스마트콘텐츠와 웹 융합 응용SW개발자 양성과정

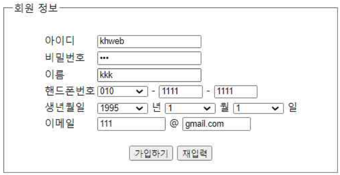
강사 - Innova Lee(이상훈)

학생- jonghyeon Yoo(유종현)

회원가입 기능

이메일 주소란에 직접입력하는 기능을 만들고 싶은데 못했습니다...

회원 가입



회원 정보

아이디 khweb

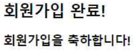
비밀번호 •••

이름 kkk

핸드폰번호 010 - 1111 - 1111

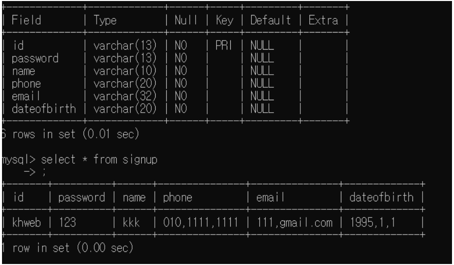
생년월일 1995 년 1 월 1 일

이메일 111 @ gmail.com



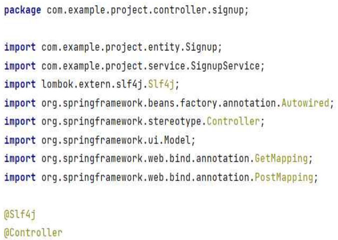
회원가입 완료!

회원가입을 축하합니다!

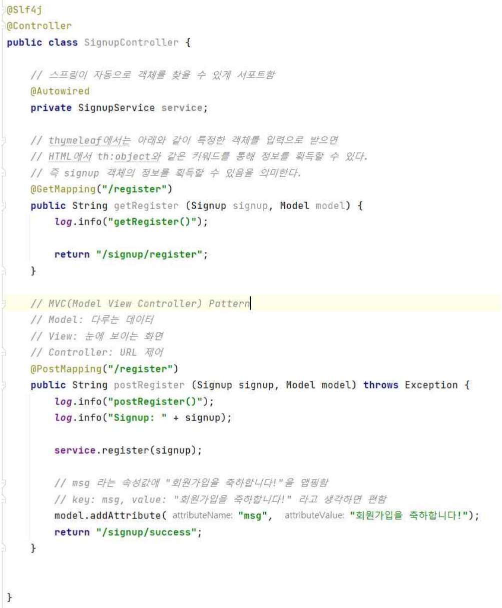


```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| id           | varchar(13) | NO   | PRI | NULL    |       |
| password     | varchar(13) | NO   |     | NULL    |       |
| name         | varchar(10) | NO   |     | NULL    |       |
| phone        | varchar(20) | NO   |     | NULL    |       |
| email        | varchar(32) | NO   |     | NULL    |       |
| dateofbirth  | varchar(20) | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql> select * from signup
    -> ;
+-------+----------+------+---------------+---------------+-------------+
| id    | password | name | phone         | email         | dateofbirth |
+-------+----------+------+---------------+---------------+-------------+
| khweb | 123      | kkk  | 010,1111,1111 | 111,gmail.com | 1995,1,1    |
+-------+----------+------+---------------+---------------+-------------+
1 row in set (0.00 sec)
```

SignupController

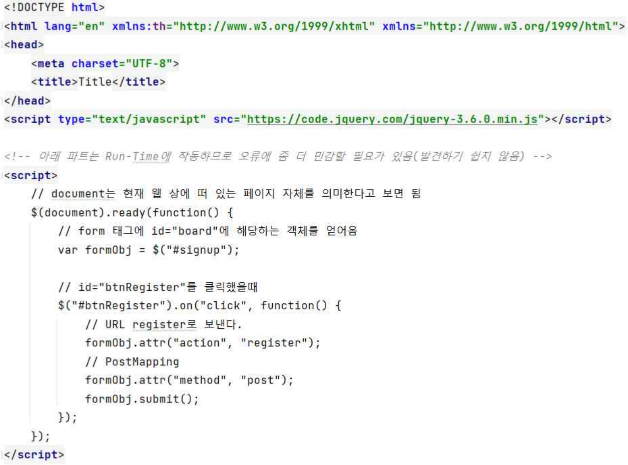


```java
package com.example.project.controller.signup;

import com.example.project.entity.Signup;
import com.example.project.service.SignupService;
import lombok.extern.slf4j.Slf4j;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

@Slf4j
@Controller
```

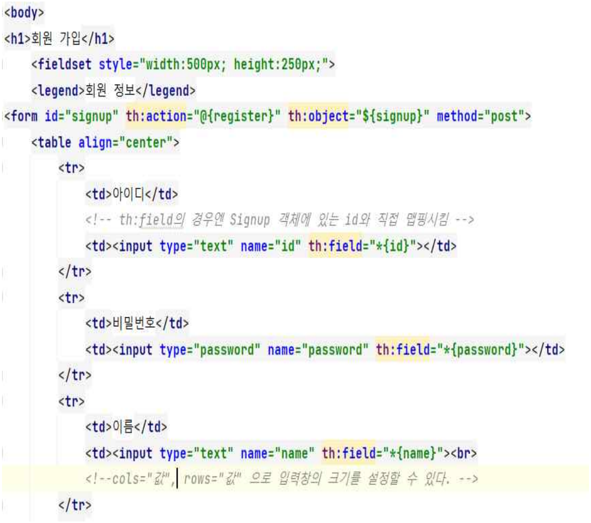


```java
@Slf4j
@Controller
public class SignupController {

    // 스프링이 자동으로 객체를 찾을 수 있게 서포트함
    @Autowired
    private SignupService service;

    // thymeleaf에서는 아래와 같이 특정한 객체를 입력으로 받으면
    // HTML에서 th:object와 같은 키워드를 통해 정보를 획득할 수 있다.
    // 즉 signup 객체의 정보를 획득할 수 있음을 의미한다.
    @GetMapping("/register")
    public String getRegister (Signup signup, Model model) {
        log.info("getRegister()");

        return "/signup/register";
    }

    // MVC(Model View Controller) Pattern
    // Model: 다루는 데이터
    // View: 눈에 보이는 화면
    // Controller: URL 제어
    @PostMapping("/register")
    public String postRegister (Signup signup, Model model) throws Exception {
        log.info("postRegister()");
        log.info("Signup: " + signup);

        service.register(signup);

        // msg 라는 속성값에 "회원가입을 축하합니다!"을 맵핑함
        // key: msg, value: "회원가입을 축하합니다!" 라고 생각하면 편함
        model.addAttribute( attributeName: "msg",  attributeValue: "회원가입을 축하합니다!");
        return "/signup/success";
    }


}
```

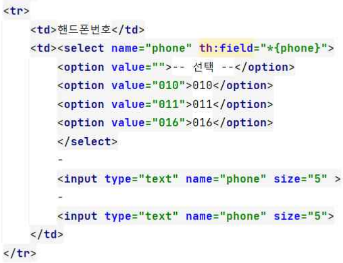
register.html



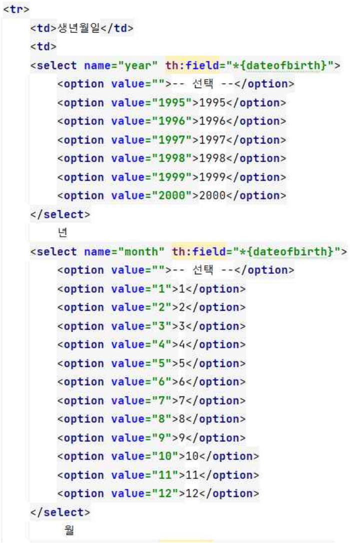
```html
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.w3.org/1999/xhtml" xmlns="http://www.w3.org/1999/html">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<script type="text/javascript" src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<!-- 아래 파트는 Run-Time에 작동하므로 오류에 좀 더 민감할 필요가 있음(발견하기 쉽지 않음) -->
<script>
    // document는 현재 웹 상에 떠 있는 페이지 자체를 의미한다고 보면 됨
    $(document).ready(function() {
        // form 태그에 id="board"에 해당하는 객체를 얻어옴
        var formObj = $("#signup");

        // id="btnRegister"를 클릭했을때
        $("#btnRegister").on("click", function() {
            // URL register로 보낸다.
            formObj.attr("action", "register");
            // PostMapping
            formObj.attr("method", "post");
            formObj.submit();
        });
    });
</script>
```



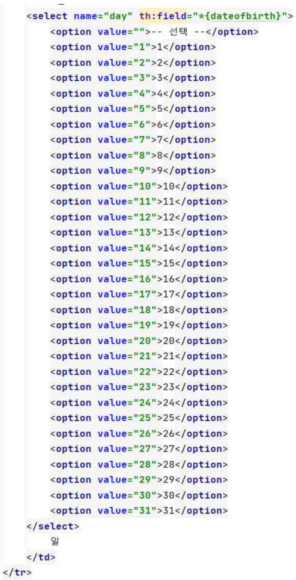
```html
<body>
<h1>회원 가입</h1>
    <fieldset style="width:500px; height:250px;">
    <legend>회원 정보</legend>
<form id="signup" th:action="@{register}" th:object="${signup}" method="post">
    <table align="center">
        <tr>
            <td>아이디</td>
            <!-- th:field의 경우엔 Signup 객체에 있는 id와 직접 맵핑시킴 -->
            <td><input type="text" name="id" th:field="*{id}"></td>
        </tr>
        <tr>
            <td>비밀번호</td>
            <td><input type="password" name="password" th:field="*{password}"></td>
        </tr>
        <tr>
            <td>이름</td>
            <td><input type="text" name="name" th:field="*{name}"><br>
            <!--cols="값", rows="값" 으로 입력창의 크기를 설정할 수 있다. -->
        </tr>
```



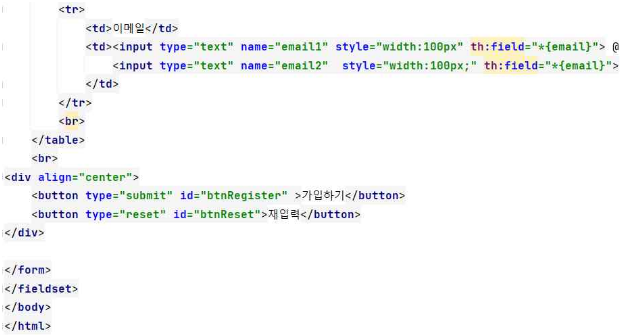
```html
<tr>
    <td>핸드폰번호</td>
    <td><select name="phone" th:field="*{phone}">
        <option value="">-- 선택 --</option>
        <option value="010">010</option>
        <option value="011">011</option>
        <option value="016">016</option>
        </select>
        -
        <input type="text" name="phone" size="5" >
        -
        <input type="text" name="phone" size="5">
    </td>
</tr>
```



```
<tr>
    <td>생년월일</td>
    <td>
    <select name="year" th:field="*{dateofbirth}">
        <option value="">-- 선택 --</option>
        <option value="1995">1995</option>
        <option value="1996">1996</option>
        <option value="1997">1997</option>
        <option value="1998">1998</option>
        <option value="1999">1999</option>
        <option value="2000">2000</option>
    </select>
        년
    <select name="month" th:field="*{dateofbirth}">
        <option value="">-- 선택 --</option>
        <option value="1">1</option>
        <option value="2">2</option>
        <option value="3">3</option>
        <option value="4">4</option>
        <option value="5">5</option>
        <option value="6">6</option>
        <option value="7">7</option>
        <option value="8">8</option>
        <option value="9">9</option>
        <option value="10">10</option>
        <option value="11">11</option>
        <option value="12">12</option>
    </select>
        월
```

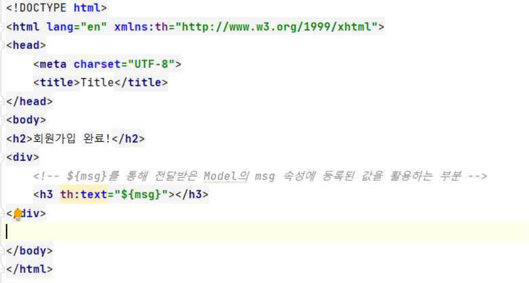


```
    <select name="day" th:field="*{dateofbirth}">
        <option value="">-- 선택 --</option>
        <option value="1">1</option>
        <option value="2">2</option>
        <option value="3">3</option>
        <option value="4">4</option>
        <option value="5">5</option>
        <option value="6">6</option>
        <option value="7">7</option>
        <option value="8">8</option>
        <option value="9">9</option>
        <option value="10">10</option>
        <option value="11">11</option>
        <option value="12">12</option>
        <option value="13">13</option>
        <option value="14">14</option>
        <option value="15">15</option>
        <option value="16">16</option>
        <option value="17">17</option>
        <option value="18">18</option>
        <option value="19">19</option>
        <option value="20">20</option>
        <option value="21">21</option>
        <option value="22">22</option>
        <option value="23">23</option>
        <option value="24">24</option>
        <option value="25">25</option>
        <option value="26">26</option>
        <option value="27">27</option>
        <option value="28">28</option>
        <option value="29">29</option>
        <option value="30">30</option>
        <option value="31">31</option>
    </select>
        일
    </td>
</tr>
```



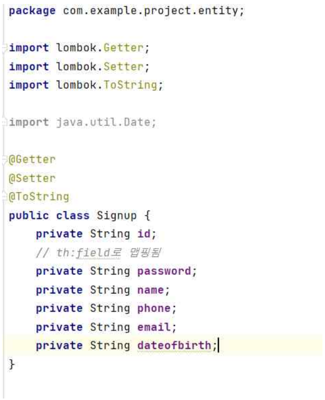
```
        <tr>
            <td>이메일</td>
            <td><input type="text" name="email1" style="width:100px" th:field="*{email}"> @
                <input type="text" name="email2"  style="width:100px;" th:field="*{email}">
            </td>
        </tr>
        <br>
    </table>
    <br>
<div align="center">
    <button type="submit" id="btnRegister" >가입하기</button>
    <button type="reset" id="btnReset">재입력</button>
</div>

</form>
</fieldset>
</body>
</html>
```

success.html



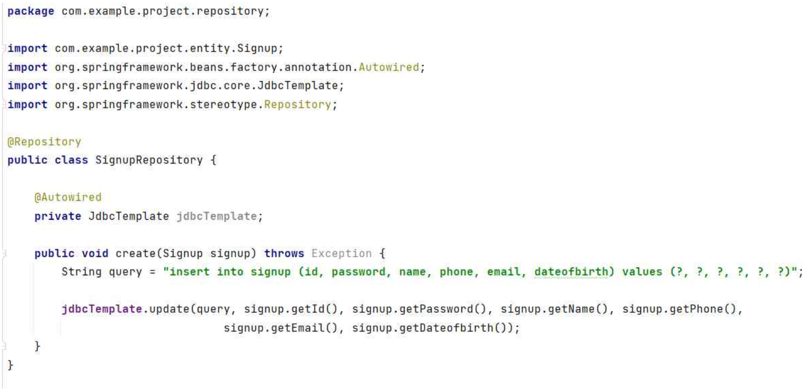
```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<h2>회원가입 완료!</h2>
<div>
    <!-- ${msg}를 통해 전달받은 Model의 msg 속성에 등록된 값을 활용하는 부분 -->
    <h3 th:text="${msg}"></h3>
</div>

</body>
</html>
```

entitiy.Signup



```java
package com.example.project.entity;

import lombok.Getter;
import lombok.Setter;
import lombok.ToString;

import java.util.Date;

@Getter
@Setter
@ToString
public class Signup {
    private String id;
    // th:field로 맵핑됨
    private String password;
    private String name;
    private String phone;
    private String email;
    private String dateofbirth;
}
```

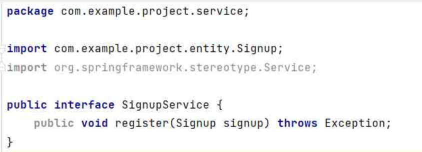
repository.SignupRepository



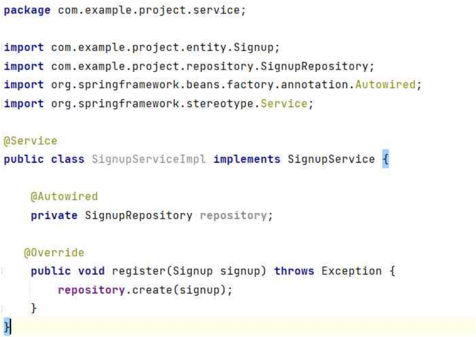
```java
package com.example.project.repository;

import com.example.project.entity.Signup;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

@Repository
public class SignupRepository {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    public void create(Signup signup) throws Exception {
        String query = "insert into signup (id, password, name, phone, email, dateofbirth) values (?, ?, ?, ?, ?, ?)";

        jdbcTemplate.update(query, signup.getId(), signup.getPassword(), signup.getName(), signup.getPhone(),
                                signup.getEmail(), signup.getDateofbirth());
    }
}
```

service

SignupServiceImpl



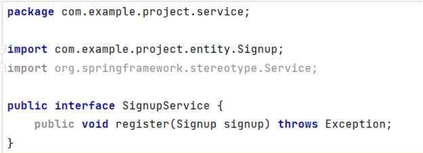
```java
package com.example.project.service;

import com.example.project.entity.Signup;
import org.springframework.stereotype.Service;

public interface SignupService {
    public void register(Signup signup) throws Exception;
}
```



```java
package com.example.project.service;

import com.example.project.entity.Signup;
import com.example.project.repository.SignupRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class SignupServiceImpl implements SignupService {

    @Autowired
    private SignupRepository repository;

   @Override
    public void register(Signup signup) throws Exception {
        repository.create(signup);
    }
}
```

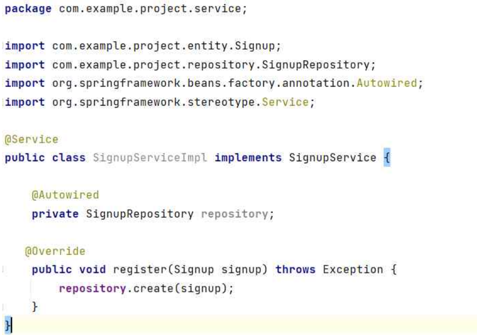
service

- SignupService



```java
package com.example.project.service;

import com.example.project.entity.Signup;
import org.springframework.stereotype.Service;

public interface SignupService {
    public void register(Signup signup) throws Exception;
}
```

- SignupServiceImpl



```java
package com.example.project.service;

import com.example.project.entity.Signup;
import com.example.project.repository.SignupRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class SignupServiceImpl implements SignupService {

    @Autowired
    private SignupRepository repository;

   @Override
    public void register(Signup signup) throws Exception {
        repository.create(signup);
    }
}
```