

(디지털 컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

훈련기간 : 2021.05.07 ~ 2021.12.08

JPA 설정

application.yaml

```
dialect: org.hibernate.dialect.MySQL8Dialect

jpa:
  open-in-view: false
  database: mysql
  show-sql: true
  generate-ddl: true
  hibernate:
    ddl-auto: update

logging:
  level:
    org:
      hibernate:
        SQL: DEBUG
```

build.gradle

```
dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
```

혹시 root계정에서 생성된거면 접근이 거부되기때문에 권한부여 해주고 접근제한을 풀어준다.

JPA 설정


```
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.ToString;
import javax.persistence.*;

@Data
@NoArgsConstructor
@Entity
@ToString
public class DaumNews {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long newsSeq;

    @Column(length = 20, nullable = false)
    private String newsNo;
    @Column(length = 20, nullable = false)
    private String category;

    @Column(length = 500, nullable = false)
    private String title;

    @Column(length = 2000, nullable = false)
    private String address;
    @Builder
    public DaumNews(String newsNo, String category, String title, String address) {
        this.newsNo = newsNo;
        this.address = address;
        this.category = category;
        this.title = title;
    }
}
```

 DaumNewsRepository

```
package com.example.demo.repository;
```

```
import com.example.demo.entity.DaumNews;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface DaumNewsRepository extends JpaRepository<DaumNews, Long> {
    public DaumNews findByNewsNo(String newsNo);
}
```

```
mysql> desc daum_news;
```

Field	Type	Null	Key	Default	Extra
news_seq	bigint	NO	PRI	NULL	auto_increment
address	varchar(2000)	NO		NULL	
category	varchar(20)	NO		NULL	
news_no	varchar(20)	NO		NULL	
title	varchar(500)	NO		NULL	

```
5 rows in set (0.01 sec)
```

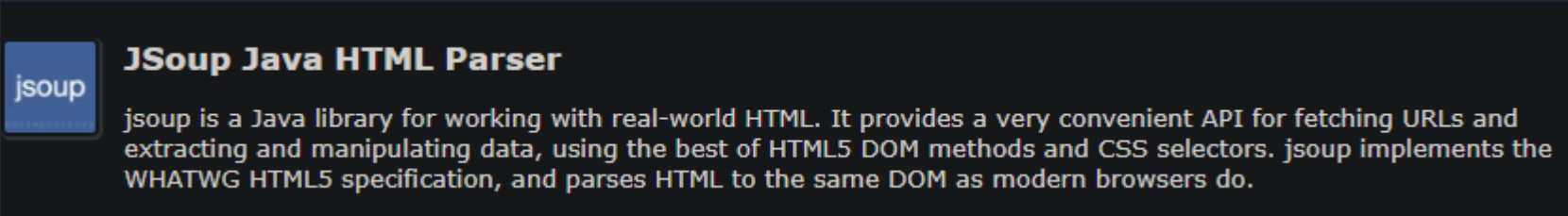
```
mysql> select * from daum_news;
```

news_seq	address	category	news_no	title
217	https://news.v.daum.net/v/20210728211519716	society	1	김호규 금속
218	https://news.v.daum.net/v/20210728184228051	society	2	수요일 어김
219	https://news.v.daum.net/v/20210728200139139	society	3	"언론에 재깁
220	https://news.v.daum.net/v/20210728195908063	society	4	소방청-소방
221	https://news.v.daum.net/v/20210728192421206	society	5	비리로 해임
222	https://news.v.daum.net/v/20210728164200995	society	6	'故' 박원순
223	https://news.v.daum.net/v/20210728210007420	society	7	박원순 유족
224	https://news.v.daum.net/v/20210728181000087	society	8	내달부터 50
225	https://news.v.daum.net/v/20210728203840936	society	9	정부 "모더
226	https://news.v.daum.net/v/20210728152255597	society	10	김호규 금속
227	https://v.daum.net/v/20210728161922099	society	11	투마니아 축
228	https://v.daum.net/v/20210728200355185	society	12	"샤넬 싸게
229	https://v.daum.net/v/20210728172351358	society	13	제주 중학생
230	https://v.daum.net/v/20210728173819861	society	14	"장난으로 모
231	https://v.daum.net/v/20210728175600564	society	15	'윤석열 아
232	https://v.daum.net/v/20210728165740564	society	16	고양서 화이
233	https://v.daum.net/v/20210728165207375	society	17	갑자기 추락
234	https://v.daum.net/v/20210728180835026	society	18	"합격 축하

ENTITY의 구조로 JPA가 자동으로 DB를 구성한다.

JSOUP

Maven Repository에서 JSOUP을 찾아서 build 추가한다.



라이브러리 :

HTML5 DOM 메서드와 CSS 선택기를 통해 URL을 가져오고 데이터를 추출한다

출력 값

```
public class lianxue {  
    public static void main(String[] args) throws IOException {  
        System.out.println(Jsoup.connect( url: "https://weather.naver.com/").get());  
    }  
}
```

IOException을 추가해주는 것을 보면 입출력관련 처리를 한다는 작업임을 알수있다

IOException을 추가해주는 것을 보면 입출력관련 처리를 한다는 작업임을 알수있다.

```
Document document = Jsoup.connect( url: "https://weather.naver.com/").get();
```

```
org.jsoup.nodes
public class Document
extends org.jsoup.nodes.Element
```

A HTML Document.

Gradle: org.jsoup:jsoup:1.13.1

```
org.jsoup.Connection
public abstract org.jsoup.nodes.Document get()
throws java.io.IOException
```

Execute the request as a GET, and parse the result.

Returns: parsed Document

Throws: `java.net.MalformedURLException` – if the request URL is not a HTTP or HTTPS URL, or is otherwise malformed

`org.jsoup.HttpStatusException` – if the response is not OK and HTTP response errors are not ignored

`org.jsoup.UnsupportedMimeTypeException` – if the response mime type is not supported and those errors are not ignored

`java.net.SocketTimeoutException` – if the connection times out

`java.io.IOException` – on error

Gradle: org.jsoup:jsoup:1.13.1

Document: jsoup으로 크롤링해온 결과 HTML 문서

Element: Document의 HTML 문서

Elements: Element 들이 모인 자료형

```
<div class="notify_area">
<div class="notify_inner"> <a href="https://blog.naver.com/naver_weather/222408929154" class="link_notify on" onclick="nclk_v2(this, 'gnb.notibar
<storing class="badge">
    ◆◆◆◆◆
</storing> <span class="text">◆◆◆◆◆ ◆0◆ ◆◆◆◆</span> </a> <a href="https://search.naver.com/search.naver?sm=mtp_hty.top&amp;where=m&amp
<storing class="badge">
    U◆◆
</storing> <span class="text">◆◆◆◆◆ ◆◆◆◆◆素</span> </a>
</div>
</div>
</div>
<div id="container" role="main" class="is_today korean">
<div id="content">
<div class="section_center">
<div class="card card_today">
<h2 class="blind">◆◆◆◆◆ ◆◆◆◆◆</h2>
<div class="today_weather">
<div class="cp_area"> <span class="blind">◆◆◆◆◆ ◆◆◆◆◆◆ ◆◆◆◆◆</span>
<ul class="list_cp">
<li class="item kma on"> <button type="button" class="button" data-cp-name="KMA" onclick="nclk_v2(this, 'wtk.cpskmi', '', '')" aria-selected=
<li class="item accuweather "> <button type="button" class="button" data-cp-name="ACCUWEATHER" onclick="nclk_v2(this, 'wtk.cpsacc', '', '')"
<li class="item twc "> <button type="button" class="button" data-cp-name="TWC" onclick="nclk_v2(this, 'wtk.cpstw", '', '')" aria-selected=
</ul>
<div class="btn_area"> <button type="button" class="btn_setting" onclick="nclk_v2(this, 'wtk.cpsmore', '', '')"><span class="text">◆◆◆◆◆◆◆
```

위와같이 `jsoup.connect`를 통해 해당 페이지의 모든 HTML 코드를 가져온다.

JSOUP 메서드가 다양해서 알맞게 사용하면 될것같다.

```
public class lianxue {  
    public static void main(String[] args) throws IOException {  
        Document doc = (Document) Jsoup.connect( url: "https://search.naver.com/search.naver?w  
        Elements el = doc.getElementsByAttributeValue("class", "recite _recite result");  
        System.out.println(el.get(0).text());  
    }  
}
```

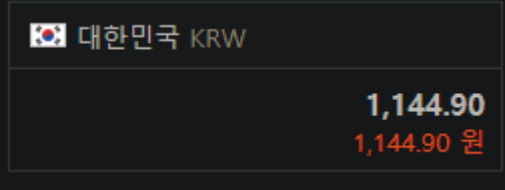
1,144.90 원

```
public class lianxue {  
    public static void main(String[] args) throws IOException {  
        //      System.out.println(Jsoup.connect("https://weather.naver.com/").get());  
        Document document = Jsoup.connect( url: "https://weather.naver.com/").get();  
  
        Elements elements = document.getElementsByAttributeValue("class", "current");  
  
        System.out.println(elements.get(0));  
    }  
}
```

<strong class="current">현재 온도32°

System.out.println(elements.get(0).text());

현재 온도32°



1,144.90

CssQuery에대해서 헛갈리는부분이 어느부모 클래스를 기준을 잡고 자식을 타고 들어가야하는지 몰랐는데 그냥 구분이 되도록 스스로 원하는 기준을 잡고 들어가면 다 출력이 된다. 최상단부터 잡고 들어가도 되고 최하단을 기준으로 부모 몇개만 적고 들어가도 된다.

```
public class lianxue {  
    public static void main(String[] args) throws IOException {  
        Document document = Jsoup.connect( url: "https://weather.naver.com/").get();  
  
        Elements elements = document.select( cssQuery: "div.weather_area>strong.current");  
  
        String[] str = elements.text().split( regex: "");  
        List<String> st = Collections.singletonList(elements.text());  
        String s = elements.text();  
  
        System.out.println(st.get(0));  
  
        System.out.println(s);  
  
        for (int i = 0; i < str.length; i++) {  
            System.out.printf(str[i]);  
        }  
    }  
}
```

현재 온도32°

현재 온도32°

현재 온도32°

(cssQuery: ".today_weather>.weather_area>.current");

팁으로 div, strong span 이런것 적을 필요없이 .으로 앞에 태그명을 생략하고 잡을수도있다. 이름이 비슷한경우는 명시해주는게 좋을것같다. 물론 다받은다음에 배열에서 빼주는 방법을 선택 해도 된다. 이름이 같은경우는 데이터를 다 가져오기때문이다.