

Spring Data JPA

프로젝트 : B21bSpringDataJPA1

준비사항

- 의존설정 : JDBC API, Lombok, Oracle Driver, Spring Web, Spring Data JPA
- JSP 사용을 위한 설정을 한다.
- Refresh Gradle Project 를 눌러 적용한다.

프로퍼티 : application.properties

```
1 spring.application.name=B21bSpringDataJPA1
2
3 #포트설정
4 server.port=8586
5
6 #JSP 설정
7 spring.mvc.view.prefix=/WEB-INF/views/
8 spring.mvc.view.suffix=.jsp
9
10 #오라클 접속 정보
11 spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
12 spring.datasource.url=jdbc:oracle:thin:@localhost:1521:xe
13 spring.datasource.username=musthave
14 spring.datasource.password=1234
15 spring.datasource.hikari.maximum-pool-size=5
16
17 #JPA설정 : persistence.xml과 동일한 내용임
18 spring.jpa.database-platform=org.hibernate.dialect.OracleDialect
19 spring.jpa.open-in-view=false
20 spring.jpa.properties.hibernate.show_sql=true
21 spring.jpa.properties.hibernate.format_sql=true
22 # none, create, create-drop, update, validate
23 spring.jpa.properties.hibernate.hbm2ddl.auto=none
24
```

엔티티 : Member.java

```
1 package edu.spring.boot.jpa;
2
3*import java.time.LocalDate;
14
15 @Data
16 @AllArgsConstructor
17 @NoArgsConstructor(access = AccessLevel.PROTECTED)
18 @Builder
19 @Entity(name = "JPAMEMBER01")
20 public class Member {
21     @Id
22     @GeneratedValue
23     private Long id;
24     private String username;
25     @Column(name = "create_date")
26     private LocalDate createDate;
27 }
```

레포지토리 : edu/spring/boot/jpa/MemberRepository.java

```
1 package edu.spring.boot.jpa;
2
3*import org.springframework.data.jpa.repository.JpaRepository;
5
6 @Repository
7 public interface MemberRepository extends JpaRepository<Member, Long>{
8     //제네릭 타입매개변수 : long이 아니라 Long으로 작성
9     //기본적인 Create, Read, Update, Delete 자동으로 생성
10 }
11
```

서비스 : /B21bSpringDataJPA1/src/main/java/edu/spring/boot/jpa/MemberService.java

```
1 package edu.spring.boot.jpa;
2
3*import java.util.List;
8
9 @Service
10 public class MemberService {
11
12     @Autowired
13     private MemberRepository dao;
14
15     public Member insert(Member member) {
16         Member resultMember = dao.save(member);
17         return resultMember;
18     }
19 }
```

```

20 public Optional<Member> select(Long id){
21     Optional<Member> member = dao.findById(id);
22     return member;
23 }
24
25 public List<Member> selectAll(){
26     return dao.findAll();
27 }
28
29 public void delete(Long id) {
30     dao.deleteById(id);
31 }
32
33 public Member update(Member member) {
34     Member resultMember = dao.save(member);
35     return resultMember;
36 }
37 }

```

컨트롤러 : /B21bSpringDataJPA1/src/main/java/edu/spring/boot/MainController.java

```

1 package edu.spring.boot;
2
3 import java.time.LocalDate;
15
16 @Controller
17 public class MainController {
18
19     @GetMapping("/")
20     public String main() {
21         return "main";
22     }
23
24     @Autowired
25     MemberService memberService;
26
27     //입력
28     @GetMapping("/insert.do")
29     public String insert(@RequestParam("username") String name,
30                         Model model) {
31         Member member = Member.builder()
32             .username(name)
33             .createDate(LocalDate.now())
34             .build();
35         Member result = memberService.insert(member);
36         model.addAttribute("member", result);
37         return "insert";
38     }

```

```

39 //조회
40 @GetMapping("/select.do")
41 public String select(@RequestParam("id") Long p_id,
42                     Model model) {
43     Optional<Member> result = memberService.select(p_id);
44     if (result.isPresent()) {
45         model.addAttribute("member", result.get());
46     }
47     else {
48         model.addAttribute("member", null);
49     }
50     return "select";
51 }
52 //전체조회
53 @GetMapping("/selectAll.do")
54 public String selectAll(Model model) {
55     List<Member> result = memberService.selectAll();
56     model.addAttribute("members", result);
57     return "selectAll";
58 }
59 //삭제
60 @GetMapping("/delete.do")
61 public String delete(@RequestParam("id") Long pid) {
62     memberService.delete(pid);
63     return "delete";
64 }
65 //수정
66 @GetMapping("/update.do")
67 public String update(Member member, Model model) {
68     member.setCreateDate(LocalDate.now());
69     Member result = memberService.update(member);
70     model.addAttribute("member", result);
71     return "update";
72 }
73 }

```

스프링 부트 3.2 이슈

스프링 부트는 3.2부터 더 이상 바이트코드를 파싱할때 파라미터 이름을 추론하지 않는다고 한다.

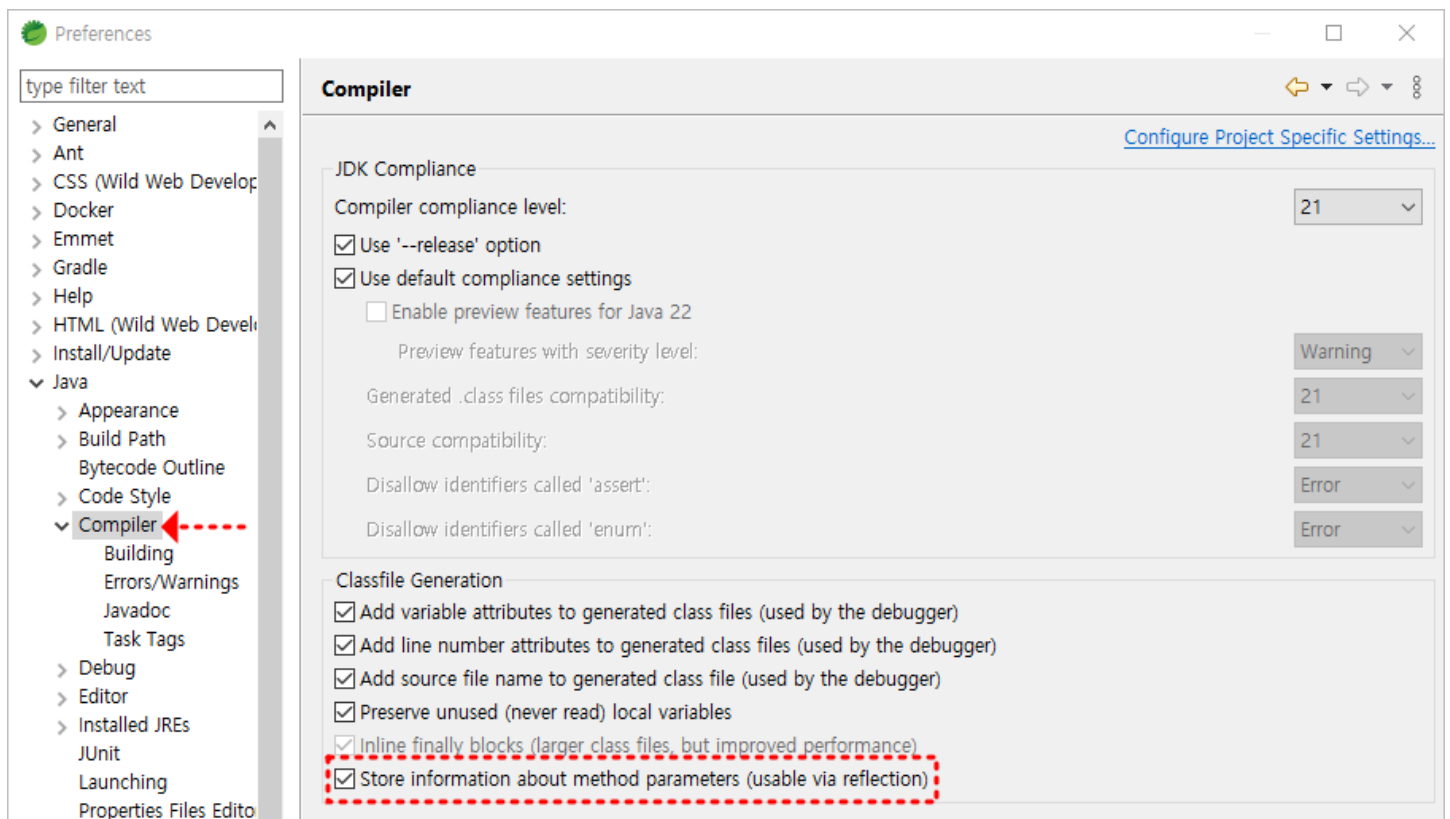
그리고 의존성 주입 또는 속성을 바인딩 할 때는 -parameters 옵션으로 컴파일 했는지 꼭 확인하라고 한다.

이로 인해 발생하는 오류는 다음과 같다.

Cannot resolve parameter names for constructor

해결 방법은 이클립스에서는 다음과 같이 설정하면 된다.

상단 메뉴 ⇒ Window ⇒ Preferences ⇒ Java ⇒ Compiler ⇒ Store information about method parameters (usable via reflection) ⇒ 체크



뷰 : /B21bSpringDataJPA1/src/main/webapp/WEB-INF/views/main.jsp

```
9<body>
10    <h2>Spring Data JPA</h2>
11    <ul>
12        <!-- 최초 실행시 test1부터 5까지 추가 후 테스트.
13             재시작시 초기화되므로 none으로 변경. -->
14        <li><a href=/insert.do?username=test1>데이터 추가</a></li>
15        <li><a href=/select.do?id=1>개별 조회</a></li>
16        <li><a href=/selectAll.do>전체 조회</a></li>
17        <li><a href=/delete.do?id=2>데이터 삭제</a></li>
18        <li><a href=/update.do?id=1&username=홍길동>데이터 수정</a></li>
19    </ul>
20 </body>
21 </html>
```

뷰 : /B21bSpringDataJPA1/src/main/webapp/WEB-INF/views/insert.jsp

```
10<body>
11    <h2>Spring boot 프로젝트</h2>
12    <ul>
13        <li><a href="/">루트</a></li>
14    </ul>
15
16    <h2>Spring Data JPA - insert</h2>
17    <p>
18        아이디 : ${member.id}<br>
19        이름 : ${member.username}<br>
20        날짜 : ${member.createDate}
21    </p>
22 </body>
23 </html>
```

뷰 : /B21bSpringDataJPA1/src/main/webapp/WEB-INF/views/select.jsp

```
10<body>
11    <h2>Spring boot 프로젝트</h2>
12    <ul>
13        <li><a href="/">루트</a></li>
14    </ul>
15
16    <h2>Spring Data JPA - select</h2>
17    <p>
18        아이디 : ${member.id}<br>
19        이름 : ${member.username}<br>
20        날짜 : ${member.createDate}
21    </p>
22 </body>
23 </html>
```

뷰 : /B21bSpringDataJPA1/src/main/webapp/WEB-INF/views/selectAll.jsp

```
10 <body>
11     <h2>Spring boot 프로젝트</h2>
12     <ul>
13         <li><a href="/">루트</a></li>
14     </ul>
15
16     <h2>Spring Data JPA - 전체 select</h2>
17     <c:forEach items="${members}" var="row">
18     <p>
19         아이디 : ${row.id}<br>
20         이름 : ${row.username}<br>
21         날짜 : ${row.createDate}
22     </p>
23     </c:forEach>
24 </body>
25 </html>
```

뷰 : /B21bSpringDataJPA1/src/main/webapp/WEB-INF/views/delete.jsp

```
10 <body>
11     <h2>Spring boot 프로젝트</h2>
12     <ul>
13         <li><a href="/">루트</a></li>
14     </ul>
15
16     <h2>Spring Data JPA - delete</h2>
17     <p>삭제되었습니다.</p>
18 </body>
19 </html>
```

뷰 : /B21bSpringDataJPA1/src/main/webapp/WEB-INF/views/update.jsp

```
10 <body>
11     <h2>Spring boot 프로젝트</h2>
12     <ul>
13         <li><a href="/">루트</a></li>
14     </ul>
15
16     <h2>Spring Data JPA - update</h2>
17     <p>
18         아이디 : ${member.id}<br>
19         이름 : ${member.username}<br>
20         날짜 : ${member.createDate}
21     </p>
22 </body>
23 </html>
```