

Baskin Robbins Kiosk

JAVA

박경호

JAVA기반 스마트 웹&앱 개발자 양성

목차

01 SCHEDULE p. 3 - 4

- 진행 일정

02 STRUCTURE p. 5 - 7

- 폴더 구조 설명
- MVC 구조 설명

03 CODE p. 8 - 19

- 코드 설명

04 REVIEW p. 20 - 21

- 후기

01

SCHEDULE

SCHEDULE

진행일정 (11.07 - 11.12)



11.07

MVC 구조 구상

11.08 - 11.09

JSP 변환
AJAX 변환
Node.JS 변환

11.10 - 11.11

SSH 연결
외부 DB 연결

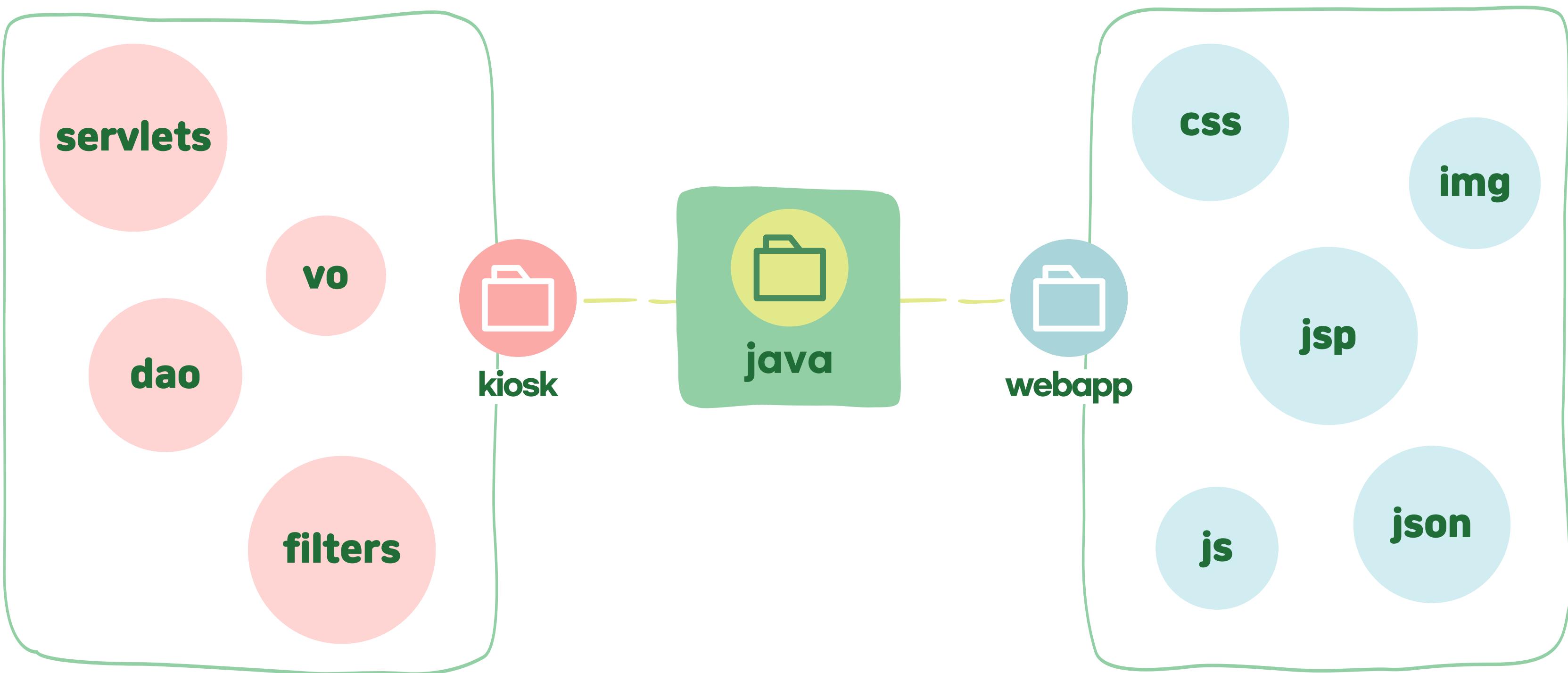
11.12

테스트
PPT 작업

02

STRUCTURE

STRUCTURE FOLDER



STRUCTURE

MVC

.com

MODEL

외부 DB에서 데이터를 전달받아
용도에 맞게 VO, Controller에 전달
DB의 CRUD를 담당

VIEW

JSP을 통하여 데이터를 화면에 표시
VO에서 필요한 데이터를 받거나
CSS, JS을 연결시켜 웹페이지를
동적으로 표현

CONTROLLER

입력받은 동작의 역할을 Servlet에서
VIEW와 MODEL을 구분하여 실행
SSH, DB에 초기 접속을 진행

03

CODE

CODE _ JS

KioskPage.js

- ◆ 기존 XMLHttpRequest 방식 Deprecated 발생
- ◆ jQuery AJAX으로 변경 작업 진행

기존 AJAX

```

let xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
    if(this.readyState == 4 && this.status == 200) {
        let memberList = JSON.parse(this.responseText);
        memberList.forEach((v,i,a)=>{
            if(v.number == memberInfo.getNumber) {
                memberInfo.setPoint = v.point;
                memberInfo.setId = v.id;
                checkNumberJSON = 1;
            }
        })
        if(checkNumberJSON != 1)
            memberInfo.setPoint = 1000;
    }
}
xmlhttp.open("GET", "./src/memberList.json");
xmlhttp.send();

```

변경 AJAX

```

$.ajax({
    url: './json/memberList.json',
    dataType: 'json',
    async: false,
    type: 'GET',
    method: 'GET',
    success: function(data) {
        console.log('okok')
        data.forEach((v,i,a)=>{
            if(v.number == memberInfo.getNumber) {
                memberInfo.setPoint = v.point;
                memberInfo.setId = v.id;
                checkNumberJSON = 1;
            }
        })
        if(checkNumberJSON != 1){
            memberInfo.setPoint = 1000;
        }
    },
    error: function(request, status, error) {
        console.log(request.responseText)
    }
});

```

CODE _ JSP

KioskPage.jsp

— — —

◆ VO에서 Member Class Import 진행

◆ HttpServlet에 저장된 Member Class 가져오기

◆ 가져온 Member Class의 Getter Method 실행

```

1 <%@ page import="kiosk.vo.Member" %>
2 <%@ page language="java" contentType="text/html; charset=UTF-8"
3   pageEncoding="UTF-8"%>
4 <!DOCTYPE html>
5<html lang="KO">
6<head>..
39<body>
40<section id="pageHome">..
51<section id="pageProduct">..
120<section id="pageOptionCup">..
156<section id="pageOptionIce">..
192<section id="pageOptionCake">..
229<section id="pageOptionCoffee">..
265<section id="flavorPopupArea">..
276<section id="pageFlavor">..
327<section id="pageCart">..
398<section id="pageOptionPoint">..
437<section id="pageInputNum">..
503<section id="pagePointView">..
521<section id="pageUsePoint">..
598<section id="pageFinalCost">..
644<section id="pageCard">..
674<section id="pageFinish">..
696<section id="cardPopupArea">..
707 <% Member memberCount = (Member) request.getAttribute("memberCount"); %>
708<section id="hiddenData">
709<form method="post">
710  <input id="hiddenId" type="text" name="id">
711  <input id="hiddenNumber" type="text" name="number">
712  <input id="hiddenPoint" type="text" name="point">
713  <input id="hiddenDate" type="text" name="date">
714  <input id="hiddenDay" type="text" name="day" value="<%=(memberCount.getDayMax()%)">
715  <input id="hiddenCount" type="text" name="count" value="<%=(memberCount.getCount()%)">
716  <input id="hiddenKeyCount" type="text" name="keyCount" value="1">
717  <input id="hiddenKey" type="text" name="key">
718  <input id="adminDataList" type="text" name="adminDataList">
    <input id="hiddenBtn" type="submit">
```

CODE _ VO

Admin.java, Member.java

- ◆ 매출 정보를 담은 Table의 Data VO 생성
- ◆ 회원 정보를 담은 Table의 Data VO 생성
- ◆ 각 VO에 Getter, Setter Method 구현

```

public class Admin {
    protected int id;
    protected String category;
    protected String menu;
    protected int count;
    protected long price;
    protected String date;
    public Admin(int _id, String _category, String _menu, int _count) {
        this.id = _id;
        this.category = _category;
        this.menu = _menu;
        this.count = _count;
        this.price = _price;
        this.date = _date;
    }
    public int getId() { ... }
    public String getCategory() { ... }
    public String getMenu() { ... }
    public int getCount() { ... }
    public long getPrice() { ... }
    public String getDate() { ... }
}

public class Member {
    // DB DATA 변수
    protected int id;
    protected String number;
    protected int point;
    protected int count;
    protected String date;
    public Member() { ... }
    public Member(int _id, String _number, int _point, int _count) {
        this.id = _id;
        this.number = _number;
        this.point = _point;
        this.count = _count;
    }
    public int getId() { ... }
    public String getNumber() { ... }
    public int getPoint() { ... }
    public int getCount() { ... }
    public String getDate() { ... }

    // Back-end 활용 변수
    protected int countMax = 0;
    protected int targetId = 1;
    protected String dayMax = "";
    public int getCountMax() { ... }
    public int getTargetId() { ... }
    public String getDayMax() { ... }
    public void setCountMax(int _countMax) { ... }
    public void setTargetId(int _targetId) { ... }
    public void setDayMax(String _dayMax) { ... }
}

```

관리자 Table VO

회원 Table VO

CODE _ DAO

MemberDao.java - 1

- ◆ setConnection()
 - ◆ MemberDao Connection 연결 Method 생성

- ◆ selectMemberList()
 - ◆ 회원 정보 Table의 값을 JSON 형태로 구성 후 저장

```

class MamberDao {
    Connection connection;
    public void setConnection(Connection connection) {
        this.connection = connection;
    }
    public List<Member> selectMemberList() throws Exception {
        PreparedStatement stmt = null;
        ResultSet rs = null;
        System.out.println("selectMemberList() 호출됨");
        try {
            stmt = this.connection.prepareStatement("SELECT * FROM KioskMember");
            rs = stmt.executeQuery();
            SimpleDateFormat simpleDate = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'");
            ArrayList<Member> members = new ArrayList<Member>();
            while(rs.next()) {
                members.add(new Member(rs.getInt("id"), rs.getString("number"), rs.getInt("point"),
                    rs.getInt("count"), simpleDate.parse(rs.getString("date"))));
            }
            int memberListCount = 0;
            String memberListString = "[";
            for(Member member : members) {
                memberListString += ("{\"id\": " + member.getId() + ", ");
                memberListString += ("\"number\": \"\" + member.getNumber() + "\", ");
                memberListString += ("\"point\": " + member.getPoint() + ", ");
                memberListString += ("\"count\": " + member.getCount() + ", ");
                memberListCount++;
                if(memberListCount == (members.size())) {
                    memberListString += ("\"date\": \"\" + member.getDate() + \"}\"");
                } else {
                    memberListString += ("\"date\": \"\" + member.getDate() + \"}, ");
                }
            }
            memberListString += "]";
            Writer writer = new FileWriter("C:\\\\Users\\\\parkgh\\\\eclipse-workspace\\\\kiosk\\\\src\\\\main\\\\java\\\\com\\\\kiosk\\\\member\\\\MemberList.txt");
            writer.write(memberListString);
        }
    }
}

```

CODE _ DAO

MemberDao.java - 2

— — —

- ◆ maxMemberList()
- ◆ 주문번호 관리 위한 마지막 주문 회원 정보 읽어오기
- ◆ 마지막 번호, 일자 VO에 Setter 실행 후 VO Return

```

    lc Member maxMemberList() throws Exception {
        PreparedStatement stmt_1 = null;
        PreparedStatement stmt_2 = null;
        ResultSet rs_1 = null;
        ResultSet rs_2 = null;
        System.out.println("maxMemberList() 호출됨");
        try {
            stmt_1 = this.connection.prepareStatement(
                "SELECT count 'max', id FROM KioskMember WHERE count = (SE
            );
            rs_1 = stmt_1.executeQuery();
            rs_1.next();
            Member memberCount = new Member();
            memberCount.setCountMax(rs_1.getInt("max"));
            memberCount.setTargetId(rs_1.getInt("id"));
            stmt_2 = this.connection.prepareStatement("SELECT date FROM KioskM
            rs_2 = stmt_2.executeQuery();
            rs_2.next();
            SimpleDateFormat simpleDate = new SimpleDateFormat("yyyy-MM-dd'T'
            memberCount.setDayMax( simpleDate.format(rs_2.getDate("date")) );
            return memberCount;
        } catch(Exception e) {
            throw e;
        } finally {
            try {if (rs_1 != null) rs_1.close();} catch(Exception e) {}
            try {if (stmt_1 != null) stmt_1.close();} catch(Exception e) {}
            try {if (rs_2 != null) rs_2.close();} catch(Exception e) {}
            try {if (stmt_2 != null) stmt_2.close();} catch(Exception e)
        }
    }
}

```

CODE _ DAO

AdminDao.java - 1

— — —

◆ setConnection()

- ◆ AdminDao Connection 연결 Method 생성

◆ selectAdminList()

- ◆ 회원 정보 Table의 값을 JSON 형태로 구성 후 저장
한글 깨짐 문제로 BufferedWriter로 UTF-8 설정

```

public void setConnection(Connection connection) {
    this.connection = connection;
}

public List<Admin> selectAdminList() throws Exception {
    PreparedStatement stmt = null;
    ResultSet rs = null;
    System.out.println("selectAdminList 호출");
    try {
        stmt = this.connection.prepareStatement("SELECT * FROM KioskAdmin");
        rs = stmt.executeQuery();
        SimpleDateFormat simpleDate = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'");
        ArrayList<Admin> admins = new ArrayList<Admin>();
        while(rs.next()) {
            admins.add(new Admin(rs.getInt("id"), rs.getString("category"), rs.getString("menu"),
                rs.getInt("count"), rs.getDouble("price")));
        }
        int adminListCount = 0;
        String adminListString = "[";
        for(Admin admin : admins) {
            adminListString += ("{\"id\": " + admin.getId() + ", ");
            adminListString += ("\"category\": \"\" + admin.getCategory() + "\", ");
            adminListString += ("\"menu\": \"\" + admin getMenu() + "\", ");
            adminListString += ("\"count\": " + admin.getCount() + ", ");
            adminListString += ("\"price\": " + admin.getPrice() + ", ");
            adminListCount++;
        }
        if(adminListCount == (admins.size())) {
            adminListString += ("\"date\": \"\" + admin.getDate() + \"}\"");
        } else {
            adminListString += ("\"date\": \"\" + admin.getDate() + \"}, ");
        }
    }
    adminListString += "]";
    BufferedWriter writer= new BufferedWriter(new OutputStreamWriter(new FileOutputStream("C:/KioskAdminList.txt")));
    writer.write(adminListString);
    writer.flush();
}

```

CODE – DAO

AdminDao.java - 2

— — —

◆ resetCount()

- ◆ 마지막 주문 회원 일자와 현재 주문 일자가 틀릴 경우 실행되는 주문번호 초기화 Method

◆ updateMember()

- ◆ 기존 회원일 경우 Data Update 실행 Method

```
public void resetCount(int _userId) throws Exception {
    PreparedStatement stmt = null;
    System.out.println("resetCount 호출");
    try {
        stmt = this.connection.prepareStatement(
            "UPDATE KioskMember SET count = 0 WHERE id != " + _userId
        );
        stmt.executeUpdate();
    } catch (Exception e) {
        throw e;
    } finally {
        try {if (stmt != null) stmt.close();} catch (Exception e) {}
    }
}

public void updateMember(String _userPoint, String _userCount, String _user
    PreparedStatement stmt = null;
    System.out.println("updateMember 호출");
    try {
        stmt = this.connection.prepareStatement(
            "UPDATE KioskMember SET point = " + _userPoint + ", count = "
        );
        stmt.executeUpdate();
    } catch (Exception e) {
        throw e;
    } finally {
        try {if (stmt != null) stmt.close();} catch (Exception e) {}
    }
}
```

CODE _ DAO

AdminDao.java - 3

- ◆ insertMember()
- ◆ 신규 회원일 경우 Record Insert Method

- ◆ insertAdmin()
- ◆ 주문 목록을 JSON 형태로 받아와 다중 Inset 실행

```

public void insertMember(String _userNumber, String _userPoint, String _userCount, String _date) throws Exception {
    PreparedStatement stmt = null;
    System.out.println("insertMember 호출");
    try {
        stmt = this.connection.prepareStatement(
            "INSERT INTO KioskMember (number, point, count, date) VALUES ('" + _userNumber +
            "','" + _userPoint + "','" + _userCount + "','" + _date + "')");
        stmt.executeUpdate();
    } catch (Exception e) {
        throw e;
    } finally {
        try {if (stmt != null) stmt.close();} catch (Exception e) {}
    }
}

public void insertAdmin(LinkedList<String[]> _adminList) throws Exception {
    PreparedStatement stmt = null;
    System.out.println("insertAdmin 호출");
    String sql = "INSERT INTO KioskAdmin (category, menu, count, price, date) VALUES (?, ?, ?, ?, ?)";
    try {
        stmt = this.connection.prepareStatement(sql);
        for(int i=0; i<_adminList.size(); i++) {
            stmt.setString(1, _adminList.get(i)[0]);
            stmt.setString(2, _adminList.get(i)[1]);
            stmt.setString(3, _adminList.get(i)[2]);
            stmt.setString(4, _adminList.get(i)[3]);
            stmt.setString(5, _adminList.get(i)[4]);
            stmt.addBatch();
        }
        stmt.executeBatch();
    } catch (Exception e) {
        throw e;
    } finally {
        if (stmt != null) stmt.close();} catch (Exception e) {}
}

```

CODE _ Servlet

KioskMainServlet.java - 1

- ◆ doGet()
 - ◆ 회원 정보 Table JSON 파일로 저장
 - ◆ 매출 정보 Table JSON 파일로 저장
 - ◆ 마지막 주문 번호 저장
 - ◆ KioskPage.jsp 파일 실행

```
@WebServlet("/")
public class KioskMainServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        try {
            response.setContentType("text/html; charset=UTF-8");
            ServletContext sc = this.getServletContext();
            Connection conn = (Connection) sc.getAttribute("conn");
            // DAO에서 MemberList DB 값 불러오기 및 JSON 저장
            MamberDao memberDao = new MamberDao();
            memberDao.setConnection(conn);
            request.setAttribute("members", memberDao.selectMemberList());
            // 마지막 주문 번호 저장
            request.setAttribute("memberCount", memberDao.maxMemberList());
            // 관리자 화면 매출 리스트 불러오기 및 JSON 저장
            AdminDao adminDao = new AdminDao();
            adminDao.setConnection(conn);
            request.setAttribute("admins", adminDao.selectAdminList());
            RequestDispatcher rd = request.getRequestDispatcher("/jsp/Kios:
            rd.include(request, response);
        } catch(Exception e) {
            throw new ServletException(e);
        }
}
```

CODE _ Servlet

KioskMainServlet.java - 2

◆ doPost()

- ◆ 구매한 회원 정보 및 주문 목록 JSON 형태의 문자열을
2차원 배열로 변환 작업
- ◆ 주문 목록 Table에 추가 진행
- ◆ 신규 & 기존 회원 구분하여 Method 실행
- ◆ 홈 화면으로 이동

```

void doPost(HttpServletRequest request, HttpServletResponse response) throws
Exception {
    response.setContentType("text/html; charset=UTF-8");
    ServletContext sc = this.getServletContext();
    Connection conn = (Connection) sc.getAttribute("conn");
    String userId = request.getParameter("id");
    String userNumber = request.getParameter("number");
    String userPoint = request.getParameter("point");
    String userCount = request.getParameter("count");
    String userDate = request.getParameter("date");
    String keyCount = request.getParameter("keyCount");
    String userKey = request.getParameter("key");
    AdminDao adminDao = new AdminDao();
    adminDao.setConnection(conn);
    if(keyCount.equals("2")) {adminDao.resetCount(Integer.parseInt(userId));}
    String adminData = request.getParameter("adminDataList");
    System.out.println(adminData);
    LinkedList<String[]> adminList = new LinkedList<>();
    String[] adminListSample = adminData.split("\\|\\|\\|\\|");
    System.out.println(Arrays.toString(adminListSample));
    for(int i=0; i<adminListSample.length; i++) {
        adminList.add(adminListSample[i].split(","));
    }
    adminDao.insertAdmin(adminList);
    if(userKey.equals("0")) {
        adminDao.insertMember(userNumber, userPoint, userCount, userDate);
    } else {
        adminDao.updateMember(userPoint, userCount, userDate, Integer.parseInt(
    }
    response.sendRedirect("kiosk");
    catch (Exception e) {

```

CODE _ Servlet

AppInitServlet.java

— — —

- ◆ init()
 - ◆ SSH에 연결 실행
 - ◆ SSH의 외부 DataBase에 연결

- ◆ destroy()
 - ◆ 웹 종료 시 연결 종료

```

@suppressWarnings("serial")
public class AppInitServlet extends HttpServlet {
    private Session session;
    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        try {
            ServletContext sc = this.getServletContext();
            JSch jsch = new JSch();
            session = jsch.getSession(sc.getInitParameter("ssh_id"), sc.getInitParameter("ssh_ip"));
            session.setPassword(sc.getInitParameter("ssh_pw"));
            java.util.Properties config_ssh = new java.util.Properties();
            config_ssh.put("StrictHostKeyChecking", "no");
            this.session.setConfig(config_ssh);
            session.connect();
            System.out.println("SSH Connection...");
            int forward_port = session.setPortForwardingL(0, "127.0.0.1", 3306);
            System.out.println(forward_port + " --> 3306");
            Class.forName(sc.getInitParameter("driver"));
            Connection conn = DriverManager.getConnection(
                "jdbc:mysql://localhost:" + forward_port + "/cl6st10",
                sc.getInitParameter("username"),
                sc.getInitParameter("password"));
            System.out.println("DB Connection...");
            sc.setAttribute("conn", conn);
        } catch(Throwable e) {
            throw new ServletException(e);
        }
    }
    public void destroy() { }
}

```

04

REVIEW

REVIEW

자바 교육과정에서 Servlet과 MVC 구조에 대해 학습을
하게 되어 기준에 팀별 작업으로 제작한 KIOSK Project를
Node.JS으로 구성된 Back-end 부분을 JAVA로 구현을
해보게 되었습니다.

JAVA로 변환하는 과정에서 코드의 길이가 Node.JS에 비해 상당히
길어지게 되면서 MVC의 역할 분담의 필요성을 느끼게 되었습니다.
이번 기회로 추후에 만날 MVC 구조에 대해 보다 쉽게 접근할 수 있
을 것 같아 좋은 경험이었습니다.

감사합니다

박경호

JAVA기반 스마트 웹&앱 개발자 양성