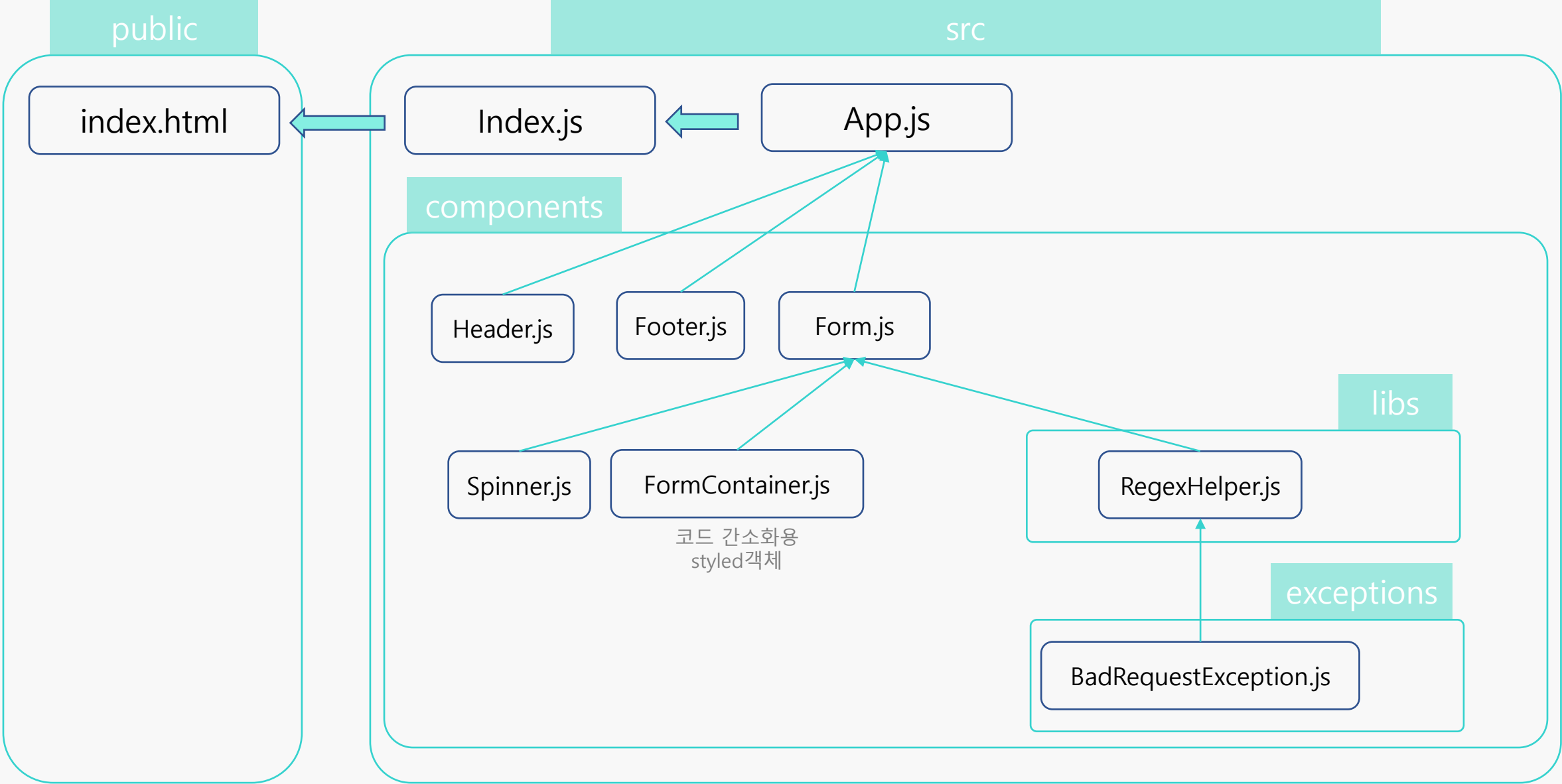


풀스택 자바(JAVA)웹개발자(프론트엔드&백엔드)_A

React 네이버 회원가입 페이지 클론코딩

박정모

소스코드 구조



최종 결과

원본 페이지

NAVER

아이디

비밀번호

비밀번호 재확인

이름

생년월일

년(4자)

월

일

성별

성별

본인 확인 이메일(선택)

휴대전화

대한민국 +82

인증번호 받기

가입하기

클론 페이지

NAVER

아이디

비밀번호

비밀번호 재확인

이름

생년월일

년(4자)

월

일

성별

성별

본인 확인 이메일(선택)

휴대전화

대한민국 +82

인증번호 받기

가입하기

최종 결과

아이디 입력 안했을 때

아이디

@naver.com

필수 정보입니다.

아이디만 입력했을 때

아이디

eeeeeeee

@naver.com

멋진 아이디네요!

비밀번호



필수 정보입니다.

비밀번호 입력 양식이 틀릴 때

아이디

eeeeeeee

@naver.com

멋진 아이디네요!

비밀번호



8~16자 영문 대 소문자, 숫자, 특수문자를 사용하세요.

비밀번호 확인이 틀릴 때

아이디

eeeeeeee

@naver.com

멋진 아이디네요!

비밀번호



비밀번호 재확인



비밀번호가 일치하지 않습니다.

최종 결과

아이디 비밀번호 입력 완료

아이디

eo0eo0eo

@naver.com

멋진 아이디네요!

비밀번호

.....



비밀번호 재확인

.....



이름

필수 정보입니다.

생일 입력

이름

박정모

생년월일

년(4자)

월



일

태어난 년도 4자리를 정확하게 입력하세요.

생년월일

1990

월



일

태어난 월을 선택하세요.

생년월일

1990

5



일

태어난 일(날짜) 2자리를 정확하게 입력하세요.

최종 결과

성별 입력 후

성별

선택 안함



본인 확인 이메일(선택)

선택 입력

휴대전화

대한민국 +82



전화번호 입력

인증번호 받기

인증번호 입력하세요

필수 정보입니다.

이메일 입력시

본인 확인 이메일(선택)

asd|

이메일 주소를 다시 확인해주세요.

휴대전화

대한민국 +82



전화번호 입력

인증번호 받기

인증번호 입력하세요

필수 정보입니다.

최종 결과

가입하기

비밀번호

localhost:3000 내용:
회원가입이 완료되었습니다.

확인

비밀번호

이름

박정모

생년월일

1990

5

12

성별

선택 안함

본인 확인 이메일(선택)

asd@naver.com

휴대전화

대한민국 +82

01072828132

인증번호 받기

12423

가입하기

이용약관 | 개인정보 처리방침 | 책임의 한계와 법적고지 | 회원정보 고객센터

NAVER Copyright NAVER Corp. All Rights Reserved.

json-server

http://localhost:4000/member

localhost:4000/member

```
[
  {
    userid: "eoeoeoeo",
    password: "Abcd!234",
    name: "박정모",
    birthday: 1990512,
    gender: "unknown",
    email: "asd@naver.com",
    phone_no: "01072828132",
    id: 1
  }
]
```

상세 코드

App.js

```
/**
 * @filename: App.js
 * @description: 컴포넌트 출력 페이지
 * @author: 박정모(oooperbjm@gmail.com)
 */

import React from "react";
import Header from "../components/Header";
import Form from "../components/Form";
import Footer from "../components/Footer";
import { createGlobalStyle } from "styled-components";

const GlobalStyles = createGlobalStyle`
  body {
    background-color: #F5F6F7;
  }
  h3, button, input, select, table, textarea, span, a {
    font-family: Dotum, '돋움', Helvetica, sans-serif;
  }
  label {
    cursor: pointer;
  }
  h3 {
    font-size: 14px;
    font-weight: 700;
    margin: 0;
  }

  // input박스 입력시 자동으로 배경색상 지정되는 기능 해제
  input:-webkit-autofill {
    -webkit-box-shadow: 0 0 0 1000px white inset;
  }
`;

const App = () => {
  return (
    <div style={{ width: "100%" }}>
      <GlobalStyles />
      <Header />
      <Form />
      <Footer />
    </div>
  );
};

export default App;
```

Header.js

```
/**
 * @filename: Header.js
 * @description: 웹브라우저 최상단 logo 부분
 * @author: 박정모(oooperbjm@gmail.com)
 */

import React from 'react';
import styled from 'styled-components';
import logo from '../assets/img/logo.png';

const HeaderContainer = styled.div`
  width: 768px;
  height: 116px;
  margin: auto;

  div {
    width: 240px;
    height: 44px;
    position: relative;
    top: 52px;
    margin: auto;
    background: url(${logo}) no-repeat;
    background-size: 240px auto;
    a {
      display: block;
      width: 100%;
      height: 100%;
    }
  }
`;

const Header = () => {
  return (<HeaderContainer>
    <div>
      <a href='https://www.naver.com'></a>
    </div>
  </HeaderContainer>);
};

export default Header;
```


상세 코드

Footer.js

```
/**
 * @filename: Footer.js
 * @description: 웹브라우저 최하단 footer 부분
 * @author: 박정호(ooperbjm@gmail.com)
 */

import React from "react";
import styled from "styled-components";
import logo from "../assets/img/logo.png";

const FooterContainer = styled.div`
  margin-top: 20px;
  .footer1 {
    width: 460px;
    margin: auto;
    font-size: 12px;
    text-align: center;
    letter-spacing: -0.5px;
    a {
      color: #666;
      text-decoration: none;
      letter-spacing: 0.1px;
    }
    .bld {
      font-weight: bold;
    }
  }

  .footer2 {
    display: block;
    width: 460px;
    margin: auto;

    .footerbg {
      background: url(${logo}) no-repeat 0px -19px;
      width: 70px;
      height: 20px;
      display: block;
      position: relative;
      left: 80px;
      top: -3px;
      float: left;
      background-size: 100px 40px;
      z-index: 10;
    }
  }
  p {
    position: relative;
    width: 100%;
    font-size: 9px;
    font-family: "Verdana", serif;
    line-height: 14px;
    top: -4px;
    right: -80px;
    font-style: normal;
    letter-spacing: 0.05em;
    span {
      font-weight: bold;
    }
  }
};
```

```
const Footer = () => {
  return (
    <FooterContainer>
      <div className="footer1">
        <a href="https://policy.naver.com/rules/service.html">이용약관 | </a>
        <a href="https://policy.naver.com/policy/privacy.html" className="bld">
          개인정보처리방침
        </a>
        <a href="https://policy.naver.com/rules/disclaimer.html"> | 책임의 한계와 법적고지 | </a>
        <a href="https://help.naver.com/support/service/main.help?serviceNo=532&_membership_p.membership_p.membership_26&from=alias" target="_blank" rel=
"noreferrer">
          회원정보 고객센터
        </a>
      </div>
      <address className="footer2">
        <div className="footerbg"></div>
        <p>
          Copyright <span>NAVER Corp.</span> All Rights Reserved.
        </p>
      </address>
    </FooterContainer>
  );
};

export default Footer;
```

RegexHelper.js

```
/**
 * @filename      : RegexHelper.js
 * @author       : 박정모 (oooperbjm@gmail.com)
 * @description   : 정규표현식 검사 수행
 */

import BadRequestException from '../exceptions/BadRequestException';

class RegexHelper {
  // constructor() {}

  success(id, msg2){
    document.getElementById(id).style.color='green';
    document.getElementById(id).innerHTML=msg2;
  }
  fail(id, msg){
    document.getElementById(id).style.color='red';
    document.getElementById(id).innerHTML=msg;
  }

  /**
   * 값의 존재 여부를 검사한다.
   * @param {HTMLElement} field 검사할 대상에 대한 <input>요소의 DOM 객체
   * @param {string} msg 실패시 표시할 메시지 내용
   * @param {string} msg2 성공시 표시할 메시지 내용
   * @param {string} id 메시지를 표시할 대상의 id값
   */
  value(field, msg, msg2, id) {
    const content = field.value;

    if (content === undefined || content === null || (typeof content === "string" && content.trim().length === 0)) {
      this.fail(id, msg);
      throw new BadRequestException(msg, field);
    } else {
      this.success(id, msg2);
    }

    return true;
  }
}
```

msg2와 id값을 파라미터로 주고
success와 fail 메서드를 추가하여
alert 대신 input태그 아랫줄에
msg를 출력했습니다.

이하 코드도 같은 방식으로 수정

상세 코드

Form.js(1)

```
/**
 * @filename: Form.js
 * @description: 회원가입 양식 부분
 * @author: 박정모(oooperbjm@gmail.com)
 */

import React, { useRef } from 'react';
import useAxios from 'axios-hooks';
import FormContainer from './FormContainer';
import Spinner from './Spinner';
import regexHelper from '../libs/RegexHelper';

// 1~12의 숫자를 저장하기 위한 배열(생년월일)
const month = [];
for (let i = 1; i < 13; i++) month.push(i);

const Form = () => {

  // 비밀번호 우측 로그인 컨트롤용 ref
  const icon1 = useRef();
  const icon2 = useRef();

  /** 국가별 지역 코드를 가져오기 위한 요청 */

  // 출처 : https://sleepydeveloper.tistory.com/entry/2020-국제전화-나라번호-목록-수집하기

  const [{data}] = useAxios('http://localhost:4000/list');

  // 국가명 가나다 순으로 정렬
  let country = null;
  if (data != null) {
    country = data.sort((a, b) => {
      return a.country < b.country ? -1 : a.country > b.country ? 1 : 0;
    });
  }
}
```

이름

그레나다 +1-473
그리스 +30
그린란드 +299
생기니 +224
기니비사우 +245
나미비아 +264
나우루 +674
나이지리아 +234
성남수단 +211
남아프리카 공화국 +27
남오세티야 +995-34
네덜란드 +31
네팔 +977
본노르웨이 +47
누벨칼레도니 +687
뉴질랜드 +64
니우에 +683
니제르 +227
후니카라과 +505
대한민국 +82
대한민국 +82

상세 코드

Form.js(2)

```
/** 취득한 데이터를 POST하는 axios-hook */
// manual 설정으로 수동적용
const [{loading}, refetch] = useAxios({
  method: 'POST',
  url: 'http://localhost:4000/member',
}, {manual: true});

/** <form>의 submit 이벤트 */
const onSubmit = (e) => {
  e.preventDefault();

  // 이벤트가 발생한 폼 객체
  const current = e.target;

  // 유효성 검사
  try {
    regexHelper.value(current.id, '필수 정보입니다.', '멋진 아이디네요!', 'idError');
    regexHelper.engNum(current.id, 'id는 영어나 숫자로 입력하세요', '멋진 아이디네요!', 'idError');
    regexHelper.value(current.password, '필수 정보입니다.', '', 'password_error');
    regexHelper.navPw(current.password, '8~16자 영문 대 소문자, 숫자, 특수문자를 사용하세요.', '', 'password_error');
    if (regexHelper.navPw(current.password, '8~16자 영문 대 소문자, 숫자, 특수문자를 사용하세요.', '', 'password_error')) {
      icon1.current.style.backgroundColor = '-54px -27px';
    }
    regexHelper.value(current.password_check, '필수 정보입니다.', '', 'password_check_error');
    regexHelper.compareTo(current.password_check, current.password, '비밀번호가 일치하지 않습니다.', '', 'password_check_error');
    if (regexHelper.compareTo(current.password_check, current.password, '비밀번호가 일치하지 않습니다.', '', 'password_check_error')) {
      icon2.current.style.backgroundColor = '-81px 0';
    } else {
      icon2.current.style.backgroundColor = '-27px 0';
    }
    regexHelper.value(current.name, '필수 정보입니다.', '', 'name_error');
    regexHelper.value(current.birth_year, '태어난 년도 4자리를 정확하게 입력하세요.', '', 'birth_error');
    regexHelper.value(current.birth_month, '태어난 월을 선택하세요.', '', 'birth_error');
    regexHelper.value(current.birth_date, '태어난 일(날짜) 2자리를 정확하게 입력하세요.', '', 'birth_error');
    regexHelper.value(current.gender, '필수 정보입니다.', '', 'gender_error');
    if (current.email.value) {
      regexHelper.email(current.email, '이메일 주소를 다시 확인해주세요.', '', 'email_error');
    }
    regexHelper.value(current.phone_no, '필수 정보입니다.', '', 'phone_error');
    regexHelper.value(current.certification_num, '인증이 필요합니다.', '', 'certification_error');
  } catch (e) {
    e.field.focus();
    return;
  }
}
```

상세 코드

Form.js(3)

```
// 입력받은 값 취득
const id = current.id.value;
const password = current.password_check.value;
const name = current.name.value;
const birth_year = current.birth_year.value;
const birth_month = current.birth_month.value;
const birth_date = current.birth_date.value < 10 ? '0' + current.birth_date.value : current.birth_date.value;
const gender = current.gender.value;
const email = current.email.value;
const phone_no = current.phone_no.value;

let json = null;

// POST refetch를 위한 async-await
(async () => {
  try {
    const response = await refetch({
      data: {
        userid: id,
        password: password,
        name: name,
        // 생일은 8자리 생년월일로 결합하여 저장
        birthday: parseInt(birth_year + birth_month + birth_date),
        gender: gender,
        email: email,
        phone_no: phone_no
      }
    });

    json = response.data;
  } catch (e) {
    console.error(e);
    window.alert(`[${e.response.status}] ${e.response.statusText}\n${e.message}`);
  }

  if (json !== null) {
    window.alert('회원가입이 완료되었습니다.');
```

// 저장 후 input 초기화를 위한 페이지 새로고침

// window.location.href='https://www.naver.com';

window.location.reload();

```
  }
})();
}
```

상세 코드

Form.js(4)

```
return (<FormContainer>
  <Spinner visible={loading}/>
  <div className='content'>
    <form onSubmit={onSubmit}>
      <div className='form1'>
        <div className='input_id'>
          <h3><label htmlFor='id'>아이디</label></h3>
          <span className='inputWrapper'>
            <input type='text' id='id' name='id' />
            <span className='domain'>@naver.com</span>
          </span>
          <span className='error_message' id='idError'></span>
        </div>

        <div className='input_id'>
          <h3><label htmlFor='password'>비밀번호</label></h3>
          <span className='inputWrapper'>
            <input type='password' id='password' name='password' maxLength='16' minLength='8' />
            <span className='icon' ref={icon1}></span>
          </span>
          <span className='error_message' id='password_error'></span>

          <h3><label htmlFor='password_check'>비밀번호 재확인</label></h3>
          <span className='inputWrapper'>
            <input type='password' id='password_check' name='password_check' maxLength='16'
              minLength='8' />
            <span className='icon check' ref={icon2}></span>
          </span>
          <span className='error_message' id='password_check_error'></span>
        </div>
      </div>

      <div className='form1 form_padding'>
        <div className='input_id'>
          <h3><label htmlFor='name'>이름</label></h3>
          <span className='inputWrapper'>
            <input type='text' id='name' name='name' maxLength='40' />
          </span>
          <span className='error_message' id='name_error'></span>
        </div>
      </div>
    </form>
  </div>
</FormContainer>
)
```

```
<div className='input_id'>
  <h3><label htmlFor='birth_year'>생년월일</label></h3>
  <div className='flex_box'>
    <span className='inputWrapper small'>
      <input className='small_input' type='text' id='birth_year' name='birth_year'
        placeholder='년(4자)' maxLength='4' minLength='4' />
    </span>
    <span className='inputWrapper small'>
      <select id='birth_month' name='birth_month' defaultValue=''>
        <option value=''>월</option>
        {month.map((v, i) => {
          return <option value={v} key={i}>
            {v}
          </option>
        })}
      </select>
    </span>
    <span className='inputWrapper small'>
      <input className='small_input' type='text' id='birth_date' name='birth_date'
        placeholder='일' maxLength='2' minLength='1' />
    </span>
  </div>
  <span className='error_message' id='birth_error'></span>
</div>

<div className='input_id'>
  <h3><label htmlFor='gender'>성별</label></h3>
  <span className='inputWrapper'>
    <select className='select_full' id='gender' name='gender'>
      <option value=''>성별</option>
      <option value='male'>남자</option>
      <option value='female'>여자</option>
      <option value='unknown'>선택 안함</option>
    </select>
  </span>
  <span className='error_message' id='gender_error'></span>
</div>
```

상세 코드

Form.js(5)

```
<div className='input_id'>
  <h3><label htmlFor='email'>본인 확인 이메일<span
    style={{fontSize: '12px', color: '#8e8e8e', fontWeight: '400'}}>(선택)</span></label></h3>
  <span className='inputWrapper'>
    <input type='text' id='email' name='email' defaultValue='' placeholder='선택입력' />
  </span>
  <span className='error_message' id='email_error'></span>
</div>

</div>

<div className='form1'>
  <div className='input_id'>
    <h3><label htmlFor='country_num'>휴대전화</label></h3>
    <span className='inputWrapper'>
      { /* selected가 아닌! defaultValue를 사용해야하지만, 작동이 되지 않아 부득이하게 사용함 */ }
      <select className='select_full' id='country_num' name='country_num'>
        {country && country.map((v, i) => {
          return (
            v.code === '+82' ?
              <option value={v.code} key={i} selected>{v.country} {v.code}</option> :
              <option value={v.code} key={i}>{v.country} {v.code}</option>
            )
          )
        })}
      </select>
    </span>

    <h3><label htmlFor='password'></label></h3>
    <div className='position'>
      <span className='inputWrapper phone'>
        <input type='tel' id='phone_no' name='phone_no' placeholder='전화번호 입력' />
      </span>
      <span className='button'>인증번호 받기</span>
    </div>

    <h3><label htmlFor='password'></label></h3>
    <span className='inputWrapper'>
      <input type='text' id='certification_num' name='certification_num'
        placeholder='인증번호 입력하세요' />
    </span>
    <span className='error_message' id='phone_error'></span>
    <span className='error_message' id='certification_error'></span>
  </div>
</div>

  <button type='submit' className='btn_join'>가입하기</button>
</div>
</form>
</div>
</FormContainer>;
```

후기

기존에 사용했던 클론코딩의 경우 정규표현식 검사 결과를 alert창으로 띄웠었는데
이제는 어설피도 비슷하게라도 따라만들 수 있겠다는 생각에 아예 새로 만들었습니다.
입력하는 즉시 정규표현식 검사를 돌려서 실제 페이지처럼 바로바로 적용을 시키고 싶었으나
그 방법은 좀 더 궁리를 해봐야 할 것 같습니다.

정규표현식 검사 성공시 이벤트 처리 방법을 고민하다가 시간을 너무 많이 잡아먹었습니다.
regexHelper를 수정할 생각을 못하고 Form에서만 뒤적거리다가 그만....