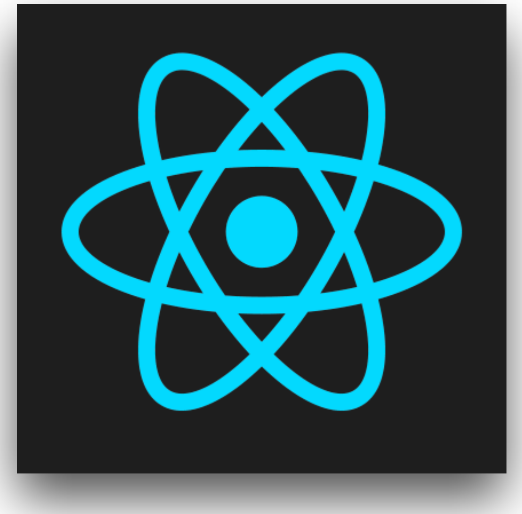
TO DO-List

한정윤



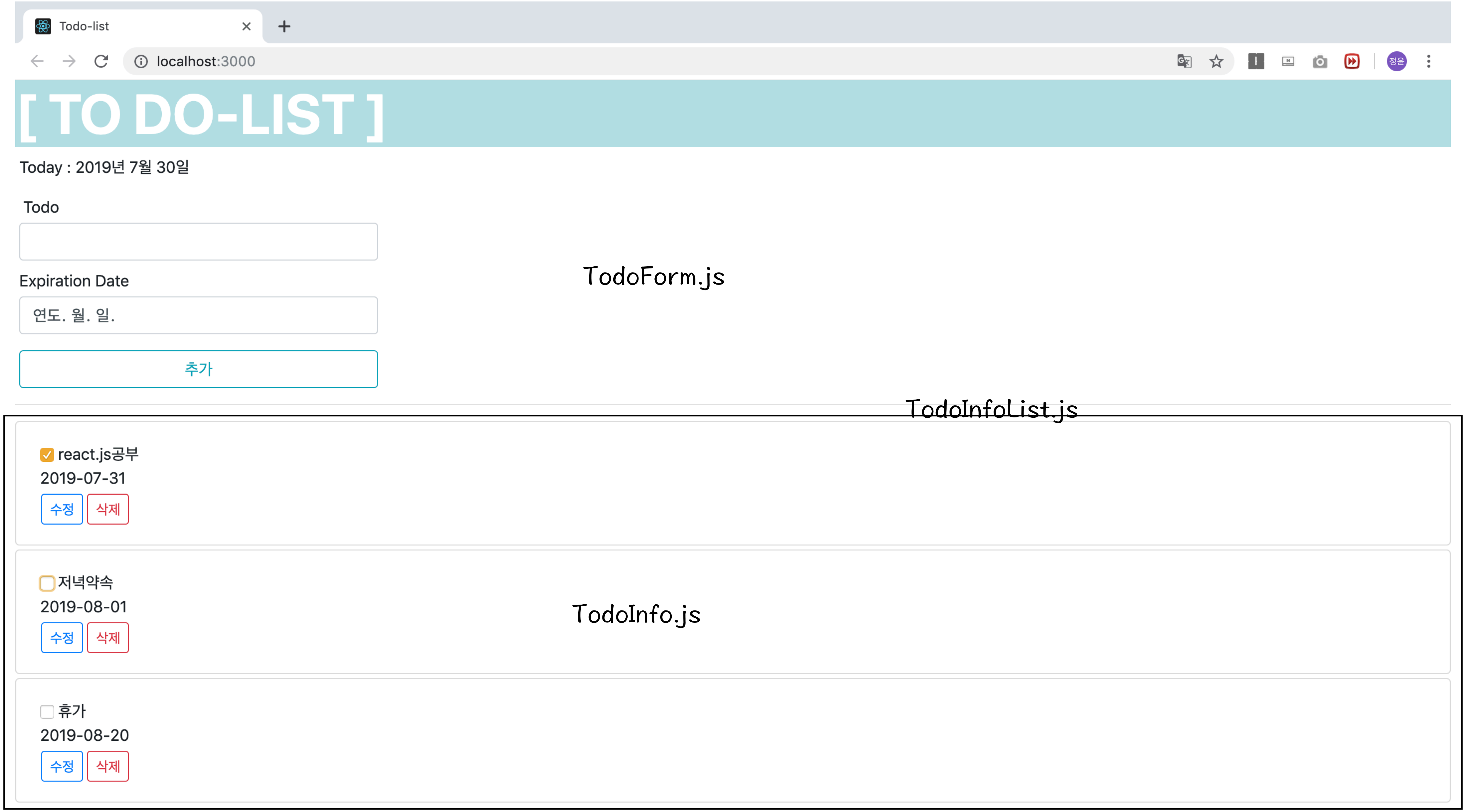
create-react-app

bootstrap4

구현기능

- Todo / Expiration Date
- 추가 / 수정 / 삭제 / 날짜별로 정렬
- 추가나 수정시 빈칸이 있으면 '입력하세요'라고 알려주기
- 오늘보다 이전 날짜를 선택하면 이전 날짜라고 알려주기
- ~~체크박스에 체크하면 줄 그어지기~~

```
ㄴsrc
  app.js
  ㄴ component
       TodoForm.js
       TodoInfo.js
       TodoInfoList.js
```



[TO DO-LIST]

Today : 2019년 7월 30일

Todo

Expiration Date

연도. 월. 일.

추가

TodoForm.js

TodoInfoList.js

☑ react.js공부
2019-07-31
수정 삭제

□ 저녁약속
2019-08-01
수정 삭제

□ 휴가
2019-08-20
수정 삭제

TodoInfo.js

App.js

추가

```
handleCreate = (data) => {
  const { information } = this.state;
  this.setState({
   information : information.concat({id : this.id++, ...data})
  })
 }
```

수정

```
handleUpdate = (id, data) => {
  const {information} = this.state;
  this.setState({
   information : information.map(
    info => id === info.id ? { ...info, ...data} : info
   )
  })
 }
```

삭제

```
handleRemove = (id) => {
  const {information} = this.state;
  this.setState({
   information : information.filter(info => info.id !== id)
  })
 }
```

정렬

```
function Sorting(arr){
 arr.sort((a,b) => {
  return new Date(a.expired).getTime() - new Date(b.expired).getTime();
 }).sort();
}
```

TodoForm.js

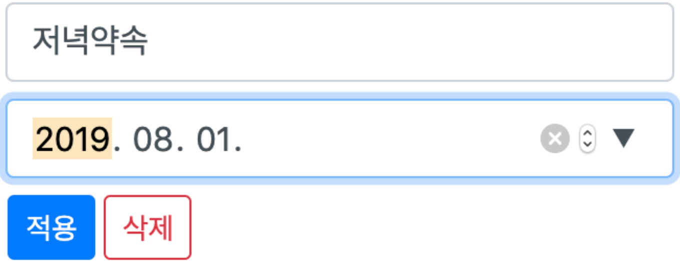
```
class TodoForm extends Component{
    state = {
        todo : '',
        expired : '',
    }



handleSubmit = (e) => {
        e.preventDefault(); // To prevent page reloading
        if(this.state.todo === "" || this.state.expired === ""){
            alert('입력하세요');
        }
        else{
            var date = new Date(this.state.expired).getTime();
            var today = new Date().getTime()-(24*60*60*1000);
            if(today > date){
                alert('선택한 날짜가 오늘보다 이전입니다.');
            }
            else{
                this.props.onCreate(this.state);
            }
        }
        // 상태 초기화
        this.setState({
            todo : '',
            expired : '',
        })
    }
```

TodoInfo.js

```
class TodoInfo extends Component{
    static defaultProps = {
        info : {
            todo : 'todo',
            expired : '',
            id : 0,
        }
    }
    state = {
        editing : false,
        todo : '',
        expired : '',
    }
    shouldComponentUpdate(nextProps, nextState){
        if(!this.state.editing && !nextState.editing &&
           nextProps.info === this.props.info){
            return false;
        }
        return true;
    }
    handleToggleEdit = () => {
        const {editing} = this.state;
        this.setState({editing : !editing});
    }
```

```
componentDidUpdate(prevProps, prevState){
        const {info, onUpdate} = this.props;
        if(!prevState.editing && this.state.editing){
            this.setState({
                todo: info.todo,
                expired: info.expired,
            })
        }
        if(prevState.editing && !this.state.editing){
            if(this.state.todo === "" || this.state.expired === ""){
                alert('입력하세요');
            }
            else{
                var date = new Date(this.state.expired).getTime();
                var today = new Date().getTime()-(24*60*60*1000);
                if(today > date){
                    alert('선택한 날짜가 오늘보다 이전입니다.');
                }
                else{
                    onUpdate(info.id, {
                        todo: this.state.todo,
                        expired: this.state.expired,
                    });
                }
            }
        }
    }
```

```
render() {
        const {editing} = this.state;
        // edit mode
        if(editing){
            return(
                <div className="card">
                    <div className="form-group card-body">
                        <div className="col-sm-3">
                            <input
                                type="text"
                                value={this.state.todo}
                                name="todo"
                                onChange={this.handleTodoChange}
                                className="form-control"
                            />
                        </div>
                        <div className="col-sm-3 mt-2">
                            <input
                                type="date"
                                value={this.state.expired}
                                name="expired"
                                onChange={this.handleExpiredChange}
                                className="form-control"
                            />
                        </div>
                        <div className="row mt-2 ml-3">
                            <button onClick={this.handleToggleEdit} className="btn btn-outline-primary btn-sm">적용</button>
                            <button onClick={this.handleRemove} className="btn btn-outline-danger btn-sm ml-1">삭제</button>
                        </div>
                    </div>
                </div>
            );
        }
```



저녁약속

2019. 08. 01.

```
// general mode
        const {todo, expired} = this.props.info;
        return (
            <div className="card mt-1">
                <div className="ml-1 card-body" >
                    <div className="form-inline">
                        <input type="checkbox"/>
                        <div className="ml-1">{todo}</div>
                    </div>
                    <div>{expired}</div>
                    <div className="row mt-1">
                        <button onClick={this.handleToggleEdit} className="btn btn-outline-primary btn-sm ml-3">수정</button>
                        <button onClick={this.handleRemove} className="btn btn-outline-danger btn-sm ml-1">삭제</button>
                    </div>
                </div>
            </div>
        );
    }
```

TodoInfoList.js

```
import TodoInfo from './TodoInfo';

class TodoInfoList extends Component {
    static defaultProps = {
        data : [],
        onRemove : () => console.warn('onRemove not defined'),
        onUpdate : () => console.warn('onUpdate not defined')
    }
    shouldComponentUpdate(nextProps, nextState){
        return nextProps.data !== this.props.data;
    }
    render() {
        const {data, onRemove, onUpdate} = this.props;
        const list = data.map(info => (
            <TodoInfo
                key={info.id}
                info={info}
                onRemove={onRemove}
                onUpdate={onUpdate}/>
            )
        );
        return(
            <div>
                {list}
            </div>
        );
    }
}
```