

Calender

(Using React.js)

Park Woo Jin

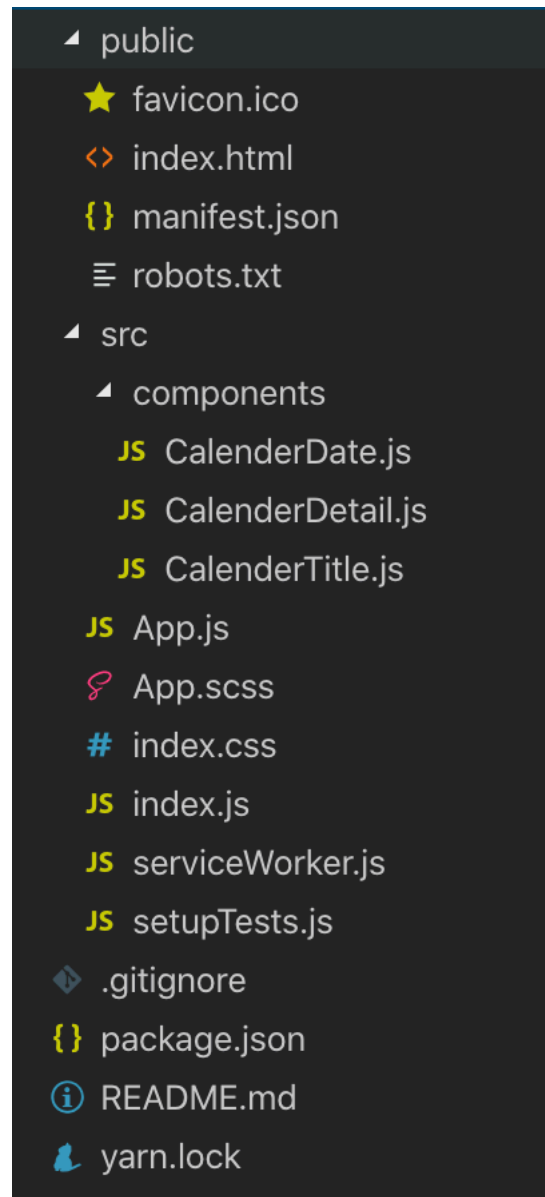
누구든지 하는 리액트 3편: JSX | VEI ×Konkuk-JJabaries/ReactJS-Stud ×React App ×스포츠포츠 : 네이버 스포츠 ×+

← → ↺localhost:3000☆S▼📷우진⋮

앱GV공부웃알고리즘포털메일SES — Boto 3 Doc...폴젝Commits · wjp103/...간편한 종합정보시스템네프

2020년 1월<오늘>

일	월	화	수	목	금	토
			1일	2일	3일	4일
5일	6일	7일	8일	9일	10일	11일
12일	13일	14일	15일	16일	17일	18일
19일	20일	21일	22일	23일	24일	25일
26일	27일	28일	29일	30일	31일	



Method	Description
getFullYear()	Get the year as a four digit number (yyyy)
getMonth()	Get the month as a number (0-11)
getDate()	Get the day as a number (1-31)
getHours()	Get the hour (0-23)
getMinutes()	Get the minute (0-59)
getSeconds()	Get the second (0-59)
getMilliseconds()	Get the millisecond (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.

```
//file :: /src/App.js

import React, { Component } from 'react';
import CalenderDetail from './components/CalenderDetail'
import CalenderTitle from './components/CalenderTitle'
import './App.scss';

class App extends Component{
  constructor(props){
    super(props);
    this.date = new Date();
    this.firstDate = new Date(this.date.getFullYear(), this.date.getMonth(), 1)
    this.lastDate = new Date(this.date.getFullYear(), this.date.getMonth()+1, 0)
  }

  render(){
    //calendar 는 0월부터 시작
    const today = this.date.getFullYear() + "년 " + (this.date.getMonth()+1)+"월";
    const lstDate = this.lastDate.getDate();
    const fstDay = this.firstDate.getDay();
    const lstDay = this.lastDate.getDay();
    return (
      <div className="App">
        <CalenderTitle
          date = { today }
        ></CalenderTitle>
        <CalenderDetail
          last = { lstDate }
          fstDate = { fstDay }
          lstDay = { lstDay }
        ></CalenderDetail>
      </div>
    );
  }
}

export default App;
```

```
//file:: /src/components/calenderTitle.js
import React, { Component } from 'react';

class CalenderTitle extends Component{
  render(){
    return (
      <>
        <div className="todayDate">{this.props.date}</div>
        <div className = "calenderTitle">
          <button className="arr_btn">&lt;</button>
          <button className="today">오늘</button>
          <button className="arr_btn">&gt;</button>
        </div>
      </>
    );
  }
}
export default CalenderTitle;
```

```
//file :: /src/components/CalenderDetail.js
import React, { Component } from 'react';
import CalenderDate from './CalenderDate'
import "../App.scss"

class CalenderDetail extends Component{

  render(){
    const { last, fstDate, lstDay } = this.props;
    let tmpArr = []
    let tmpLine = 0
    let tmpDict = {}

    for(let i=0;i<6-fstDate;i++){
      tmpArr.push(0);
    }
    for(let j=1; j<last+1; j++){
      tmpLine = Math.floor(j/7);
      if(tmpArr.length === 7){
        tmpDict[tmpLine] = {date: tmpArr};
        tmpArr = [];
      }
      tmpArr.push(j);
    }
    for(let i=0;i<6-lstDay;i++){
      tmpArr.push(i)
    }
    tmpDict[tmpLine] = {date: tmpArr}
  }
}
```

```
const calenderDate = Object.values(tmpDict).map((t,k) => {
  return (
    <CalenderDate key={k} value={t}></CalenderDate>
  )
});

return (
  <div className="calenderDetail">
    <tr>
      <th>일</th>
      <th>월</th>
      <th>화</th>
      <th>수</th>
      <th>목</th>
      <th>금</th>
      <th>토</th>
    </tr>
    { calenderDate }
  </div>
);
}

export default CalenderDetail;
```

```
//file :: /src/compoents/CalenderDate.js
import React, { Component } from 'react';

class CalenderDate extends Component{
  render(){
    const dates = Object.values(this.props.value.date).map((t, k) => {
      return (
        <td key={k} value={t}>
          { t === 0 ? ':' : t+'일' }
        </td>
      )
    })
    return (
      <tr>
        { dates }
      </tr>
    );
  }
}
export default CalenderDate;
```

```

body{
    background-color: #1E1F22;
    color: lightgrey;
}

.todayDate{
    display: inline-block;
    font-size: 40px;
    left: 13%;
}

.calenderTitle{
    float: right;
    .arr_btn, .today{
        background-color: #565759;
        color: lightgrey;
        font-size: 30px;
        border-radius: 8px;
    }
    .today{
        margin: 0px 10px;
        height: 40px;
        width: 85px;
    }
    .arr_btn{
        height: 40px;
        width: 40px;
    }
}

```

```

// calender detail
th{
    font-size: 30px;
    width: 150px;
    text-align: right;
}
td{
    border: 1px solid lightgrey;
    font-size: 15px;
    height: 120px;
    text-align: right;
    padding-top: 5px;
    padding-right: 10px;
}

.calenderDetail{
    position: absolute;
    left: 14%;
}

```



```
class App extends Component{
  constructor(props){
    super(props);
    this.date = new Date();
  }

  render(){
    return (
      <div className="App">
        <CalenderTitle date={this.date}></CalenderTitle>
        <CalenderDate></CalenderDate>
      </div>
    );
  }
}
export default App;
```

Error: Objects are not valid as a React child (found: Sat Jan 04 2020 20:05:06 GMT+0900 (한국 표준시)). ✕
If you meant to render a collection of children, use an array instead.

- in div (at CalenderTitle.js:7)
- in calenderTitle (at App.js:17)
- in div (at App.js:16)
- in App (at src/index.js:7)

Object.keys, values, entries

For plain objects, the following methods are available:

- `Object.keys(obj)` – returns an array of keys.
- `Object.values(obj)` – returns an array of values.
- `Object.entries(obj)` – returns an array of `[key, value]` pairs.

<https://javascript.info/keys-values-entries>