

■ 게시판

● 필요 기술 / 라이브러리

- HTML Design (bootstrap)
- Database (H2)
- JPA
 - Model (@Entity)
 - ModelRepository (JpaRepository 상속)
- Controller
- Thymeleaf

■ 게시판 – 작성

● templates/board/write.html

```
...  
<div class="jumbotron">  
  <div class="container text-center">  
    <form method="post" action="/board/write">  
      <div class="form-group">  
        <label for="title">Title:</label>  
        <input type="text" class="form-control" name="title" id="title">  
      </div>  
      <div class="form-group">  
        <label for="content">Content:</label>  
        <textarea class="form-control" rows="5" name="content" id="content">  
        </textarea>  
      </div>  
      <button type="button" id="complete" class="btn btn-primary btn-block">  
        글쓰기  
      </button>  
    </form>  
  </div>  
</div>  
...
```

```
<script>  
  $("#complete").click(function() {  
    $("form").submit();  
    return false;  
  });  
</script>
```

Title:

Content:

글쓰기

■ 게시판 – 작성

● model/Board.java

```
package com.example.board.model;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

import lombok.Data;

@Entity
@Data
public class Board {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long id;
    private String title;
    private String content;
    private String userId;
}
```

■ 게시판 – 작성

● repository/BoardRepository.java

```
package com.example.board.repository;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
import com.example.board.model.Board;  
  
public interface BoardRepository extends JpaRepository<Board, Long>  
{  
}
```

■ 게시판 – 작성

● controller/BoardController.java

```
@Controller
public class BoardController {
    @Autowired
    BoardRepository boardRepository;

    @Autowired
    HttpSession session;

    @GetMapping("/board/write")
    public String boardWrite() {
        return "board/write";
    }

    @PostMapping("/board/write")
    public String boardWritePost(@ModelAttribute Board board) {
        User user = (User) session.getAttribute("user_info");
        String userId = user.getEmail();
        board.setUserId(userId);
        boardRepository.save(board);

        return "board/write";
    }
}
```

■ 게시판 – 작성

● <http://localhost:8080/board/write>

The screenshot shows a web browser at localhost:8080/board/write. The page has a navigation bar with 'Portfolio', 'Home', 'Board', 'Map', and 'Contact'. A 'Sign out ggoreb님 반갑습니다.' link is on the right. The main form has a 'Title:' label and a text input field containing '제목~~'. Below it is a 'Content:' label and a larger text area containing '내용~~'. A blue '글쓰기' (Write) button is at the bottom. A 'Footer Text' is visible at the very bottom.



This is an identical screenshot of the web form shown on the left, representing the state after a successful database insert.

The screenshot shows a database client window with the title '주의 요함 | 192.168.0.2:8082/login.do?jsessi...'. The left sidebar shows a tree view of the database schema. The 'BOARD' table is selected, and its columns are listed: ID, CONTENT, TITLE, and USER_ID. The 'SQL 문:' (SQL Statement) field contains the query 'SELECT * FROM BOARD'. The '실행' (Execute) button is highlighted. Below the query, the results are displayed in a table with 1 row and 4 columns: ID, CONTENT, TITLE, and USER_ID. The values are 2, 내용~~, 제목~~, and seorab@naver.com respectively. The execution time is shown as (1 row, 13 ms).

ID	CONTENT	TITLE	USER_ID
2	내용~~	제목~~	seorab@naver.com

■ 게시판 – 작성

● ID가 2 이상 부터 입력되는 현상 수정

– 테이블 마다 자동 증가 기능을 따로 처리하도록 수정

● application.properties

```
# datasource
spring.datasource.url=jdbc:h2:tcp://localhost/~ /board
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=

# jpa
spring.jpa.hibernate.ddl-auto=update
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.show-sql=true

# auto increment
spring.jpa.hibernate.use-new-id-generator-mappings=false
```

시퀀스

- HIBERNATE_SEQUENCE
- SYSTEM_SEQUENCE_507BFD88_125
- SYSTEM_SEQUENCE_894ACA0_7C
- SYSTEM_SEQUENCE_DBDA987A_47

■ 게시판 - 조회(목록)

● templates/board/list.html

```
...
<div class="jumbotron">
  <div class="container text-center">
    <table class="table table-hover">
      <thead>
        <tr>
          <th>번호</th><th>제목</th><th>작성자</th>
        </tr>
      </thead>
      <tbody>
        <tr th:each="board : ${list}" th:attr="id=${board.id}">
          <td th:text="${board.id}"></td>
          <td th:text="${board.title}"></td>
          <td th:text="${board.userId}"></td>
        </tr>
      </tbody>
    </table>
    <button type="button" class="btn btn-primary btn-block" id="write">글쓰기</button>
  </div>
</div>

...

<script>
  $("#write").click(function() {
    location = "/board/write";
    return false;
  });
</script>
```


■ 게시판 - 조회(목록)

● controller/BoardController.java

```
@Controller
public class BoardController {

    ...

    @GetMapping("/board")
    public String board(Model model) {
        List<Board> list = boardRepository.findAll();
        model.addAttribute("list", list);
        return "board/list";
    }
}
```

localhost:8080/board

Portfolio Home Board Map Contact Sign in

번호	제목	작성자
2	제목~~	seorab@naver.com

글쓰기

Footer Text

■ 게시판 - 조회(목록)

● 목록 정렬

번호	제목	작성자
2	제목~~	seorab@naver.com
3	제목2	seorab@naver.com
4	제목3	seorab@naver.com
5	제목4	seorab@naver.com

글쓰기

Footer Text



번호	제목	작성자
5	제목4	seorab@naver.com
4	제목3	seorab@naver.com
3	제목2	seorab@naver.com
2	제목~~	seorab@naver.com

글쓰기

Footer Text

● controller/BoardController.java

```
@GetMapping("/board")
public String board(Model model) {
    List<Board> list =
        boardRepository.findAll(Sort.by(Sort.Direction.DESC, "id"));
    model.addAttribute("list", list);
    return "board/list";
}
```

■ 게시판 - 조회(상세내용)

● templates/board/list.html

```
...  
<script>  
    $("#write").click(function() {  
        location = "/board/write";  
        return false;  
    });  
    $("tr").click(function() {  
        var id = $(this).attr("id");  
        location = "/board/" + id;  
        return false;  
    });  
</script>
```

- 테이블의 행 클릭 시 tr의 id 속성을 이용하여 url 변경

■ 게시판 - 조회(상세내용)

● templates/board/view.html

```
<!DOCTYPE html>
<html lang="en">
<head th:replace="common/head">
</head>
<body>

    <nav class="navbar navbar-inverse" th:replace="common/nav">
    </nav>

    <div class="container">
        <div class="panel-group">
            <div class="panel panel-default">
                <div class="panel-heading" th:text="${board.id} + '번' "></div>
                <div class="panel-heading" th:text="${board.title}"></div>
                <div class="panel-body" th:text="${board.content}"></div>
                <div class="panel-footer" th:text="${board.userId}"></div>
            </div>
        </div>
    </div>
    <div class="container text-right">
        <button type="button" class="btn btn-default" id="list">목록</button>
        <button type="button" class="btn btn-default" id="update">수정</button>
        <button type="button" class="btn btn-default" id="delete">삭제</button>
    </div>

    <footer class="container-fluid text-center" th:replace="common/footer">
    </footer>
```

■ 게시판 - 조회(상세내용)

● templates/board/view.html

```
<script>
    $("#list").click(function() {
        location = "/board";
        return false;
    });
    $("#update").click(function() {
        location = "/board/update/" + [[${board.id}]];
        return false;
    });
    $("#delete").click(function() {
        location = "/board/delete/" + [[${board.id}]];
        return false;
    });
</script>
</body>
</html>
```

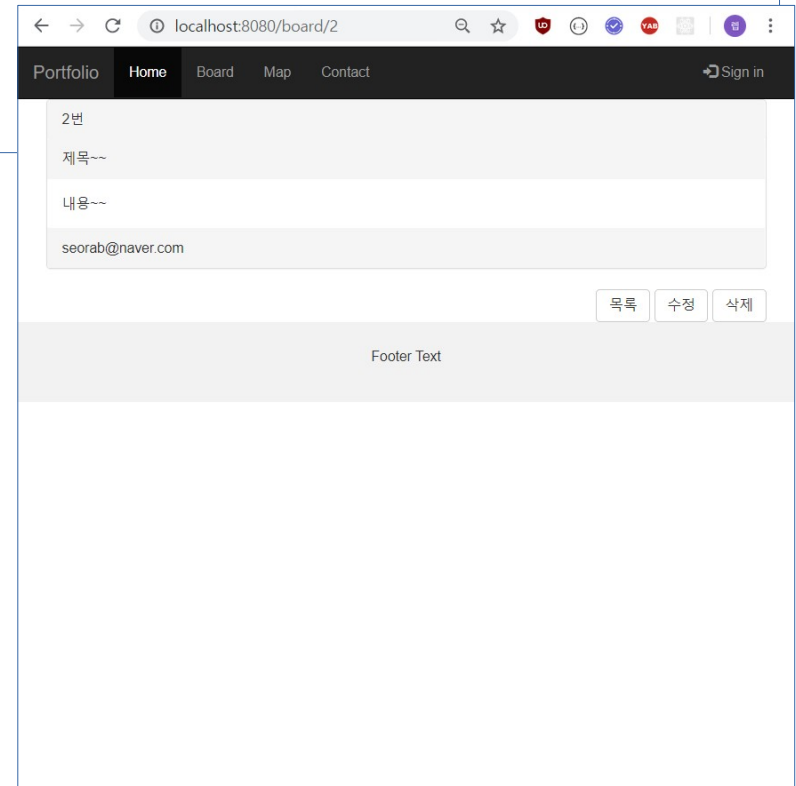
■ 게시판 - 조회(상세내용)

● controller/BoardController.java

```
@Controller
public class BoardController {

    ...

    @GetMapping("/board/{id}")
    public String boardView(Model model, @PathVariable("id") long id) {
        Optional<Board> data = boardRepository.findById(id);
        Board board = data.get();
        model.addAttribute("board", board);
        return "board/view";
    }
}
```



■ 게시판 - 수정

● templates/board/update.html

```
...
<div class="jumbotron">
  <div class="container text-center">
    <form method="post" th:action="@{/board/update/} + ${board.id}">
      <div class="form-group">
        <label for="title">Title:</label>
        <input type="text" class="form-control" name="title"
          id="title" th:value="${board.title}">
      </div>
      <div class="form-group">
        <label for="content">Content:</label>
        <textarea class="form-control" rows="5" name="content"
          id="content" th:value="${board.content}"></textarea>
      </div>
      <button type="button" id="complete" class="btn btn-primary btn-block">
        수정
      </button>
    </form>
  </div>
</div>

...

<script>
  $("#complete").click(function() {
    $("form").submit();
    return false;
  });
</script>
```

■ 게시판 - 수정

● controller/BoardController.java

```
@Controller
public class BoardController {

    ...

    @GetMapping("/board/update/{id}")
    public String boardUpdate(Model model, @PathVariable("id") long id) {
        Optional<Board> data = boardRepository.findById(id);
        Board board = data.get();
        model.addAttribute("board", board);
        return "board/update";
    }

    @PostMapping("/board/update/{id}")
    public String boardUpdatePost(
        @ModelAttribute Board board, @PathVariable("id") long id) {
        User user = (User) session.getAttribute("user_info");
        String userId = user.getEmail();
        board.setUserId(userId);
        board.setId(id);
        boardRepository.save(board);
        return "redirect:/board/" + id;
    }
}
```


■ 게시판 - 수정

● 상세내용 → 수정 → 완료

Portfolio Home Board Map Contact Sign in

2번

제목~~

내용~~

seorab@naver.com

목록 수정 삭제

Footer Text



Portfolio Home Board Map Contact Sign out ggoreb님 반갑습니다.

Title:

제목~~ 수정

Content:

내용~~

수정

Footer Text



Portfolio Home Board Map Contact Sign out ggoreb님 반갑습니다.

2번

제목~~ 수정

내용~~

seorab@naver.com

목록 수정 삭제

Footer Text

■ 게시판 - 수정

● controller/BoardController.java

```
@Controller
public class BoardController {

    ...

    @GetMapping("/board/delete/{id}")
    public String boardDelete(@PathVariable("id") long id) {
        boardRepository.deleteById(id);
        return "redirect:/board";
    }
}
```

■ 지도

● 필요 기술 / 라이브러리

- HTML Design (bootstrap)
- Database (H2)
- JPA
 - Model (@Entity)
 - ModelRepository (JpaRepository 상속)
- Kakao API (기본 지도, 여러개 마커 표시)
- Controller
- AJAX (주변 장소 데이터 가져오기)
- 위도/경도 이용 반경 계산 공식

■ 지도 (기본)

● templates/map/map.html (appkey – 38d34c29c097c5f5c8c02416cb4aefff)

```
...
<div class="jumbotron">
  <div class="container text-center">
    <div id="map" style="width:500px; height:400px; margin:0 auto;"></div>
  </div>
</div>
...

<script src="http://dapi.kakao.com/v2/maps/sdk.js?appkey=[appkey]"></script>
<script>
  var container = document.getElementById('map');
  var options = {
    center : new kakao.maps.LatLng(37.4980239, 127.027572), // 강남역
    level : 4
  };

  var map = new kakao.maps.Map(container, options);

  var markerPosition = new kakao.maps.LatLng(37.4980239, 127.027572);

  var marker = new kakao.maps.Marker({
    position: markerPosition
  });

  marker.setMap(map);
</script>
```

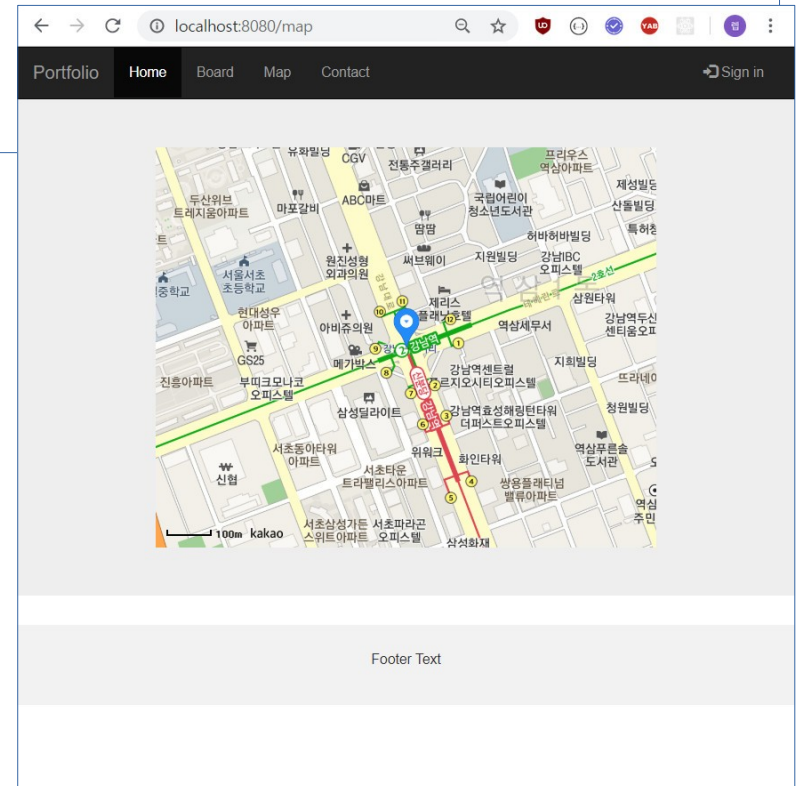
■ 지도 (기본)

● controller/MapController.java

```
package com.example.board.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class MapController {
    @GetMapping("/map")
    public String map(Model model) {
        return "map/map";
    }
}
```



■ 지도 (여러개 마커 표시)

● model/Point.java

```
package com.example.board.model;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import lombok.Data;

@Entity
@Data
public class Point {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long id;
    private String title;
    private double latitude;
    private double longitude;
}
```

● repository/PointRepository.java

```
package com.example.board.repository;

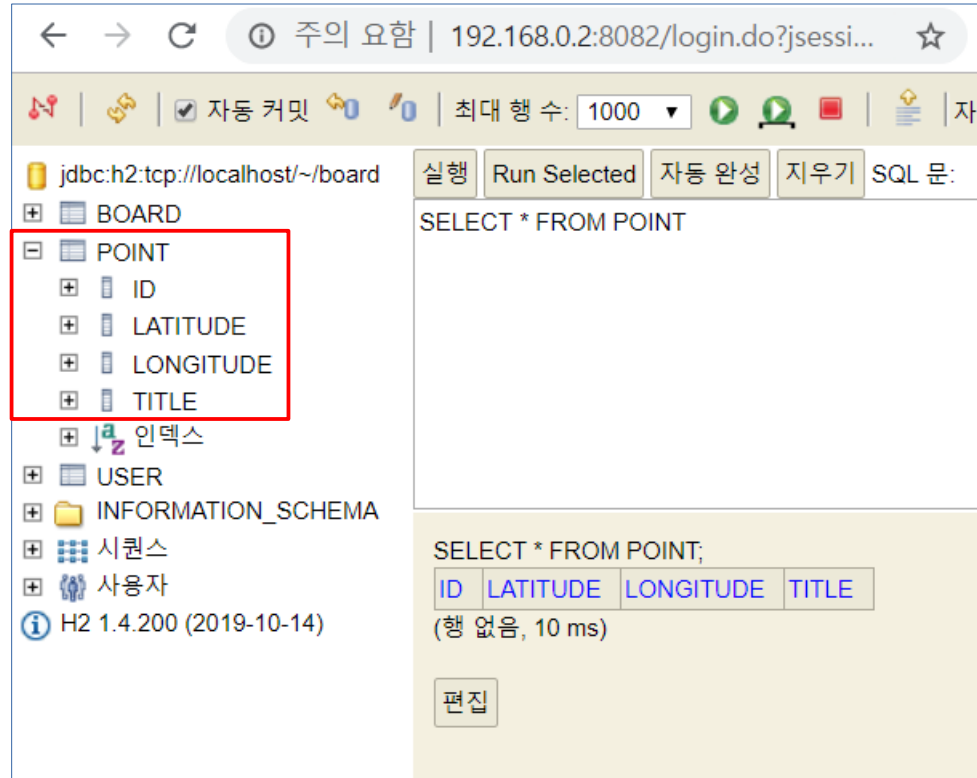
import org.springframework.data.jpa.repository.JpaRepository;
import com.example.board.model.Point;

public interface PointRepository extends JpaRepository<Point, Long>{

}
```

■ 지도 (여러개 마커 표시)

- 프로젝트 실행 → H2 Database 내에 테이블 생성 확인



■ 지도 (여러개 마커 표시)

● POINT 테이블에 데이터 직접 입력

실행

Run Selected

자동 완성

지우기

SQL 문:

```
insert into point (id, title, latitude, longitude) values (1, '코엑스', 37.5121917, 127.0566319);
insert into point (id, title, latitude, longitude) values (2, '63빌딩', 37.5192502, 126.9400105);
insert into point (id, title, latitude, longitude) values (3, '서울역', 37.5544681, 126.9686187);
insert into point (id, title, latitude, longitude) values (4, '김포공항', 37.5578738, 126.8004231);
insert into point (id, title, latitude, longitude) values (5, '정동진해변', 37.6908032, 129.032434);
```

```
insert into point (id, title, latitude, longitude) values (1, '코엑스', 37.5121917, 127.0566319);
갱신된 개수: 1
(15 ms)

insert into point (id, title, latitude, longitude) values (2, '63빌딩', 37.5192502, 126.9400105);
갱신된 개수: 1
(2 ms)

insert into point (id, title, latitude, longitude) values (3, '서울역', 37.5544681, 126.9686187);
갱신된 개수: 1
(8 ms)

insert into point (id, title, latitude, longitude) values (4, '김포공항', 37.5578738, 126.8004231);
갱신된 개수: 1
(1 ms)

insert into point (id, title, latitude, longitude) values (5, '정동진해변', 37.6908032, 129.032434);
갱신된 개수: 1
(1 ms)
```

```
insert into point (id, title, latitude, longitude)
values (1, '코엑스', 37.5121917, 127.0566319);
```

```
insert into point (id, title, latitude, longitude)
values (2, '63빌딩', 37.5192502, 126.9400105);
```

```
insert into point (id, title, latitude, longitude)
values (3, '서울역', 37.5544681, 126.9686187);
```

```
insert into point (id, title, latitude, longitude)
values (4, '김포공항', 37.5578738, 126.8004231);
```

```
insert into point (id, title, latitude, longitude)
values (5, '정동진해변', 37.6908032, 129.032434);
```


■ 지도 (여러개 마커 표시)

● controller/MapController.java

```
...  
  
@Controller  
public class MapController {  
    @Autowired  
    PointRepository pointRepository;  
  
    ...  
  
    @GetMapping("/map/getPoint")  
    @ResponseBody  
    public List<Point> getPoint() {  
        List<Point> list = pointRepository.findAll();  
        return list;  
    }  
}
```

← → ↻ ⓘ localhost:8080/map/getPoint

```
[  
  {  
    "id": 1,  
    "title": "코엑스",  
    "latitude": 37.5121917,  
    "longitude": 127.0566319  
  },  
  {  
    "id": 2,  
    "title": "63빌딩",  
    "latitude": 37.5192502,  
    "longitude": 126.9400105  
  },  
  {  
    "id": 3,  
    "title": "서울역",  
    "latitude": 37.5544681,  
    "longitude": 126.9686187  
  },  
  {  
    "id": 4,  
    "title": "김포공항",  
    "latitude": 37.5578738,  
    "longitude": 126.8004231  
  },  
  {  
    "id": 5,  
    "title": "정동진해변",  
    "latitude": 37.6908032,  
    "longitude": 129.032434  
  }  
]
```

■ 지도 (여러개 마커 표시)

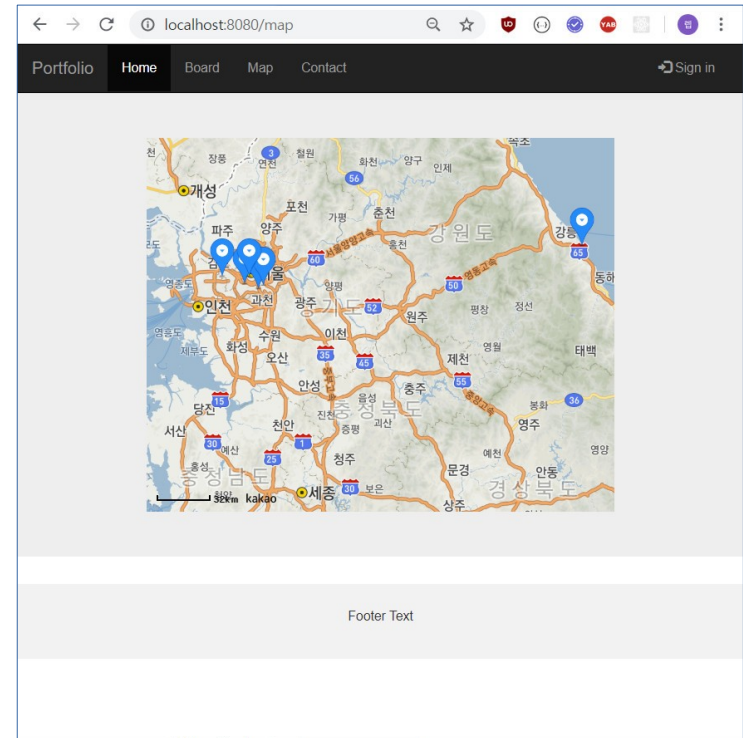
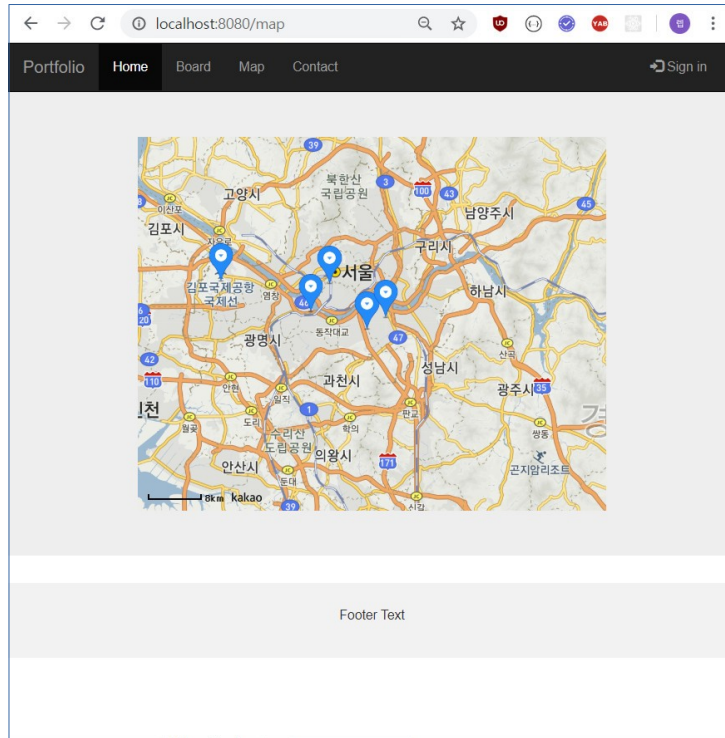
● templates/map/map.html (appkey – 38d34c29c097c5f5c8c02416cb4aefff)

```
...
<div class="jumbotron">
  <div class="container text-center">
    <div id="map" style="width:500px; height:400px; margin:0 auto;"></div>
  </div>
</div>
...

<script src="http://dapi.kakao.com/v2/maps/sdk.js?appkey=[appkey]"></script>
<script>
  ...
  $.ajax({
    url: '/map/getPoint',
    data: {},
    success: function(res) {
      for(var i = 0; i < res.length; i++) {
        var marker = new kakao.maps.Marker({
          map : map, // 마커를 표시할 지도
          position : new kakao.maps.LatLng(res[i].latitude, res[i].longitude),
          title : res[i].title,
        });
      }
    }
  });
</script>
```

■ 지도 (여러개 마커 표시)

● <http://localhost:8080/map>



■ 지도 (반경 거리 기준 표시)

● controller/MapController.java

...

@Controller

public class MapController {

@Autowired

PointRepository pointRepository;

...

@GetMapping("/map/getPoint")

@ResponseBody

public List<Point> getPoint(**double** lat, **double** lng) {

List<Point> list = pointRepository.findAll();

for(**int** i = list.size() - 1; i >= 0; i--) {

double dist =

distance(lat, lng, list.get(i).getLatitude(), list.get(i).getLongitude());

if(dist > 10) {
list.remove(i);
}

}

return list;

}

■ 지도 (반경 거리 기준 표시)

● controller/MapController.java

```
private double distance(double lat1, double lon1, double lat2, double lon2) {  
    double theta = lon1 - lon2;  
    double dist = Math.sin(deg2rad(lat1)) * Math.sin(deg2rad(lat2))  
        + Math.cos(deg2rad(lat1)) * Math.cos(deg2rad(lat2))  
        * Math.cos(deg2rad(theta));  
  
    dist = Math.acos(dist);  
    dist = rad2deg(dist);  
    dist = dist * 60 * 1.1515;  
  
    dist = dist * 1.609344;  
  
    return dist;  
}  
  
private double deg2rad(double deg) {  
    return deg * Math.PI / 180.0;  
}  
  
private double rad2deg(double rad) {  
    return rad * 180 / Math.PI;  
}  
}
```

■ 지도 (반경 거리 기준 표시)

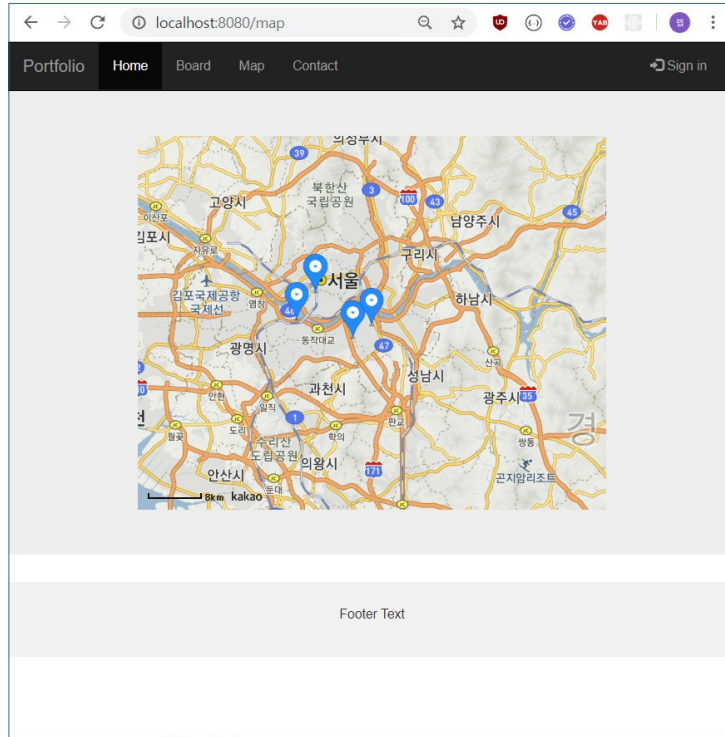
● templates/map/map.html (appkey – 38d34c29c097c5f5c8c02416cb4aefff)

```
...
<div class="jumbotron">
  <div class="container text-center">
    <div id="map" style="width:500px; height:400px; margin:0 auto;"></div>
  </div>
</div>
...

<script src="http://dapi.kakao.com/v2/maps/sdk.js?appkey=[appkey]"></script>
<script>
  $.ajax({
    url: '/map/getPoint',
    data: {"lat": 37.4980239, "lng": 127.027572},
    success: function(res) {
      for(var i = 0; i < res.length; i++) {
        var marker = new kakao.maps.Marker({
          map : map,
          position : new kakao.maps.LatLng(res[i].latitude, res[i].longitude),
          title : res[i].title,
        });
      }
    }
  });
</script>
```

■ 지도 (반경 거리 기준 표시)

● <http://localhost:8080/map>



■ 이메일 발송

● 필요 기술 / 라이브러리

- HTML Design (bootstrap)
- spring-boot-mail 라이브러리
- Google SMTP (구글 계정, 메일앱 비밀번호)
- Controller
- AJAX (메일 발송 요청)

■ 이메일 발송

● build.gradle

```
plugins {  
    id 'org.springframework.boot' version '2.2.2.RELEASE'  
    id 'io.spring.dependency-management' version '1.0.8.RELEASE'  
    id 'java'  
}  
  
group = 'com.ggoreb'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '1.8'  
  
...  
  
Dependencies {  
  
    ...  
  
    implementation group: 'org.springframework.boot',  
        name: 'spring-boot-starter-mail', version: '2.2.2.RELEASE'  
}  
  
test {  
    useJUnitPlatform()  
}
```

■ 이메일 발송

● controller/ContactController.java

```
package com.example.board.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.ResponseBody;

import com.example.board.email.Mailer;
import com.example.board.email.SMTPAuthenticator;

@Controller
public class ContactController {

    @GetMapping("/contact")
    public String contact() {
        return "contact/contact";
    }

    @PostMapping("/contact")
    @ResponseBody
    public String contactPost(String email, String comment) {
        Mailer mailer = new Mailer();
        mailer.sendMail(
            "관리자 메일 주소", "문의내용 : " + email,
            comment, new SMTPAuthenticator());
        return "ok";
    }
}
```

■ 이메일 발송

● email/SMTPAuthenticator.java

```
package com.example.board.email;

import javax.mail.Authenticator;
import javax.mail.PasswordAuthentication;

public class SMTPAuthenticator extends Authenticator {
    @Override
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(
            "seorabkim@gmail.com", "hxrdsaewjuhhti");
    }
}
```

■ 이메일 발송

● email/Mailer.java

```
package com.example.board.email;

import java.util.Properties;

import javax.mail.Address;
import javax.mail.Message;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class Mailer {
    public void sendMail(
        String to, String subject, String content, SMTPAuthenticator smtp) {
        Properties p = new Properties();
        p.put("mail.smtp.host", "smtp.gmail.com");
        p.put("mail.smtp.port", "465");
        p.put("mail.smtp.starttls.enable", "true");
        p.put("mail.smtp.auth", "true");
        p.put("mail.smtp.debug", "true");
        p.put("mail.smtp.socketFactory.port", "465");
        p.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        p.put("mail.smtp.socketFactory.fallback", "false");
```

■ 이메일 발송

● email/Mailer.java

```
try {  
    Session ses = Session.getInstance(p, smtp);  
    ses.setDebug(true);  
    MimeMessage msg = new MimeMessage(ses); // 메일의 내용을 담을 객체  
    msg.setSubject(subject); // 제목  
    Address fromAddr = new InternetAddress("seorabkim@gmail.com");  
    msg.setFrom(fromAddr); // 보내는 사람  
    Address toAddr = new InternetAddress(to);  
    msg.addRecipient(Message.RecipientType.TO, toAddr); // 받는 사람  
    msg.setContent(content, "text/html;charset=UTF-8"); // 내용과 인코딩  
    Transport.send(msg); // 전송  
} catch (Exception e) {  
    e.printStackTrace();  
}  
  
}
```

■ 이메일 발송

● templates/contact/contact.html

```
...
<div class="jumbotron">
  <div class="container text-center">
    <div class="form-group">
      <label for="email">Email:</label>
      <input type="text" class="form-control" id="email">
    </div>
    <div class="form-group">
      <label for="comment">Comment:</label>
      <textarea class="form-control" rows="5" id="comment"></textarea>
    </div>
    <button type="button" id="send" class="btn btn-primary btn-block">
      발송
    </button>
  </div>
</div>
...

<script>
  $('#send').click(function() {
    $.ajax({
      url: '/contact',
      type: 'post',
      data: { 'email': $('#email').val(), 'comment': $('#comment').val() },
      success: function(res) {
        alert('메일 발송 완료');
      }
    })
  });
</script>
```

■ 프로젝트 추가 적용사항

- 각 기능 흐름 연결 (버튼 등 UI 사용)
- 모든 입력 항목 유효성 검사
 - 회원가입, 로그인, 게시글 작성/수정, 이메일 발송
- 로그인이 필요한 메뉴는 인터셉터 등을 이용하여 접속 차단
- 작성자 본인만 게시글 수정/작성 가능
- 지도에 표시되는 지점에 대해 추가/수정/삭제 등 관리
- 디자인 수정/보완