



Waffle Studio

Frontend Seminar - 3

김기완



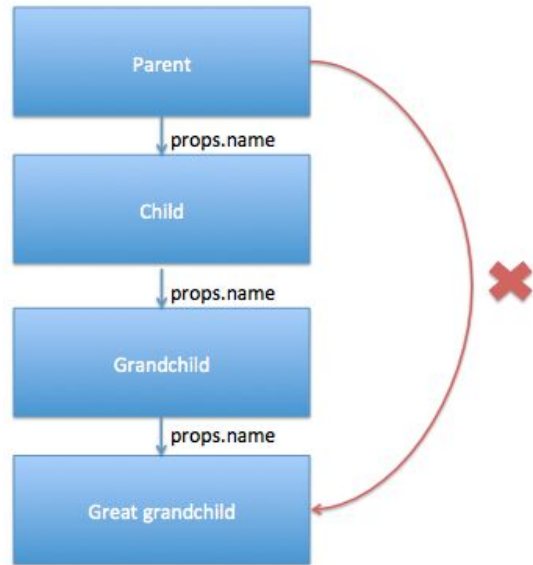
Contents

- 전역 값 관리
- Routing
- 비동기 처리

값 관리

지금까지는...

- **props** 사용
- 자식 component -> 손자 component -> ...





전역 값 관리

useContext

- 중간 component는 건너뛰고 값 전달
- createContext
- provider
- value

useContext Code Example



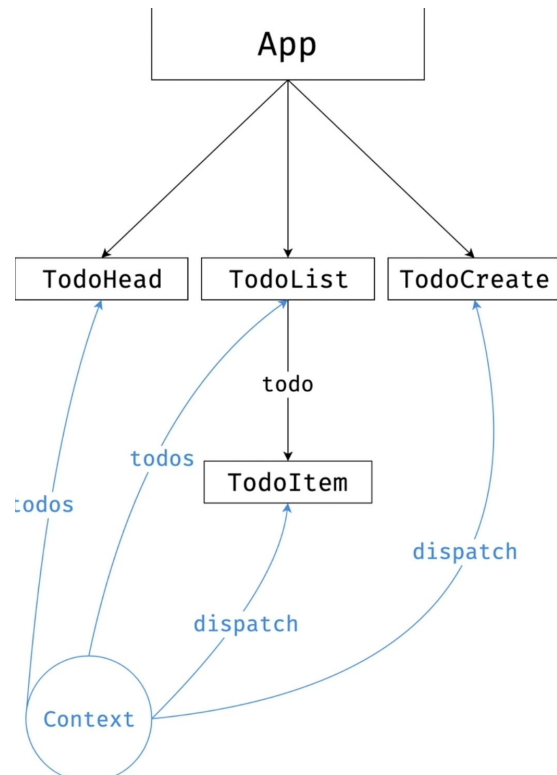
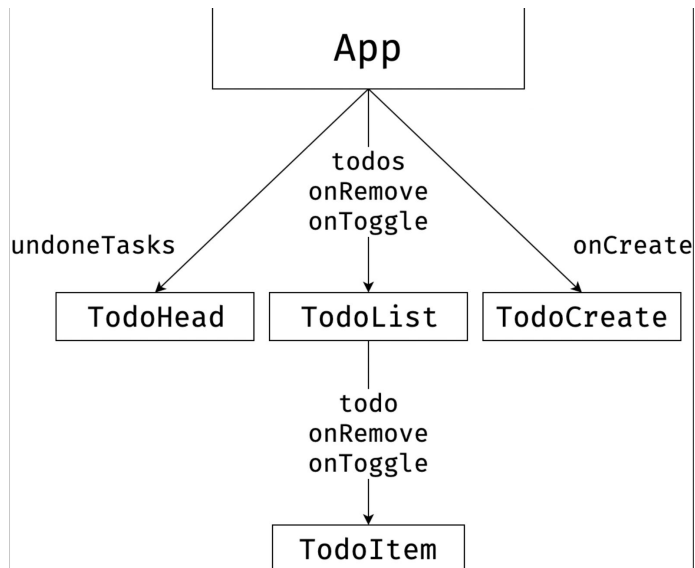
Context API

과제 2를 생각해봅시다

- '좋아요' 함수가 App -> List -> Item 순으로 전달

useState + useContext (+ useEffect)

- context + 값을 바꿀 함수



Context Code Example



Context API

무조건 Context 사용?

- 규모 작은 경우엔 상관 X

규모가 커질 수록 유용



Routing

어떤 주소에 어떤 UI를?

클라이언트 or 서버가 관리

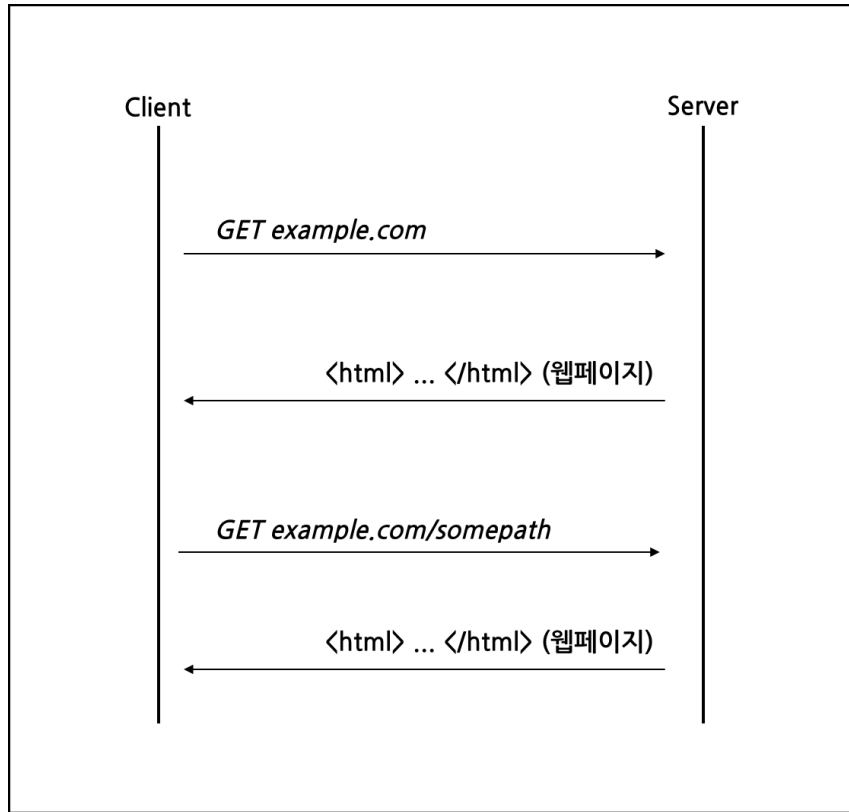
Non-SPA

서버가 라우팅을 담당

- HTML 제공
- 데이터베이스와 연동

새로운 데이터 요청

- 서버에서 새로운 HTML
- 서버 자원의 낭비



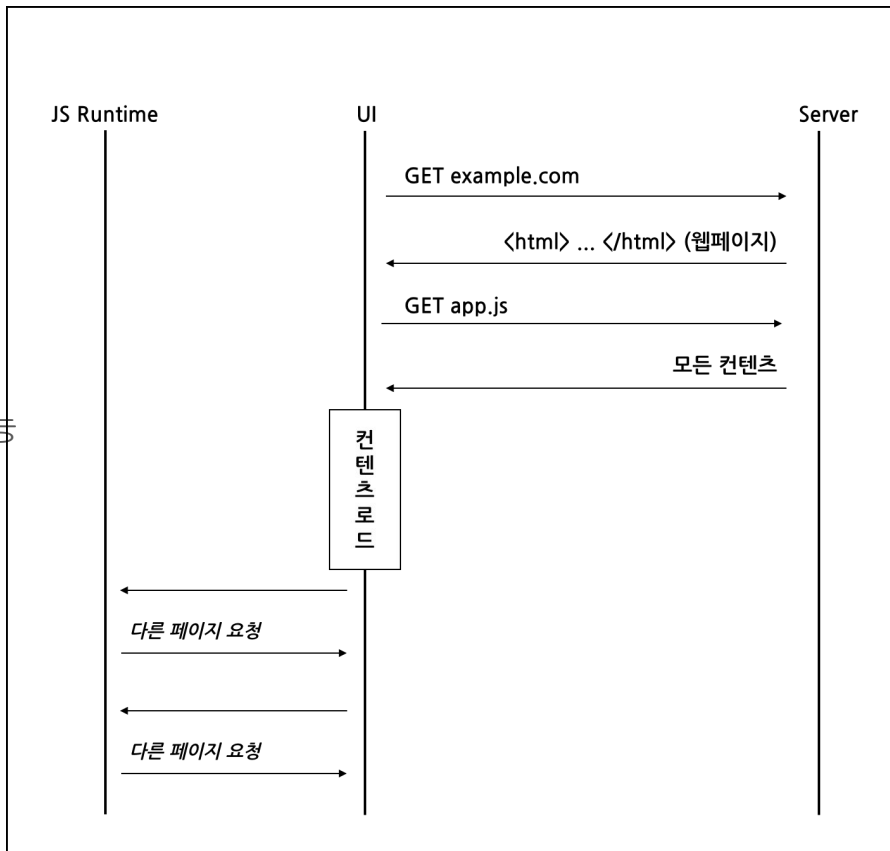
SPA

클라이언트에서 라우팅 담당

- 서버에 요청없이 페이지 제공 가능

새로운 데이터 요청

- API로 서버에 요청
- 필요한 정보만 제공





React-Router

대표적인 라우팅 라이브러리

컴포넌트를 기반으로 라우팅

```
yarn add react-router-dom
```

```
npm install react-router-dom
```



React-Router 주요 컴포넌트

BrowserRouter

- 가장 많이 사용
- HTML5 History API 사용
- 주소만 변경/ 페이지 리로드 X

`<BrowserRouter>`

```
basename: string  
getUserConfirmation: func  
forceRefresh: bool  
keyLength: number  
children: node
```



React-Router 주요 컴포넌트

HashRouter

- 옛날 브라우저 전용
- example.com/**#**/path/to/route
- 못생김

<HashRouter>

```
basename: string
getUserConfirmation: func
hashType: string
children: node
```



React-Router 주요 컴포넌트

MemoryRouter

- 브라우저의 주소와 무관
- 브라우저가 아닌 환경(리액트 네이티브)
- 임베디드 웹/앱

`<MemoryRouter>`

```
initialEntries: array  
initialIndex: number  
getUserConfirmation: func  
keyLength: number  
children: node
```




React-Router 주요 컴포넌트

Route

- 라우트를 정의
- 어떤 경로에 어떤 컴포넌트

`<Route>`

Route render methods

Route props

component

render: func

children: func

path: string

exact: bool

strict: bool

location: object

sensitive: bool



React-Router 주요 컴포넌트

Link

- Router의 주소를 변경
- a 태그로 이루어짐
- 새로그침 X

`<Link>`

`to: string`

`to: object`

`replace: bool`

`innerRef: function`

`others`



React Router 주요 컴포넌트

Switch

- 여러 Route 중 먼저 matching 되는 하나만 보여줌
- NotFound일 때 유용

React-router Basic Code Example



Parameter/Query

주소를 통해서 동적인 값

URL Parameter: /seminar/**frontend**

- id, userName 등의 정해진 데이터 조회
- useParams

URL Parameter Code Example



Parameter/Query

주소를 통해서 동적인 값

Query: /filter?class=frontend&sort_by=date

- 다양한 옵션으로 조회
- window.location.search
- 외부 라이브러리 사용(ex. **qs**)

Query Code Example



History

Component에서 Router에 직접 접근

- 함수 호출 시 특정 경로로 이동
- 뒤로 가기
- 페이지 이탈 방지

`useHistory()`



History

`goBack()`

`push() / replace()`

`block()`

History Code Example



Switch

여러 Route 중 먼저 matching 되는 하나만 보여줌

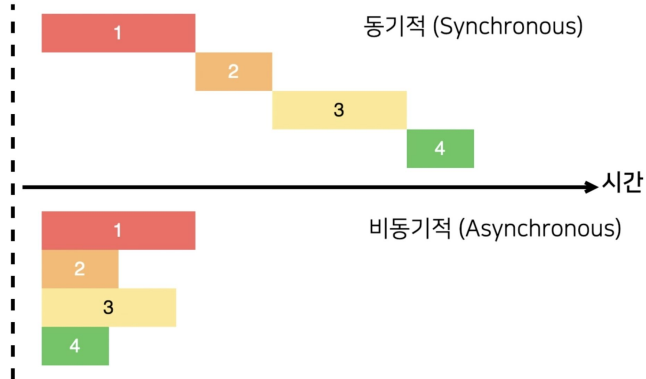
NotFound일 때 유용

Switch Code Example

비동기 처리

특정 코드의 연산이 끝나기 전에 다음 코드 실행

- AJAX 통신
- setTimeout
- 파일 읽기
- 암호화/복호화



```
console.log('one');
```

```
setTimeout(() => {  
  console.log('two');  
}, 0);
```

```
console.log('three');
```



비동기 처리 방식의 문제점 해결

1. Callback 함수
2. Promise
3. async/await



Callback Function


특정 로직이 끝나면 원하는 동작 실행

- AJAX 통신
- setTimeout
- map/forEach/filter

Callback Function Code Example

Callback 지옥

```
1 function hell(win) {
2   // for listener purpose
3   return function() {
4     loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {
5       loadLink(win, REMOTE_SRC+'/lib/async.js', function() {
6         loadLink(win, REMOTE_SRC+'/lib/easyXDM.js', function() {
7           loadLink(win, REMOTE_SRC+'/lib/json2.js', function() {
8             loadLink(win, REMOTE_SRC+'/lib/underscore.min.js', function() {
9               loadLink(win, REMOTE_SRC+'/lib/backbone.min.js', function() {
10                loadLink(win, REMOTE_SRC+'/dev/base_dev.js', function() {
11                  loadLink(win, REMOTE_SRC+'/assets/js/deps.js', function() {
12                    loadLink(win, REMOTE_SRC+'/src/' + win.loader_path + '/loader.js', function() {
13                      async.eachSeries(SRIPTS, function(src, callback) {
14                        loadScript(win, BASE_URL+src, callback);
15                      });
16                    });
17                  });
18                });
19              });
20            });
21          });
22        });
23      });
24    });
25  };
26 }
```





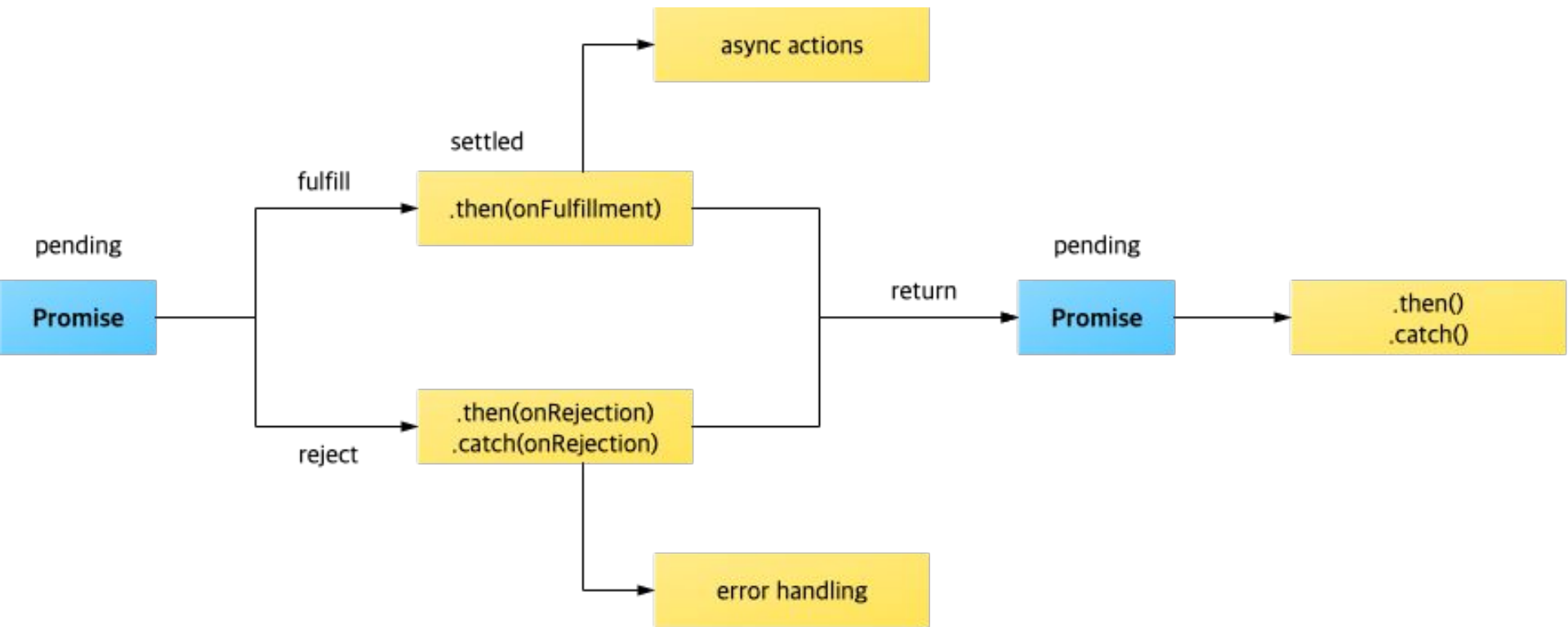
Promise

자바스크립트의 비동기 처리에 사용하는 객체

Callback 지옥을 방지

- resolve/reject
- .then()
- .catch()

값 공유, 에러 처리가 어려운 등의 단점



Promise Code Example



Async/Await

Promise를 쉽게 사용

- 함수가 끝날 때까지 정지
- 함수 선언 부: `async`
- Promise의 앞 부분: `await`
- 에러 캐치는 `try/catch` 사용

Async/Await Code Example
