

회원가입 정규표현식 유효성 검사



오늘은 정규표현식을 이용하여 회원 DB 테이블과 DTO객체에 대해서 생각해보는 시간을 가집시다! 오늘도 자바로만 기능들을 구현하다 보니 회원가입에 필요한 기능이지만 미처 작성하지 못한 기능이 있습니다. 이 부분은 뒤에서도 설명드릴게요.

유효성 검사가 뭔가요? (<https://goddino.tistory.com/52>)

```
<div class="regist">
  <form action="registUser" method="post">
    <div class="input-group mb-3">
      <span class="input-group-text" id="basic-addon1">아이디</span>
      <input type="text" name="id" class="form-control" placeholder="아이디" aria-label="Username" aria-describedby="basic-addon1">
      <span class="input-group-text" id="idCheck">아이디 중복 검사</span>
    </div>

    <div class="input-group mb-3">
      <span class="input-group-text" id="basic-addon1">비밀번호</span>
      <input type="password" name="password" class="form-control" placeholder="비밀번호는 대, 소문자와 숫자, 특수문자를 포함하여 생성해주세요" aria-
    </div>

    <div class="input-group mb-3">
      <span class="input-group-text" id="basic-addon1">비밀번호 확인</span>
      <input type="password" name="checkPassword" class="form-control" placeholder="비밀번호 확인" aria-label="Username" aria-describedby
    </div>

    <div class="input-group mb-3">
      <span class="input-group-text" id="basic-addon1">이름</span>
      <input type="text" name="name" class="form-control" placeholder="이름" aria-label="Username" aria-describedby="basic-addon1">
    </div>

    <div class="input-group mb-3">
      <label class="input-group-text" for="inputGroupSelect01">성별</label>
      <select class="form-select" id="inputGroupSelect01" name="gender">
        <option selected>성별</option>
        <option value="M">남성</option>
        <option value="F">여성</option>
      </select>
    </div>
  </form>
</div>
```

```

<div class="input-group mb-3">
  <span class="input-group-text" id="basic-addon1">이메일</span>
  <input type="text" name="emailFirst" class="form-control" placeholder="이메일" aria-label="Recipient's username" aria-describedby="basic-addon1">
  <span class="input-group-text" id="basic-addon2">@</span>
  <select name="emailLast">
    <option value="naver.com">naver.com</option>
    <option value="gmail.com">gmail.com</option>
    <option value="daum.net">daum.net</option>
    <option value="hanmail.net">hanmail.net</option>
  </select>
</div>

<div class="input-group mb-3">
  <span class="input-group-text" id="basic-addon1">전화번호</span>
  <input type="number" name="phone" class="form-control" aria-label="Username" aria-describedby="basic-addon1">
</div>

<div class="input-group mb-1">
  <span class="input-group-text" id="basic-addon1">생년월일</span>
  <input type="text" name="year" placeholder="년" min="1900" max="2023" class="form-control" aria-label="Username" aria-describedby="basic-addon1">
  <select name="month">
    <option value="월">월</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>
  <input type="number" name="day" placeholder="일" min="1" max="31" class="form-control" aria-label="Username" aria-describedby="basic-addon1">
</div>
<div>
  <input type="submit" class="btn btn-success" value="회원 가입">
</div>
</form>
</div>

```

위 코드는 제가 회원가입 폼을 만들기 위해 작성한 코드입니다. 부트스트랩을 이용하여 좀 많아보이기도 하지만, 실제로도 많은게 맞습니다.(ㅎㅎ) 하지만 이게 회원가입의 전부일까요? 위 코드를 실행하면 나오는 화면을 보여드리면

아이디	아이디	아이디 중복 검사
비밀번호	비밀번호는 대,소문자와 숫자, 특수문자를 포함하여 생성해주세요	
비밀번호 확인	비밀번호 확인	
이름	이름	
성별	성별 ▼	
이메일	이메일	@ naver.com ▼
전화번호		
생년월일	년	월 ▼ 일
회원 가입		

이런 화면이 나오게됩니다. 우리가 주로 하게되는 회원가입을 생각하면 어떤 기능, 항목들이 빠져있을까요? 우선 최근에는 필수적으로 진행되는 전화번호 인증, 혹은 이메일 인증 기능이 빠져있습니다. 그리고 또 필수적인 주민등록번호 항목도 빠져있네요. 주민등록번호가 들어가게 된다면, 주민등록번호의 앞자리로 생년월일을 대체하는 기능이나, 뒷자리가 양식에 적합한지(이상의 숫자로 시작하지 않는지 등)

을 검사해야 합니다. 그리고 쇼핑몰같은 사이트에서는 필수적인 주소 입력 칸도 보이지 않네요. 주소 입력과 같은 부분은 API가 필수적이고, 그렇게 되면 자바만이 아닌 자바스크립트도 사용할 필요가 있어 역시 제외했습니다.

(아이디 중복 검사도 버튼은 있지만, 자바스크립트를 사용하지 않고 구현할 방법이 생각나지 않아 일단 미뤄졌습니다.. 컨트롤러 단에서 할 수는 있지만 보통 그렇게 사용하지 않기 때문에..)

잠깐만 살펴봐도 회원가입에 생각보다 많은 노력과 시간이 필요하다는 것을 알 수 있습니다. 위에서 생략한 기능들 말고도, 우리는 정규표현식을 이용해서

1. 비밀번호 양식 유효성 검사
2. 전화번호 양식 유효성 검사
3. 이메일 양식 유효성 검사

를 진행하겠습니다.

제가 생성한 User dto입니다. 편의상 생성자, 게터 세터 등은 생략했습니다. 필드가 많다고 느껴질 수도 있지만, 사실 여기도 빠진 부분이 생각보다 많습니다!

dto를 보시면 이상하게 느껴지는 부분이 있으실거예요. 왜 emailFirst와 emailLast가 따로있을까? 왜 생년월일이 year, month, day로 쪼개져있을까? 등등 이런 궁금증도 뒤에서 다루겠습니다!

```
public class User {  
  
    private String id; // 아이디  
    private String password; // 비밀번호  
    private String name; // 이름  
    private String emailFirst; // 이메일 - 아이디부분  
    private String emailLast; // 이메일 - 도메인부분  
    private String email; // 이메일 합친 것  
    private int age; // 나이  
    private char gender; // 성별 ('F', 'M'과 같은 형식으로 입력하기 위해 char형)  
    private String phone; // 전화번호  
    private int year; // 생년월일의 년도  
    private int month; // 생년월일의 달  
    private int day; // 생년월일의 일  
    private String grade; // 회원 등급  
    private String createTime; // 회원 생성 일자 (DB에서 default로 가입 시의 날짜를 입력)  
  
}
```

비밀번호 양식 유효성 검사

유효성 검사를 위해 RegistCheck라는 클래스를 Util패키지에 만들었습니다. RegistCheck의 전체 코드는 첨부하겠습니다.

정규표현식에 대해 다루기 전에 (<https://hbase.tistory.com/160>) 링크를 통해 간단하게 확인하고 오시면 좋을 것 같습니다!

비밀번호 유효성 검사를 위한 메소드입니다.

```
// 컨트롤러  
// 파라미터로 User user, String checkPassword를 입력받았습니다.  
// checkPassword : 비밀번호 확인 칸의 값, 객체에 따로 필드를 만들지 않았기 때문에  
// 파라미터로 해당 값만 따로 입력받아왔습니다  
  
if(!user.getPassword().equals(checkPassword)) {  
    model.addAttribute("msg", "비밀번호가 일치하지 않습니다.");  
    return "user/registForm";  
}
```

비밀번호 비밀번호는 대,소문자와 숫자, 특수문자를 포함하여 생성해주세요

비밀번호 확인 비밀번호 확인

첫 번째로 비밀번호와 비밀번호 확인 칸의 값이 같은지를 판별하는 과정입니다. 비밀번호는 Input의 password 특성 상 사용자가 작성하며 육안으로 확인할 수 없습니다. 그렇기에 회원가입 시 잘못입력한 값으로 회원가입이 진행되어버리면 곤란하겠죠? 그것을 방지하기 위한 비밀번호 확인 칸입니다.

간단하게 입력받은 User 객체의 비밀번호와, checkPassword가 일치하는지 아닌지를 판별합니다.

```
// 컨트롤러
if(!rCheck.passCheck(user.getPassword(), user.getId())){
    model.addAttribute("msg", "비밀번호는 8~16자 영문 대 소문자, 숫자, 특수문자를 사용하세요.");
    return "user/registerForm";
}
```

```
public class RegistCheck {

    public boolean passCheck(String password, String id) {

        // 비밀번호를 최소 8자 최대 16자, 영어 대소문자를 각각 포함하고, 숫자와 특수문자를 최소 1자씩 포함하여 만들었는가
        String pattern = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[@$!%*?&])[A-Za-z\\d@$!%*?&]{8,16}$";

        // 비밀번호에 아이디가 포함되어 있는가
        if(password.contains(id)) {
            return false;
        }

        // Pattern은 위의 문자열 pattern이 아니라 자바에서 유효성 검사를 지원하는 클래스입니다
        // matches : password문자열이 pattern 정규표현식에 맞는가, 맞으면 true 반환
        return Pattern.matches(pattern, password);
    }
}
```

두 번째로 회원가입 양식의 유효성 검사를 위한 RegistCheck 클래스 내의 메소드입니다. 수상하고 암호문같은 문자열이 보이실텐데, 저게 바로 비밀번호가 최소 8자이고, 최대 16자 미만이면서 대문자와 소문자를 모두 포함하고, 숫자와 특수문자까지 다 포함하는 지를 판별해주는 정규표현식입니다.

헉 세상에 저걸 다알아야하나요?!?!?

그건 아닙니다. 어차피 정규표현식은 전부 외워서 작성하기 힘들뿐더러, 자주 사용하는 정규표현식들은 검색으로 찾을 수 있습니다. 오늘 게시글에서 가장 중요한 건, **대충 암호문같이 생긴 문자열이 있는데, 저게 정규표현식이라는 거고, 문자열의 형태를 검사할 때 사용한다더라.** 라는 사실을 알아가는 것이라고 생각합니다. 지금 당장 필요한 - 회원가입 비밀번호 정규표현식, 이메일 정규표현식, 전화번호 정규표현식 등 - 은 구글 검색으로도 충분히 찾을 수 있습니다. 우선은 남이 만든 정규표현식을 사용하여 회원가입들의 여러가지 기능들을 처리해 보고, 그 후에 정규표현식에 대해서 공부하는게 더 좋다고 생각해요.

pattern 문자열은 유효성 검사를 위한 것이고, 비밀번호에 아이디가 포함되어있는지를 검사하기 위한 if문도 있습니다. 우리가 회원가입을 진행하다 보면 비밀번호에 아이디가 있으면 빠꾸먹는 경우가 있죠. 이렇게 비밀번호 유효성 검사를 진행했습니다!

이메일 유효성 검사

이메일 유효성 검사를 하기 전에, 우리가 회원가입 시 보통 이메일을 어떻게 입력하는 지 생각해봅시다. haanwhy@gmail.com 과 같이 한꺼번에 입력할 때도 있지만, naver.com / gmail.com / hanmail.net 등의 선택지가 있는 select 태그가 있는 경우도 왕왕 있습니다. 이런 경우에 어떻게 입력받고, 객체의 필드를 어떻게 만들고, DB 테이블은 또 어떻게 만들지? 하는 생각이 듭니다.

이메일 이메일 @ naver.com ▼

그래서 이번에는 이메일을 입력할 때 앞의 아이디 부분, 뒤의 도메인 부분을 나누어 입력받았습니다.

```
private String emailFirst; // 이메일 - 아이디부분
private String emailLast; // 이메일 - 도메인부분
private String email; // 이메일 합친 것
```

그런기에 User dto를 만들 때 각각의 필드를 따로 만들어주었습니다. 이대로 DB에도 따로 저장하는 방법도 있지만, 저는 DB에는 완성된 EMAIL을 저장하고 싶었기 때문에 앞 뒤를 합쳐서 넣어줄 필드도 생성했습니다.

```
// 회원가입 폼에서 이메일 아이디 부분과 도메인 부분을 나누어서 입력받았기 때문에 합쳐서 email 필드에 넣어준다
user.setEmail(user.getEmailFirst()+"@"+user.getEmailLast());
```

그래서 우선 컨트롤러에서 이메일을 완성하여 USER객체에 넣어줬습니다.

```
// 컨트롤러
// 이메일 조건에 맞지 않을 때
    if(!rCheck.emailCheck(user.getEmail())) {
        model.addAttribute("msg", "이메일이 올바르지 않습니다.");
        return "user/registForm";
    }
```

```
// RegistCheck 클래스의 메소드
public boolean emailCheck(String email) {

    // 이메일이 aaaa@naver.com 와 같은 형태로 만들어져 있는가
    // 이메일을 입력받을때 아이디 부분과 도메인 부분을 나누어서 컨트롤러에서 합쳤기 때문에 검사해줘야한다
    String pattern = "^[_a-z0-9-]+(\\.[_a-z0-9-]+)*@(?:\\w+\\.\\.)+\\w+$";

    return Pattern.matches(pattern, email);

}
```

그렇게 넣어준 이메일을 또다시 정규표현식을 통해 검사해줍니다. 올바르게 입력받아 완성했다면 true를 반환받고 정상적으로 처리됩니다.

전화번호 유효성 검사

전화번호도 입력받는 방법이 다양하지만, 최근에는 전화번호 인증을 필수적으로 하는 만큼 한번에 입력받는 경우가 많다고 생각하여 그러한 경우로 진행했습니다. 숫자여야 할 이유도 느껴지지 않아서 User객체와 테이블 모두 String, VARCHAR형으로 생성했습니다.

```
// 컨트롤러
// 전화번호를 올바르게 입력하지 않았을 때
// 최근에는 회원가입 시 전화번호로 바로 개인 인증을 하는 경우가 많기 때문에 전화번호가 틀리면 인증 부분에서 막힌다
// 우리는 아직 번호 인증을 하지 않기 때문에 전화번호 조건 검사를 넣었다
if(!rCheck.phoneCheck(user.getPhone())) {
    model.addAttribute("msg", "전화번호가 올바르지 않습니다.");
    return "user/registForm";
}
```

```
// RegistCheck 클래스 메소드
public boolean phoneCheck(String phone) {

    // 전화번호가 01012341234 와 같은 형태인지
    // 최근 회원가입들은 전화번호 인증을 하기위해 하이픈(-)을 빼고 입력받는 경우가 많다
    // 그렇기에 이렇게 검사한다
    String pattern = "^01(?:0|1|[6-9])(?:\\d{3}|\\d{4})\\d{4}$";

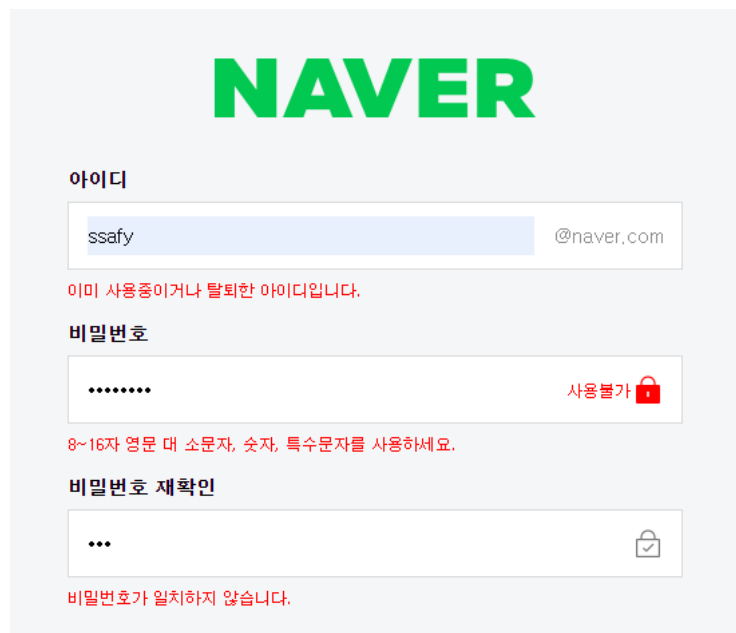
    // 하이픈(-)을 포함하는 전화번호를 검사하기 위한 정규표현식
    // "^01(?:0|1|[6-9])-(?:\\d{3}|\\d{4})-\\d{4}$"

    return Pattern.matches(pattern, phone);
}
```

}

전화번호를 인증에 사용하기 위해서는 하이픈(-)이 없어야하지만, 만약 전화번호를 앞, 중간, 뒷자리를 따로 입력받거나 저장할 때는 다른 정규표현식을 사용하면 됩니다.

이렇게 오늘은 회원가입 유효성 검사를 해봤습니다. 이렇게 자바로 열심히 유효성 검사를 하고 이런 말을 하기는 이상하지만, 사실 유효성 검사에서는 자바스크립트가 더 중요하다고 생각합니다. 하지만 수업과정에서 자바스크립트를 다루지 않은지 너무 오랜 시간이 지났고... 그리고 지금의 게시글은 유효성 검사와 정규표현식의 존재와 존재의의를 아는데 의미가 있다고 생각하여 자바를 사용하여 작성했습니다. 만약 자바스크립트를 사용하면 어떤 것을 할 수 있을까요?



The image shows a Naver login interface with the following elements:

- NAVER** logo at the top.
- 아이디** (ID) section: A text input field containing 'ssafy' and a dropdown menu showing '@naver.com'. Below the input, a red error message reads: "이미 사용중이거나 탈퇴한 아이디입니다." (This ID is already in use or has been deleted).
- 비밀번호** (Password) section: A password input field with masked characters '.....'. To the right, a red error message reads: "사용불가" (Not allowed) with a red lock icon. Below the input, a red message reads: "8~16자 영문 대 소문자, 숫자, 특수문자를 사용하세요." (Use 8~16 characters including uppercase and lowercase letters, numbers, and special characters).
- 비밀번호 재확인** (Confirm Password) section: A password input field with masked characters '...'. To the right, there is a checkmark icon. Below the input, a red error message reads: "비밀번호가 일치하지 않습니다." (Passwords do not match).

네이버 회원가입에서 볼 수 있는 장면입니다. 일부러 존재할만한 아이디를 입력했고, 비밀번호는 111111 같은걸 입력하고, 재확인 칸에도 일부러 다른 값을 입력했습니다.

그러면 이런 식으로 아이디가 중복인지 아닌지, 양식에 맞는지 아닌지를 바로 사용자가 볼 수 있게 띄워줍니다. 최근에는 이런 형태를 여러분도 많이 보셨을거라고 생각해요. 그러니 나중에 이런 기능도 구현해보면 재밌겠죠? 이건 여러분과 저의 마숙으로 남겨놓기로~~~

스프링으로 코드를 작성하다보니 너무 길어져서 전체 코드를 작성하지 못했습니다. 전체 코드를 확인하거나, 클론하여 이것저것 만져보고 싶으신 분은 (https://github.com/HAANJM/SSAFY_MINI_SPRING) 를 참고해주세요. 클론해서 새로운 레포지토리를 파셔도 괜찮습니다.

어느새 1학기 마지막 관통 프로젝트가 눈에 보일 정도로 많은 시간이 지났습니다. 우리는 아직 제대로 된 로그인과 회원가입을 해본 적이 없지만, 이렇게 하나씩 하다보면 멋진 회원가입과 로그인도 완성할 수 있고, 이런 경험으로 다른 기능도 멋지게 완성할 수 있을거라 생각합니다. 질문은 언제든지 환영이며 복사 붙여넣기보다는 조금이라도 작성해보는게 더 도움이 될 거라 생각해요! 다음에는 회원 탈퇴, 회원 정보 수정 등을 해볼까 하는데 혹시 제안이 있으시면 역시 언제든지 말해주세요 환영입니다.

그러니 오늘도 여러분 코드를 많이 작성해보시길 바라겠습니다!!! **할수있다!! 아무튼 할수있다!!!**

