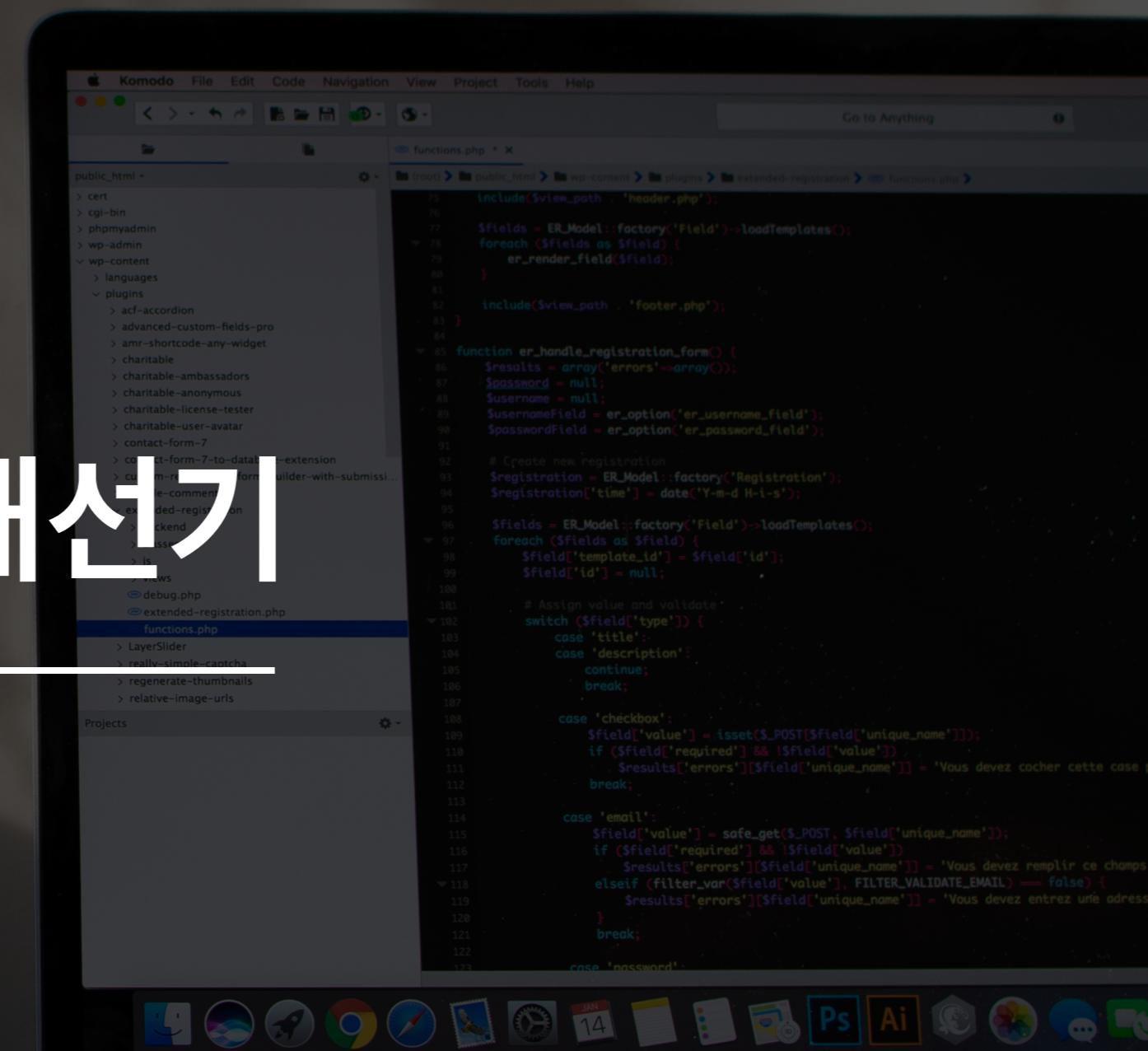


리액트 퍼포먼스 개선기

천민호

kesakiyo@naver.com



```
public_html
> cert
> cgi-bin
> phpmyadmin
> wp-admin
wp-content
> languages
plugins
> acf-accordion
> advanced-custom-fields-pro
> amr-shortcode-any-widget
> charitable
> charitable-ambassadors
> charitable-anonymous
> charitable-license-tester
> charitable-user-avatar
> contact-form-7
> contact-form-7-to-data-table-extension
> custom-metabox-builder-with-submission-validation
> e-commerce
> extended-registration
> LayerSlider
> really-simple-captcha
> regenerate-thumbnails
> relative-image-urls
debug.php
extended-registration.php
functions.php
Projects
```

```
functions.php * x
[root] > [public_html] > [wp-content] > [plugins] > [extended-registration] > [functions.php]
75 include($view_path . 'header.php');
76
77 $fields = ER_Model::factory('Field')->loadTemplates();
78 foreach ($fields as $field) {
79     er_render_field($field);
80 }
81
82 include($view_path . 'footer.php');
83
84
85 function er_handle_registration_form() {
86     $results = array('errors'=>array());
87     $password = null;
88     $username = null;
89     $usernameField = er_option('er_username_field');
90     $passwordField = er_option('er_password_field');
91
92     # Create new registration
93     $Registration = ER_Model::factory('Registration');
94     $Registration['time'] = date('Y-m-d H:i:s');
95
96     $fields = ER_Model::factory('Field')->loadTemplates();
97     foreach ($fields as $field) {
98         $field['template_id'] = $field['id'];
99         $field['id'] = null;
100
101         # Assign value and validate
102         switch ($field['type']) {
103             case 'title':
104                 case 'description':
105                     continue;
106                     break;
107
108             case 'checkbox':
109                 $field['value'] = isset($_POST[$field['unique_name']]);
110                 if ($field['required'] && !$field['value'])
111                     $results['errors'][$field['unique_name']] = 'Vous devez cocher cette case';
112                     break;
113
114             case 'email':
115                 $field['value'] = safe_get($_POST, $field['unique_name']);
116                 if ($field['required'] && !$field['value'])
117                     $results['errors'][$field['unique_name']] = 'Vous devez remplir ce champs';
118                 elseif (!filter_var($field['value'], FILTER_VALIDATE_EMAIL) == false)
119                     $results['errors'][$field['unique_name']] = 'Vous devez entrez une adresse email';
120                     break;
121
122             case 'password':
123
124         }
125     }
126
127     # Save registration
128     $Registration->save();
129
130     # Redirect to success page
131     header('Location: ' . get_permalink($Registration));
132
133     exit;
134 }
```

MacBook Pro

경력

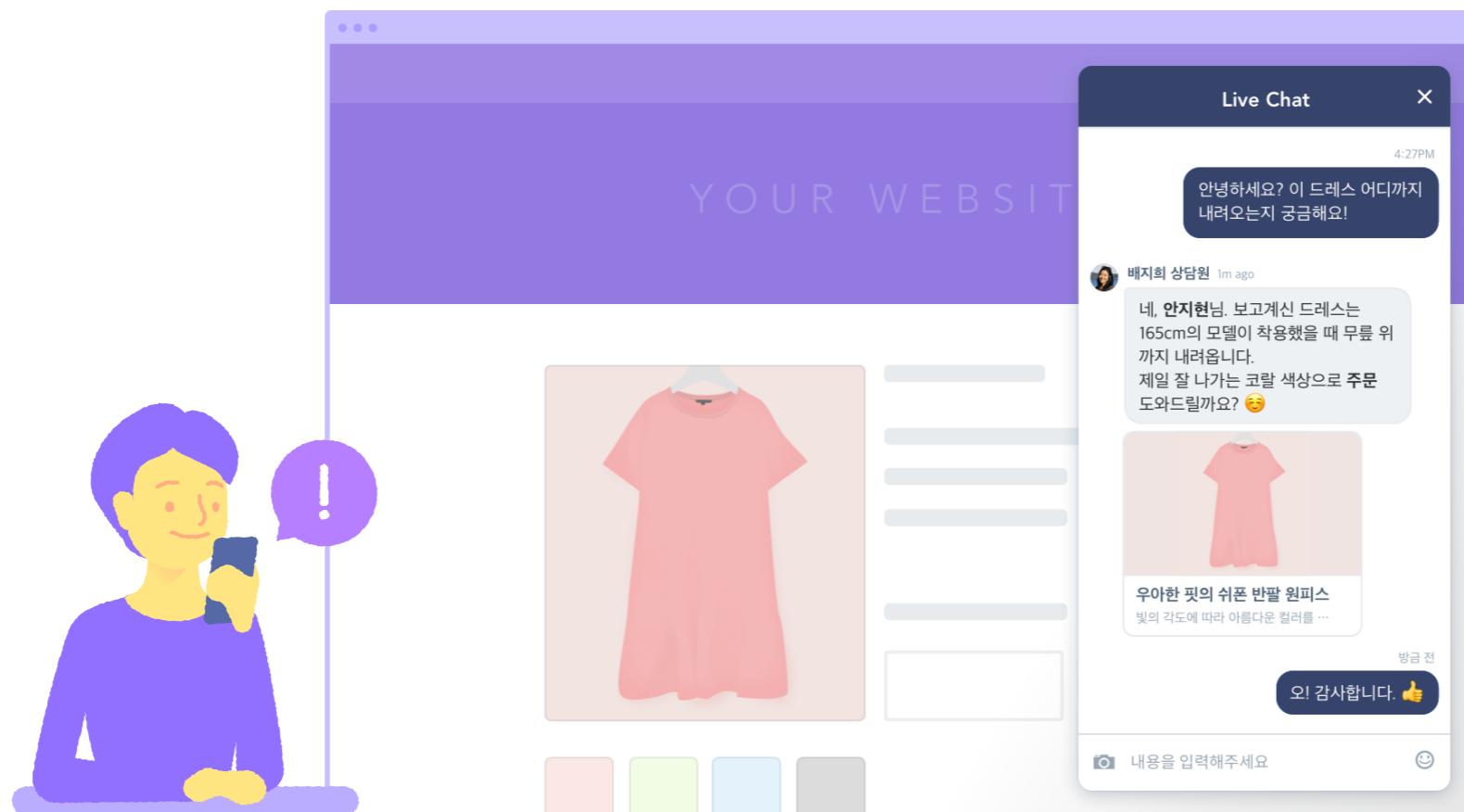
2016 | 제 1, 2회 삼성 대학생 프로그래밍 경시대회 입상

한국 대학생 프로그래밍 경시대회 은상

2017 ~ | 조이코퍼레이션 프론트엔드 개발자

채널: 온라인 접객 서비스

- 점원이 있는 오프라인 매장과는 달리 온라인 쇼핑몰의 방문객들은 홀로 방치됨
- 온라인 쇼핑몰도 점원이 있어 고객들과 대화할 수 있다면 고객들을 더 오래 머물게 하고 구매를 유도할 수 있음.



채널 데스크 (관리자 화면)

The screenshot displays the Zopim Channel Desk interface, which integrates various communication channels into a single dashboard.

Left Sidebar:

- 팀 챗 (Team Chat) with 5 unread messages
- 실시간 방문 (Real-time Visitors)
- CRM
- 검색 (Search)
- 설정 (Settings)

Top Bar:

- 진행중인 상담 (Active Chats) ▾
- 안지현 (Selected Chat)
- 상담 완료하기 (End Chat)
- 설정 (Settings)
- +1 (Additional agents)

Chat Panel (Anjiheon):

2017년 4월 7일

안지현 4:28pm

지난주 청담 매장에 사진으로 보낸 드레스 M사이즈가 없던데 혹시 들어왔나요??

배지희 @baebae 4:30pm
안녕하세요? 안지현님. 현재 고객님과 가까운 매장인, 청담, 잠실 롯데월드몰 매장엔 모든 사이즈에 재고가 들어와있는 상태네요^^ 다른 도움은 필요 없으신가요?

Bottom Input Field:

메시지를 입력해주세요...

Bottom Icons:

Reply, Mention, Emoticon, Insert, Lock

Right Side (Mobile View):

9:41 AM Koala 100%
휴대폰 번호 +82-10-4324-1248
최근 접속시간 4:28PM, 2015/10/2
최근 접속환경 Apple Safari
아이디 ringosting2...
아이피 주소 95.23.2.2
웹 세션 234
8월 6일
태그 VIP X 예전에_구입
Koala 3:32pm 안녕하세요?
질문이 있는데요~
안나현 @anna 4:12pm 네 무엇이 궁금하신가요? 😊
Koala 3:32pm 가격이 궁금합니다.
안나현 @anna 3:32pm 🔒
@jay 보시는대로 응대 부탁드립니다.
채널 가격문의입니다.

+ 내용을 입력하세요

가볍게 시작하기 좋은 리액트

- 친숙한 자바스크립트로 작성 가능
- HTML과 비슷한 JSX로 작성 가능
- 러닝커브가 낮아 쉽게 응용이 가능
- 컴포넌트로 관리를 하기 때문에 구조파악이 쉬움
- 이 외의 수많은 장점이 존재

리액트가 가지고 있는 단점

- 리액트는 프레임워크가 아닌 라이브러리
- 컨데이터나 컴포넌트냐 그것이 문제
- 앱의 규모가 조금만 커져도 느려지는 나의 앱을 발견

앱이 느려져 채팅이 불가능한 상황 팀 내부의 불만 폭주

“채널이 너무 느려 업무가 불가능하다.”

“채팅을 한번 치면 1~2초 뒤에 말을 또 칠 수 있다.”

어떤 컴포넌트가 문제인거지?

모든 컴포넌트에서?

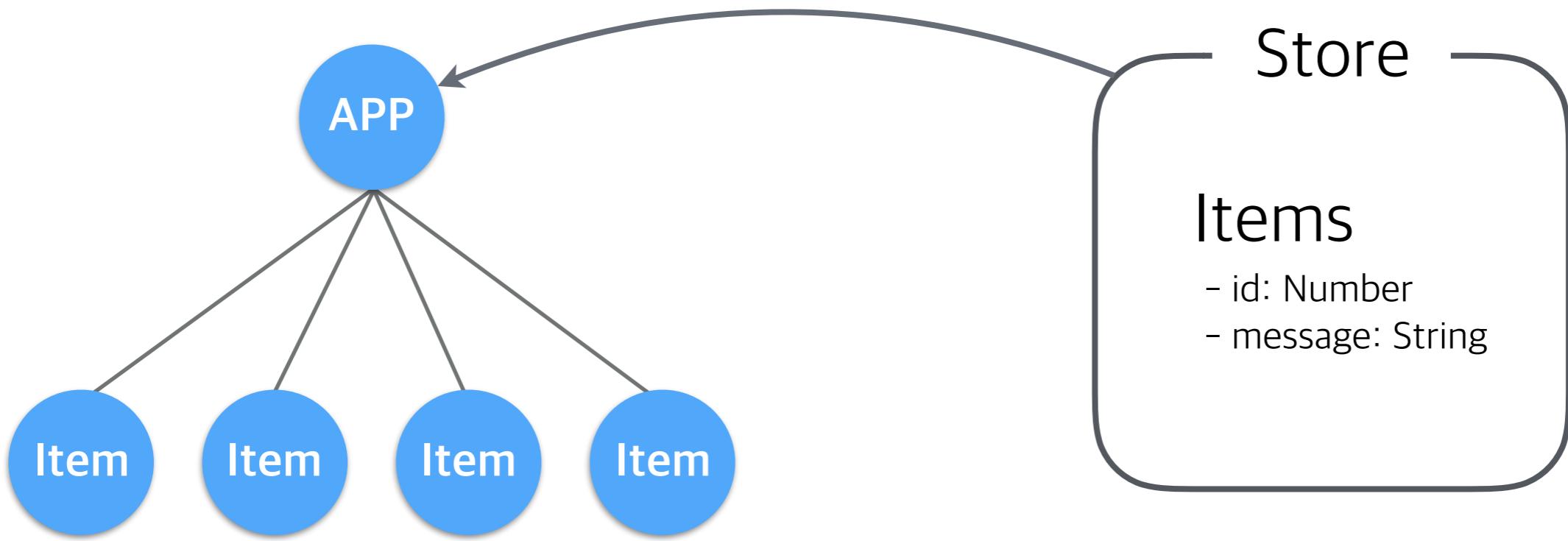
성능을 측정할 수 있는 도구들

- npm react-addons-perf
- Chrome extension react-devtools

react-addons-perf

- 그중에 사용할 것은 react-addons-perf
- 여러 API가 있지만 printWasted(…) API가 유용
- 예제를 통해서 어떻게 사용할지 알아보자

APP의 구조



- 새로운 Item이 추가되면 퍼포먼스에 어떤 영향을 미칠까?

App.js

- render function

```
render() {
  return (
    <div className={styles.wrapper}>
      <div className={styles.content}>
        {this.props.items.reverse().map(this.renderItem)}
      </div>
      <div
        className={styles.button}
        onClick={this.handleClick}>
        Add new message
      </div>
    </div>
  )
}
```

App.js

- renderItem function

```
renderItem(item) {  
  return (  
    <Item  
      key={item.get('id')}  
      item={item} />  
  )  
}
```

- handleClick function

```
handleClick() {  
  this.props.dispatch({  
    type: AT.CREATE_MESSAGE,  
    message: 'Hello React Korea!!',  
  })  
}
```

Item.js

- render function

```
render() {
  return (
    <div className={styles.wrapper}>
      {this.props.item.get('message')}
    </div>
  )
}
```

APP WebView

Have you ever tried spam? (the meat product)

Would you ever let your parents pick out a partner
for you?

If a loved one was to serenade you, what song
would you most like them to sing?

Do you think laughing at someone elses misfortune
is wrong?

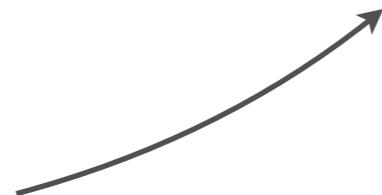
If you had to describe yourself as an animal, what
would it be?

If you had to describe yourself as a car, what would

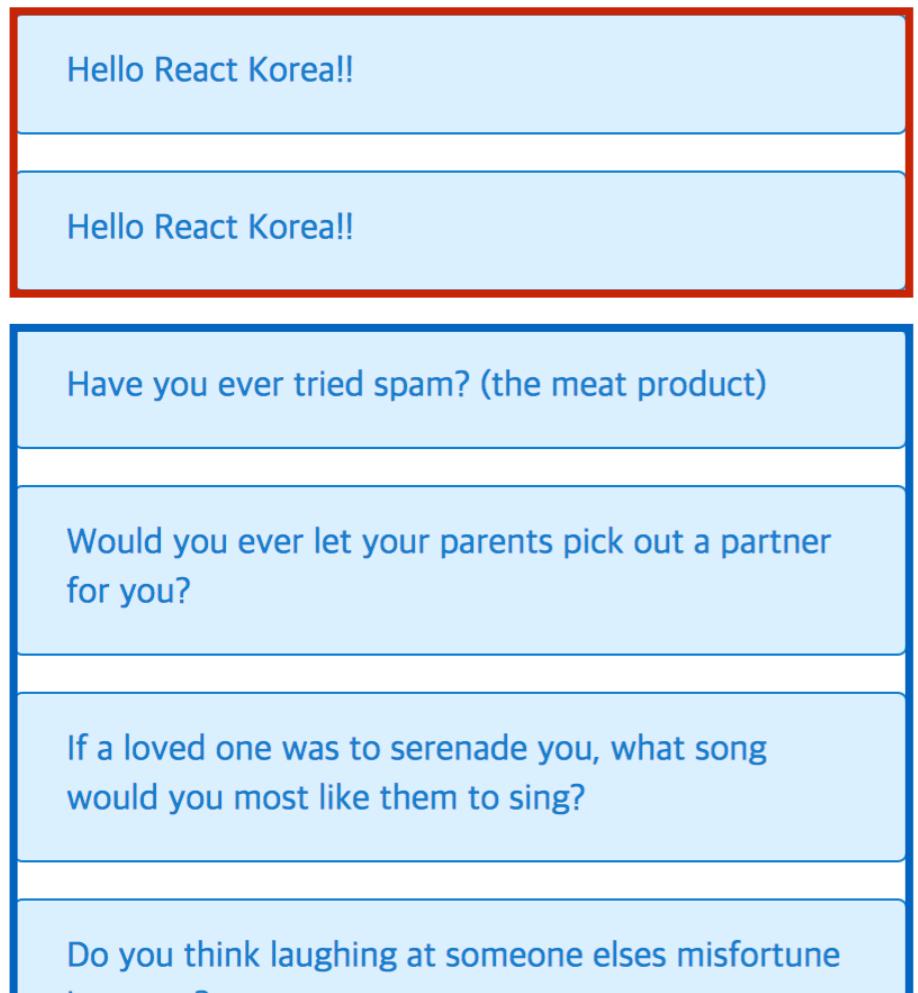
Add new message

클릭하면 새로운 Item 추가

약 800개의 Item이 render 됨



Wasted time



- 붉은 직사각형 내부는 새로 추가된 아이템
- 푸른 직사각형 내부는 기존에 있던 아이템
- 붉은 직사각형 내부만 새롭게 render
- 그 외의 동작은 wasted time으로 간주

Wasted time

- Inclusive Time

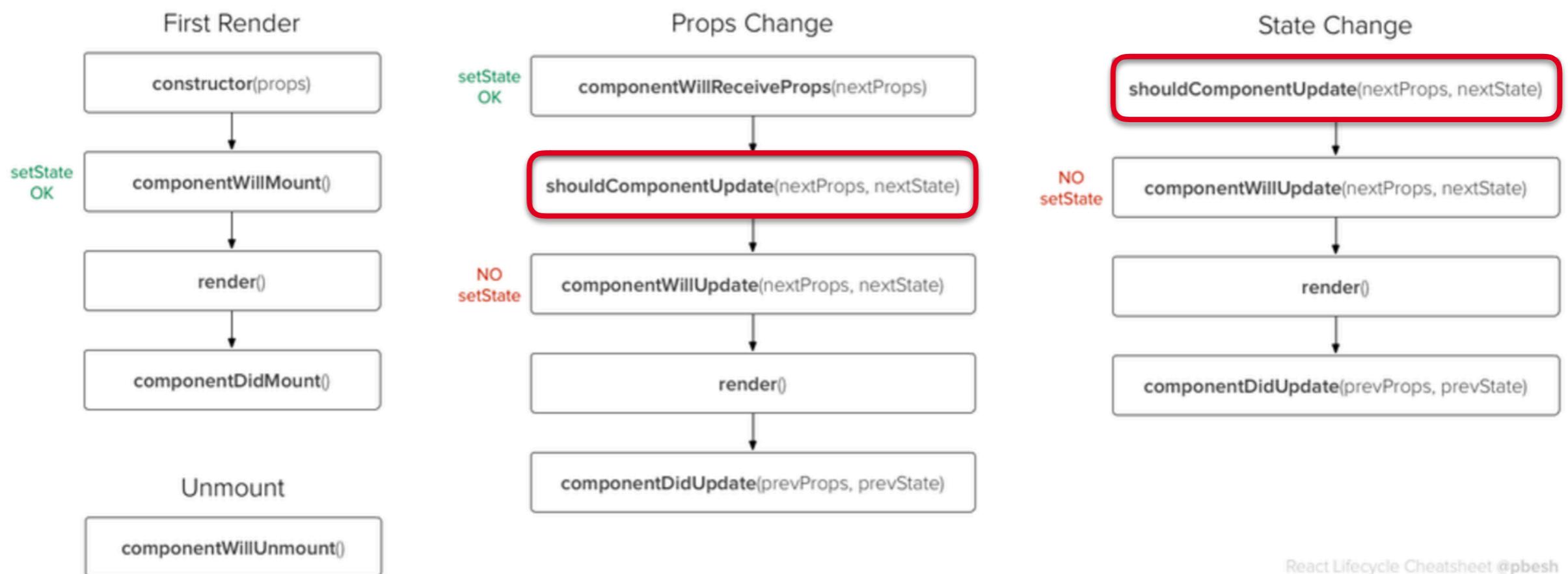
(index)	Owner > Component	Inclusive...	Instance cou...	Render count
0	"Connect(App)"	14.54	1	1
1	"Connect(App) > App"	14.52	1	1
2	"App > Item"	5.19	795	795

- Wasted Time

(index)	Owner > Comp...	Inclusive wa...	Instance cou...	Render count
0	"App > Item"	5.16	793	793

- 새롭게 추가된 두개 외 나머지 793개가 또다시 render

리액트 라이프사이클 API



shouldComponentUpdate(…)

- 성능 개선에 아주 중요한 역할을 하는 API
- 컴포넌트가 새로운 state, props를 받을 때 리렌더 유무 결정
- true를 반환하면 리렌더를 함
- false를 반환하면 리렌더를 안함

Item.js + shouldComponentUpdate

- shouldComponentUpdate function

```
shouldComponentUpdate(nextProps, nextState) {  
  return this.props.item.get('message') !== nextProps.item.get('message')  
}
```

- message가 바뀌지 않으면 다시 render할 필요가 없다

Item.js + shouldComponentUpdate

- Inclusive Time

(index)	Owner > Component	Inclusive render time	Instance count	Render count
0	"Connect(App)"	8.25	1	1
1	"Connect(App) > ..."	8.23	1	1
2	"App > Item"	0.02	2	2

- Wasted Time

null...

- 새롭게 추가된 두 개의 메세지만 render하는 것을 확인
- shouldComponentUpdate만 잘 써도 성능이 많이 올라간다

Item.js + PureComponent

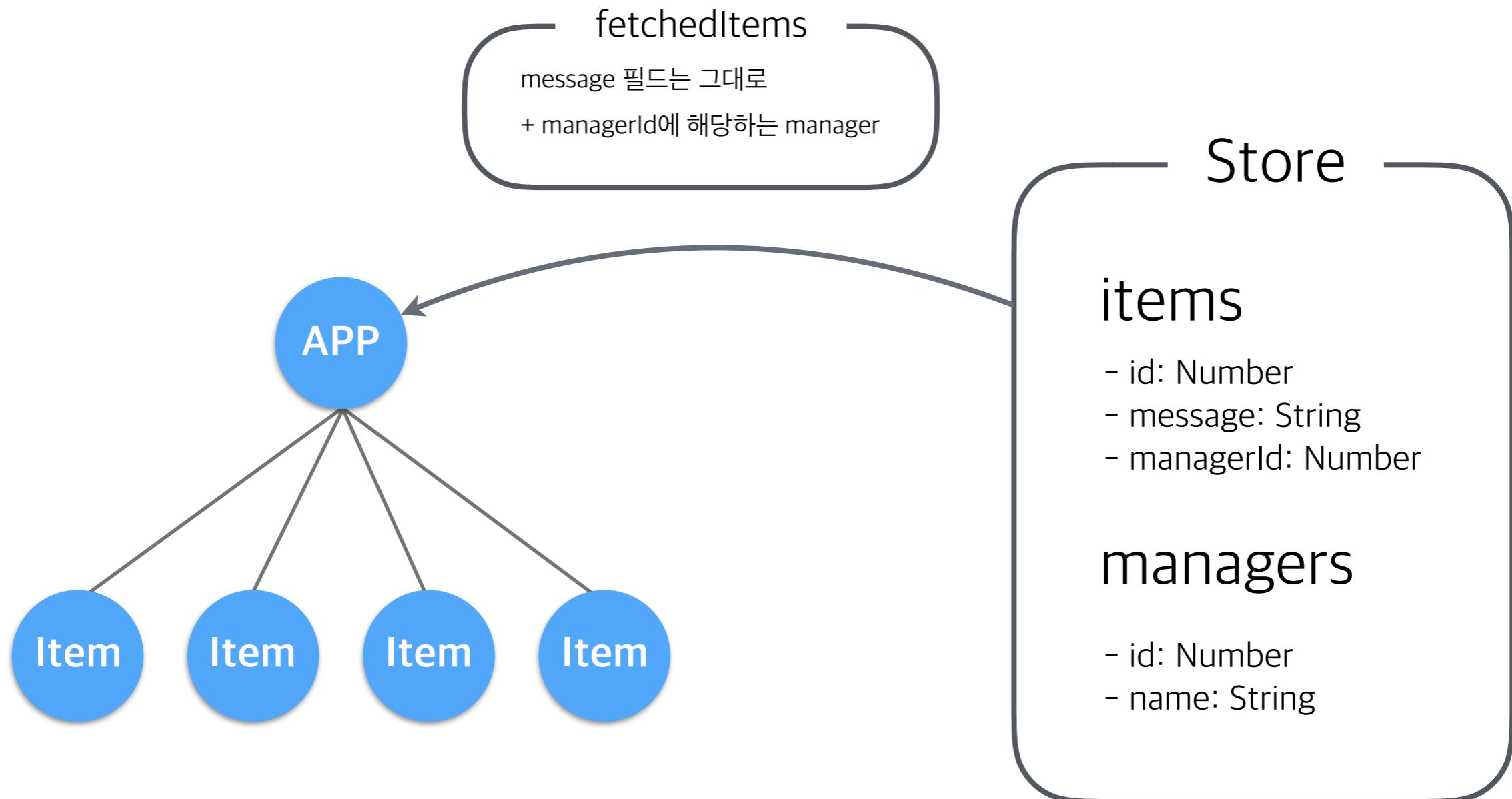
- 하지만 앱 규모가 커지면 비교해야 할게 많아진다
- 실수로 한개라도 빼먹으면 버그가 발생하는게 현실...
- 그래서 사용하는 PureComponent!!

```
class Item extends React.PureComponent {  
  render() {  
    return (  
      <div className={styles.wrapper}>  
        {this.props.message.get('message')}  
      </div>  
    )  
  }  
}
```

Item.js + PureComponent

- 다음 state와 props를 현재 state와 props와 자동으로 비교
- 단!! shallowCompare이기 때문에 구현할때 잘 생각
- 앱을 다음과 같이 바꿔보면 어떨까?

Advanced APP의 구조



Advanced App.js

- renderMessage function

```
renderMessage(message) {  
  return (  
    <Item  
      onClick={message => this.handleItemClick(message)}  
      key={message.get('id')}  
      message={message} />  
  )  
}
```

- handleItemClick function

```
handleItemClick(message) {  
  this.props.dispatch({  
    type: AT.DELETE_MESSAGE,  
    id: message.get('id'),  
  })  
}
```

Store selector

- selector function

```
const selector = state => {
  const { items, managers } = state
  return items.map(item => (
    item.set('manager', managers.find(manager => (
      manager.get('id') === item.get('managerId')))
  )));
}
```

Advanced APP WebView

Have you ever tried spam? (the meat product)	Fritz Fraley
Would you ever let your parents pick out a partner for you?	Winona Nedd
If a loved one was to serenade you, what song would you most like them to sing?	Andree Vaught
Do you think laughing at someone else's misfortune is wrong?	Margarete Kneip
If you had to describe yourself as an animal, what would it be?	Margareta Cuenca
If you had to describe yourself as a car, what would it be?	Monet Flack

Add new message

아이템 작성자의 이름을 표시

아이템을 클릭하면 아이템이 삭제

Trouble shooting

- 위와 같이 변경한 Advanced APP에서 PureComponent가 동작하지 않음
- Advanced APP에서는 두 가지 큰 실수가 존재
- 첫 번째는 event 함수에서
- 두 번째는 selector의 Higher order function(map) 에서
(Immutable 방식을 사용했을 때)

PureComponent 는
Shallow Compare!!!

Trouble shooting - event

- ES6의 Arrow function은 자동으로 this에 바인딩
- 매번 새로운 event function을 생성
- 아래 테스트 코드를 유심히 살펴보자

```
const e = () => console.log('Hello React Korea')
console.log(e.bind(this) == e.bind(this)) // false
```

- 따라서 event function은 constructor에서 bind하는게 좋음

Trouble shooting - HOF(map)

- selector에서는 아래와 같이 HOF를 사용함

```
const selector = state => {
  const { items, managers } = state
  return items.map(item => (
    item.set('manager', managers.find(manager => (
      manager.get('id') === item.get('managerId'))))
  )))
}
```

- store가 변경될때마다 매번 새로운 fetchedItem을 생성
- 설령 내용이 전혀 바뀌지 않았음에도 불구하고…

Trouble shooting - HOF(map)

- PureComponent에서는 Item의 주솟값이 바뀌면 리렌더
- shallowCompare를 하기 때문에!!
- 따라서 fetchedItem을 만들때 memoization등을 통해 내용이 똑같으면 항상 같은 주솟값을 가질 수 있게 만들어줘야 함

구현은 여러분의 뜻 😊

Any Questions?