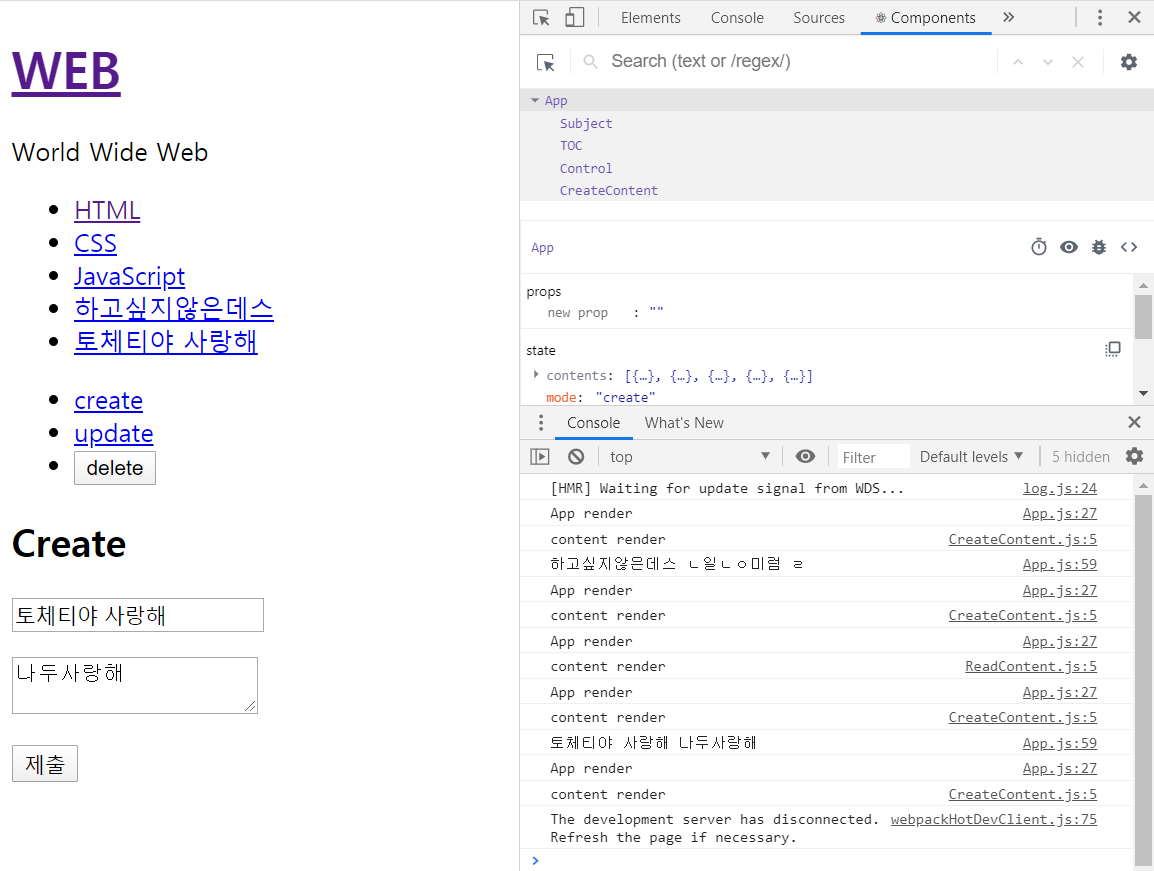
REACT 기술 공부 현황



REACT를 사용하여 기본적인 웹화면 설계해보는 중.

*import* React, { Component } *from* 'react';

*import* './App.css';

*import* TOC *from* "./components/TOC"

*import* Subject *from* "./components/Subject"

*import* ReadContent *from* "./components/ReadContent"

*import* Control *from* "./components/Control"

*import* CreateContent *from* './components/CreateContent';

class App extends Component{

  constructor(props){

    super(props);

    this.max\_content\_id = 3;

    this.state = {

      mode : 'create',

      selected\_content\_id : 2,

      subject : {title:'WEB', sub:'World Wide Web'},

      welcome : {title:'Welcome', desc:'Hello, React!!'},

      contents: [

        {id:1, title:'HTML', desc:'HTML is for infromation'},

        {id:2, title:'CSS', desc:'CSS is for Design'},

        {id:3, title:'JavaScript', desc:'JavaScript is for interactive'},

      ]

    }

  }

*//render()보다 먼저 나오게 할려면 contructor가 가장먼저 만들어져서 초기화를 담당하기에 constructor안에 코드를 쓴다.*

  render(){

    console.log('App render');

    var \_title, \_desc, \_article = null;

*if*(this.state.mode === 'welcome'){

      \_title = this.state.welcome.title;

      \_desc = this.state.welcome.desc;

      \_article =<ReadContent title={\_title} desc={\_desc}></ReadContent>

    }*else* *if*(this.state.mode === 'read'){

*for*(var i=0 ; i < this.state.contents.length ; i++){

        var data = this.state.contents[i];

*if*(data.id === this.state.selected\_content\_id){

          \_title = data.title;

          \_desc = data.desc;

*break*;

        }

      }

      \_article =<ReadContent title={\_title} desc={\_desc}></ReadContent>

    }*else* *if*(this.state.mode === 'create'){

      \_article = <CreateContent onSubmit={function(\_title, \_desc){

*//add content to this.state.contents*

        this.max\_content\_id = this.max\_content\_id+1;

*// this.state.contents.push({id:this.max\_content\_id,*

*// title:\_title,*

*// desc:\_desc});*

        var \_contents = this.state.contents.concat(

          {id:this.max\_content\_id,

            title:\_title,

            desc:\_desc}

        )

        this.setState({

          contents:\_contents

        });

        console.log(\_title, \_desc);

      }.bind(this)}></CreateContent>

    }

*return*(

      <div className="App">

        <Subject

          title={this.state.subject.title}

          sub={this.state.subject.sub}

          onChangePage={function(){

            this.setState({mode : 'welcome'});

          }.bind(this)}

        >

        </Subject>

        <TOC

          onChangePage={function(id){

            this.setState({

              mode:'read',

              selected\_content\_id : Number(id)

            });

          }.bind(this)}

          data={this.state.contents}

        ></TOC>

        <Control onChangeMode={function(\_mode){

          this.setState({

            mode : \_mode

          })

        }.bind(this)}></Control>

        {\_article}

      </div>

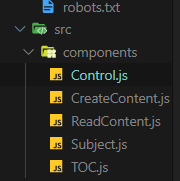
    );

  }

}

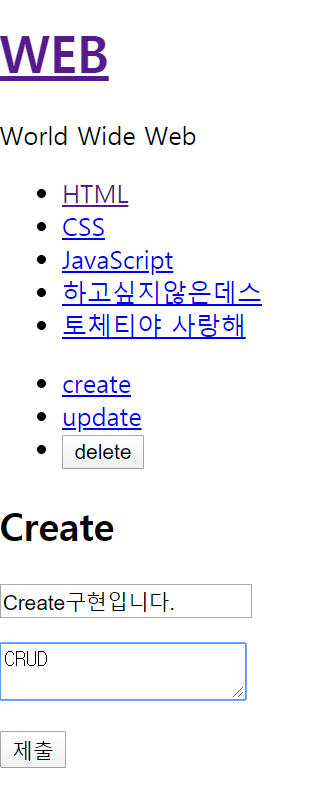
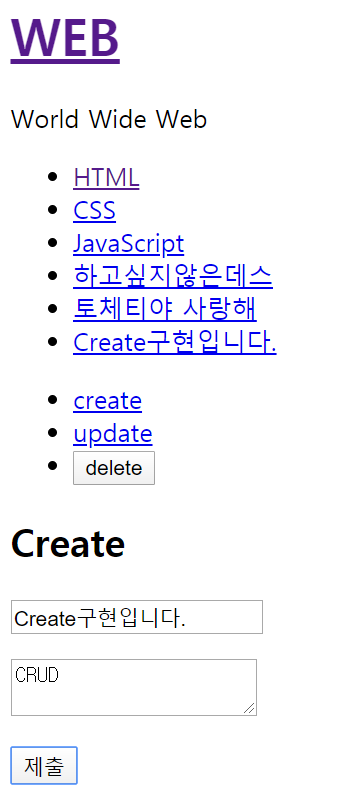
*export* *default* App;

웹화면을 구성하게 되는 App.js의 코드



App.js에서 import하여 화면 구현에 사용하는 컴퍼넌트들을 모아 놓은 폴더.

기본적인 웹 화면의 CRUD 구현과

 -> 

다음 주 까지는 우선 사용법을 익히고 css도 적용하여 A eye사이트 구현에 힘내보겠습니다.