| --- |
| <html>  <head>  <title> 실습#2 </title>  </head>  <body>  <?php  include('./dbinit.php');  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;');  $query ="select \* from (select\* from Old\_Member union all select\*from New\_Member)z";  $result= mysqli\_query($conn,$query);  echo"<table border='1'><tr><th>이름</th><th>학교</th><th>학과</th><th>성별</th><th>입학년도</th><th>이메일</th><th>전화번호</th><th>생일</th><th>학년</th><th>학점</th>";  while($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>';  }  echo "</table>";  ?>  </body>  </html> |

출력결과

테이블이(가) 표시된 사진

자동 생성된 설명

실습#3

소스코드

<app.js>

|  |
| --- |
| const express = require("express");  const ejs = require("ejs");  const path = require("path");  const app = express();  const conn = require("./db.js");  app.listen(8007, function() {  console.log("listening on port 8007");  });  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 all select\*from New\_Member)z;";  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>  <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>  <th> 학점</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>  <td> <%= i.Score%></td>  </tr> <% } %> </tbody>  </table>  </div>  </body>  </html> |

<db.js>

|  |
| --- |
| const mysql = require("mysql");    const connection = mysql.createConnection({  host:'localhost',  user:'wlgp9324',  password:'2022Wnsldj!',  port: 3306,  database:'wlgp9324'  });  connection.connect(function(err){  if(err)console.log(err);  else console.log('Connected!');  });  module.exports=connection; |

출력결과

테이블이(가) 표시된 사진

자동 생성된 설명

조사과제

REST : 자원을 이름으로 구분하여 해당자원의 상태를 주고받는 모든 것을 의미함

;소프트웨어가 관리하는 모든 것을 표현하기 위한 이름에 의해 데이터가 요청되어지는 시점에서 자원의 상태를 전달함

-> 기본적으로 웹의 기존 기술과 http프로토콜을 그대로 활용함

-> 네트워크 상에서 클라이언트와 서버사이의 통신 방식 중하나

API :데이터와 기능의 집합을 제공하여 컴퓨터 프로그램간 상호작용 촉진, 정보 교환 가능

Rest API

-사내 시스템들도 rest 기반을 시스템을 분산하여 확장성과 재사용성을 높여 유지보수 및 운용편리하게함

-REST는 http표준을 기반으로 구현;서버,클라이언트 구현 O

-델파이 클라이언트, 자바, C#,웹을 통해 제작 가능

|  |  |
| --- | --- |
| 장점 | 단점 |
| -HTTP프로토콜의 인프라를 그대로 사용하여 별도 인프라 구출할 필요가 없음  ;모든 플랫폼 사용가능  -메시지가 의도하는 바를 명확하게 나타내므로 의도하는 바를 쉽게 파악할 수 있음  -서버와 클라이언트 역할을 명확하게 분리함 | -표준이 존재하지 않음  -사용할 수 있는 method가 4가지(get, post, delete,put); 형태가 제한적  - 브라우저 서비스의 경우 쉽게 고칠 수 있는 url보다 header값이 어렵게 느껴짐  -구형 브라우저가 제대로 지원해주지 못하는 부분 존재  (put, delete X, pushState X) |

REST API 설계 기본 규칙

\*리소스 원형

도큐먼트 : 객체 인스턴스나 데이터베이스 레코드와 유사한 개념

컬렉션: 서버에서 관리하는 디렉터리라는 리소스

스토어: 클라이언트에서 관리하는 리소스 저장소

1. URI는 정보의 자원을 표현해야함

-리소스는 동사<명사, 대문자<소문자

-도큐먼트 이름->단순명사

-컬렉션 이름->복수명사

-스토어 이름->복수명사

1. 자원에 대한 행위는 http method로 표현

-URI에 http method가 들어가면 안됨

-URI행위에 대한 동사 표현은 안됨

(GET/members/show/1 -> GET/members/1)

-경로 부분 중 변하는 부분은 유일한 값으로 대체

(id는 특정 리소스를 나타내는 고유값)

1. 슬래시 구분자(/)는 계층 관계
2. URI마지막 문자로 슬래시X
3. 하이픈은(-) 가독성을 높이는데 사용
4. 밑줄(\_)은 사용안함
5. 소문자사용
6. 파일 확장자는 URI에 포함 안함
7. 리소스간 연관관계가 있는 경우

“/리소스명/리소스id/관계가 있는 다른 리소스명”