

<http://github.com/thanhtungdp/react-native-redux-i18n>

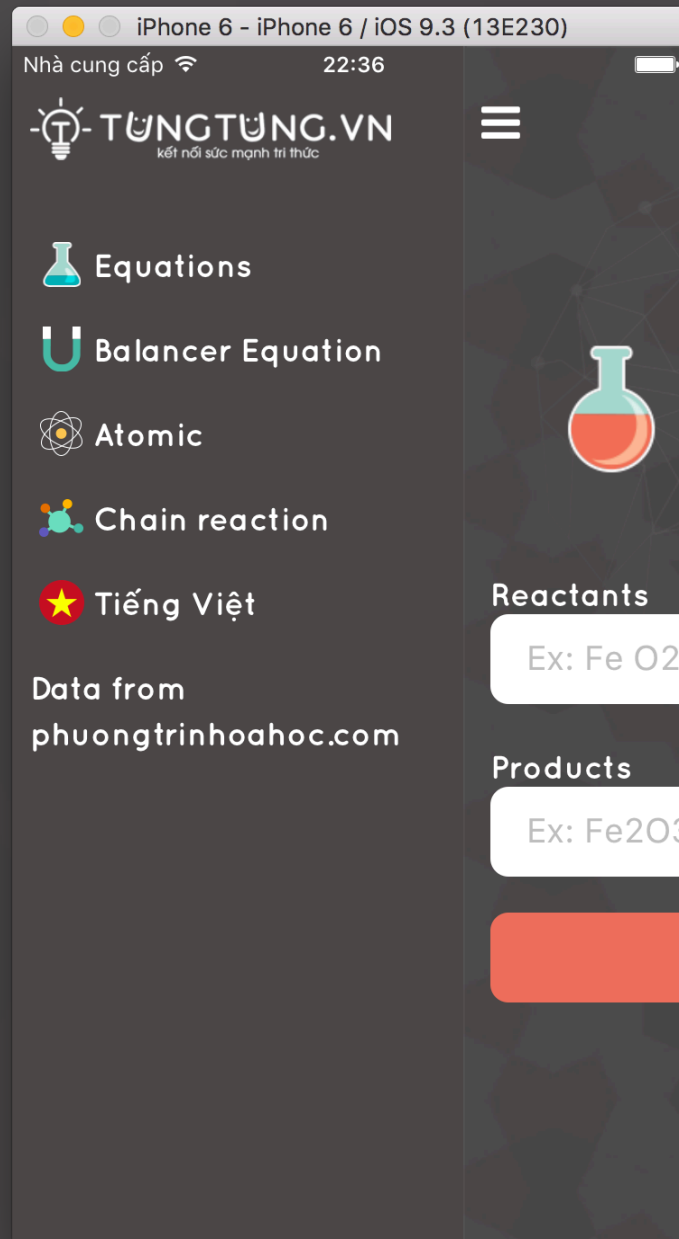
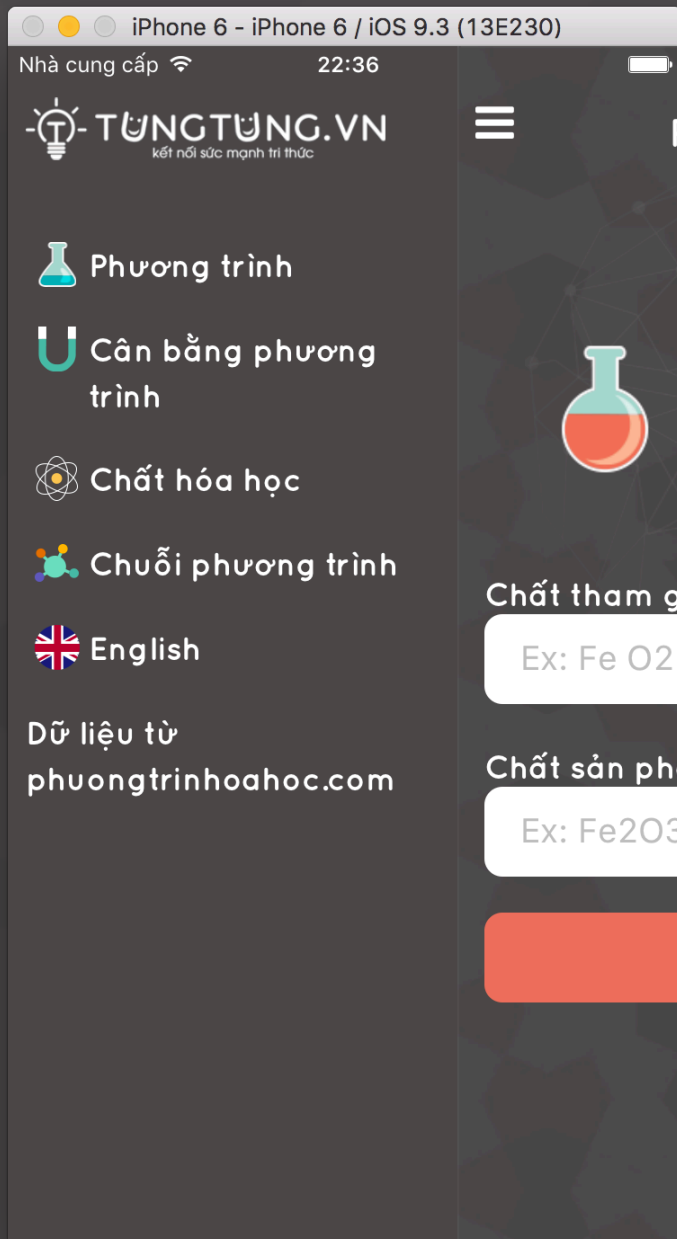


# React Native + Redux

Xây dựng ứng dụng đa ngôn ngữ



Trình bày: Phan Thanh Tùng  
Co-founder [tungtung.vn](http://tungtung.vn) – [vndev.co](http://vndev.co)



# Demo app

[hoahoc.tungtung.vn/ios](http://hoahoc.tungtung.vn/ios)

[hoahoc.tungtung.vn/android](http://hoahoc.tungtung.vn/android)



iPhone 6 - iPhone 6 / iOS 9.3 (13E230)  
Nhà cung cấp 22:36

< Results +

**$\text{Fe} + 2\text{HCl} = \text{FeCl}_2 + \text{H}_2$**

Report Details

Process: Fe for a little metal on the bottom of a test tube, add 1-2 ml of acid tube.

Categories: The reaction of oxidation-reduction, Alternative chemical reactions

Phenomenon: Dissolved metals, and colorless gas foam flying off

Other: Clohicdric acid is a strong acid, potentially react with hydrogen in the metal precedes chemical activity ranges

**$3\text{Cl}_2 + 2\text{Fe} = 2\text{FeCl}_3$**

Report Details

**$3\text{Fe} + 2\text{O}_2 = \text{Fe}_3\text{O}_4$**

Report Details

iPhone 6 - iPhone 6 / iOS 9.3 (13E230)  
Nhà cung cấp 22:36

< Kết quả +

**$\text{Fe} + 2\text{HCl} = \text{FeCl}_2 + \text{H}_2$**

Góp ý Chi tiết

Loại phản ứng: Phản ứng oxi-hoá khử, Phản ứng thế

Hiện tượng: Kim loại bị hoà tan, đồng thời có bọt khí không màu bay ra

Thông tin thêm: Axit clohicđric là axit mạnh, có khả năng tác dụng với kim loại đứng trước hidro trong dãy hoạt động hoá học

**$3\text{Cl}_2 + 2\text{Fe} = 2\text{FeCl}_3$**

Góp ý Chi tiết

**$3\text{Fe} + 2\text{O}_2 = \text{Fe}_3\text{O}_4$**

Góp ý Chi tiết

**$\text{Fe} + \text{CuSO}_4 = \text{Cu} + \text{FeSO}_4$**

# Demo app

[hoahoc.tungtung.vn/ios](http://hoahoc.tungtung.vn/ios)

[hoahoc.tungtung.vn/android](http://hoahoc.tungtung.vn/android)



# Một số các package hỗ trợ cho việc chuyển đổi ngôn ngữ

- React native i18n
  - <https://github.com/AlexanderZaytsev/react-native-i18n>
- React Native Localization
  - <https://github.com/stefalda/ReactNativeLocalization>



# Cài đặt react-native-i18n

- npm install --save react-native-i18n
- rnpm link react-native-i18n



# Sử dụng cơ bản

```
import React, {Component} from 'react';
import {View, Text} from 'react-native';
import {I18n} from 'react-native-i18n';

I18n.locale = 'vi';
I18n.translations = {
  en: {
    hello: 'Hello world'
  },
  vi: {
    hello: 'Xin chào'
  }
}

class AppLang extends Component {
  render() {
    return (
      <View style={{flex:1, paddingTop: 100}}>
        <Text>{I18n.t('hello')}</Text>
      </View>
    )
  }
}
```



# Thay đổi language trong Component

```
export default class AppLang extends Component {
  constructor() {
    super(...arguments);
    I18n.locale = 'vi';
    I18n.translations = {...}
    this.state = {
      I18n
    }
  }

  changeLocale(locale = 'vi') {
    let i18nClone = this.state.I18n;
    i18nClone.locale = locale;
    this.setState({I18n: i18nClone});
  }

  render() {
    return (
      <View style={styles.view_container}>
        <Text style={styles.hello}>{this.state.I18n.t('hello')}</Text>
        <Text onPress={this.changeLocale.bind(this, 'en')}
          style={styles.button}>
          Change to english
        </Text>
      </View>
    )
  }
}
```

```
const styles = {
  view_container: {
    flex:1, justifyContent: 'center', alignItems: 'center'
  },
  hello: {
    fontSize: 20, marginBottom: 20
  },
  button: {
    width: 150, borderWidth: 1,
    borderColor: '#333333', padding: 10
  }
};
```

Xin chào

Change to english

Hello world

Change to english

# Vấn đề

- Khi muốn sử dụng language, hoặc thay đổi language ở Child component chúng ta phải truyền thông qua props.

Điều này sẽ khá phức tạp khi ứng dụng của bạn có khá nhiều child component lồng nhau. tức là mỗi khi bạn muốn sử dụng bộ ngôn ngữ bạn phải truyền qua props

```
import React from 'react';
export function wrapTranslate({I18n, changeLocale}, component) {
  return React.cloneElement(component, {
    I18n: I18n,
    changeLocale: changeLocale
  });
}
```

```
render() {
  return (
    <View style={styles.view_container}>
      {wrapTranslate({I18n: this.state.I18n, changeLocale: this.changeLocale.bind(this)},
        <ChildComponent />
      )}
    </View>
  )
}
```

```
class ChildComponent extends Component {
  render() {
    console.log(this.props.changeLocale);
    return (
      <View style={styles.view_container}>
        <Text style={styles.hello}>{this.props.I18n.t('hello')}</Text>
        <Text onPress={() => this.props.changeLocale('en')}
          style={styles.button}>
          Change to english
        </Text>
        {wrapTranslate(this.props.I18n, <ChildComponentTwo />)}
      </View>
    );
  }
}
```





# Usage with redux

- Redux sẽ làm cho việc truyền props trở nên cực kỳ đơn giản thông qua việc xây dựng smart, dumb component. Dưới đây là ví dụ.

```
export default class AppRedux extends Component {
  render() {
    return (
      <Root>
        <View style={styles.view_container}>
          <Text style={styles.hello}>
            {translateText('hello')}
          </Text>

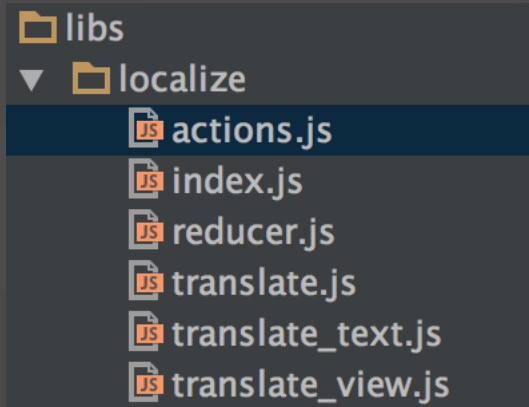
          <Text style={styles.hello}>
            {translateText('hello_name',{name: 'Tung'})}
          </Text>

          {wrapTranslate(
            {
              placeholder: {
                value: 'hello_name',
                params: {name: 'Tung'}
              }
            },
            <TextInput style={styles.button}/>
          )}
        </View>
      </Root>
    )
  }
}
```

```
export default {
  en: {
    hello: 'Hello world',
    hello_name: 'Hello {{name}}'
  },
  vi: {
    hello: 'Xin chào thế giới',
    hello_name: 'Xin chào {{name}}'
  },
}
```



# Xây dựng thư viện translateText, wrapTranslate



Cấu trúc thư mục

```
// index.js

import I18n from 'react-native-i18n';

export {
  SET_LOCALE,
  LOAD_TRANSLATIONS,
  setLocale,
  loadTranslations,
} from './actions';

export {I18n};

export { default as i18nReducer } from './reducer';

// Smart component
export {wrapTranslate, translateText} from './translate';
```

# Xây dựng Action

```
//actions.js
```

```
export const LOAD_TRANSLATIONS = 'loadTranslation';
export const loadTranslations = (translations) => {
  return {
    type: LOAD_TRANSLATIONS,
    translations
  }
};
```

```
export const SET_LOCALE = 'set locale';
export const setLocale = (locale, saveToStorage = false) => {
  return {
    type: SET_LOCALE,
    locale
  }
};
```

# Xây dựng Reducer

```
// reducer.js

import { LOAD_TRANSLATIONS, SET_LOCALE } from './actions';

export default (state = {}, action) => {
  switch (action.type) {
    case LOAD_TRANSLATIONS:
      return {
        ...state,
        translations: action.translations
      };
      break;
    case SET_LOCALE:
      return {
        ...state,
        locale: action.locale
      };
      break;
    default:
      return state;
  }
};
```

# TranslateText

## Smart component

Sử dụng cho việc hiển thị text

```
// translate_text.js
import React, {Component} from 'react';
import {connect} from 'react-redux';
import {Text} from 'react-native';
import I18n from 'react-native-i18n';

export class TranslateText extends Component{
  render(){
    I18n.locale = this.props.i18n.locale;
    I18n.fallbacks = true;
    I18n.translations = this.props.i18n.translations;

    return (
      <Text style={this.props.style}>
        {I18n.t(this.props.value, this.props.params)}
      </Text>
    )
  }
}

TranslateText.defaultProps = {
  params: {}
}

const mapStateToProps = (state, ownProps) => {
  return {
    i18n: state.i18n,
    ...ownProps
  }
}

export default connect(mapStateToProps)(TranslateText);
```

# TranslateView

## Smart component

Sử dụng cho việc truyền các language thông qua props. Ví dụ như khi sử dụng `<TextInput placeholder="">`

```
// translate_view.js

import React, {Component} from 'react';
import {connect} from 'react-redux';
import {View} from 'react-native';
import I18n from 'react-native-i18n';

export class TranslateView extends Component {
  render() {
    I18n.fallbacks = true;
    I18n.locale = this.props.i18n.locale;
    I18n.translations = this.props.i18n.translations;

    let props = {};

    Object.keys(this.props.values).map((key) => {
      if (typeof this.props.values[key] !== 'object') {
        props[key] = I18n.t(this.props.values[key]);
      } else {
        props[key] = I18n.t(this.props.values[key].value, this.props.values[key].params);
      }
    });

    return (
      <View style={this.props.style}>
        {React.cloneElement(this.props.children, props)}
      </View>
    );
  }
}

TranslateView.defaultProps = {
  keys: '',
  values: ''
};

const mapStateToProps = (state, ownProps) => {
  return {
    i18n: state.i18n,
    ...ownProps
  };
};

export default connect(mapStateToProps)(TranslateView);
```

# Xuất thư viện

```
// translate.js

import React from 'react';
import Text from './translate_text';
import View from './translate_view';

export function translateText(keyValue, params = {}) {
  return <Text value={keyValue} params={params}/>
}

export function wrapTranslate(keyProps, Component, props) {
  return <View values={keyProps} {...props}>{Component}</View>
}
```

# Sử dụng với smart component

```
import React, {Component} from 'react';
import {View, Text, TextInput} from 'react-native';
import {bindActionCreators} from 'redux';
import {connect} from 'react-redux';
import {translateText, wrapTranslate, setLocale} from '../../libs/localize/index';

class ChangeLocaleTest extends Component {
  render() {
    return (
      <View style={styles.view_container}>
        <Text style={styles.hello}>Current language: {this.props.currentLocale}</Text>
        <Text style={styles.hello}>{translateText('hello')}</Text>
        <Text style={styles.hello}>{translateText('hello_name', {name: 'Tung'})}</Text>
        {wrapTranslate(
          {placeholder: {value: 'hello_name', params: {name: 'Tung'}}},
          <TextInput style={styles.button}/>
        )}
        <Text onPress={() => this.props.setLocale('en')}
          style={styles.button}>
          Change to english
        </Text>
      </View>
    )
  }
}

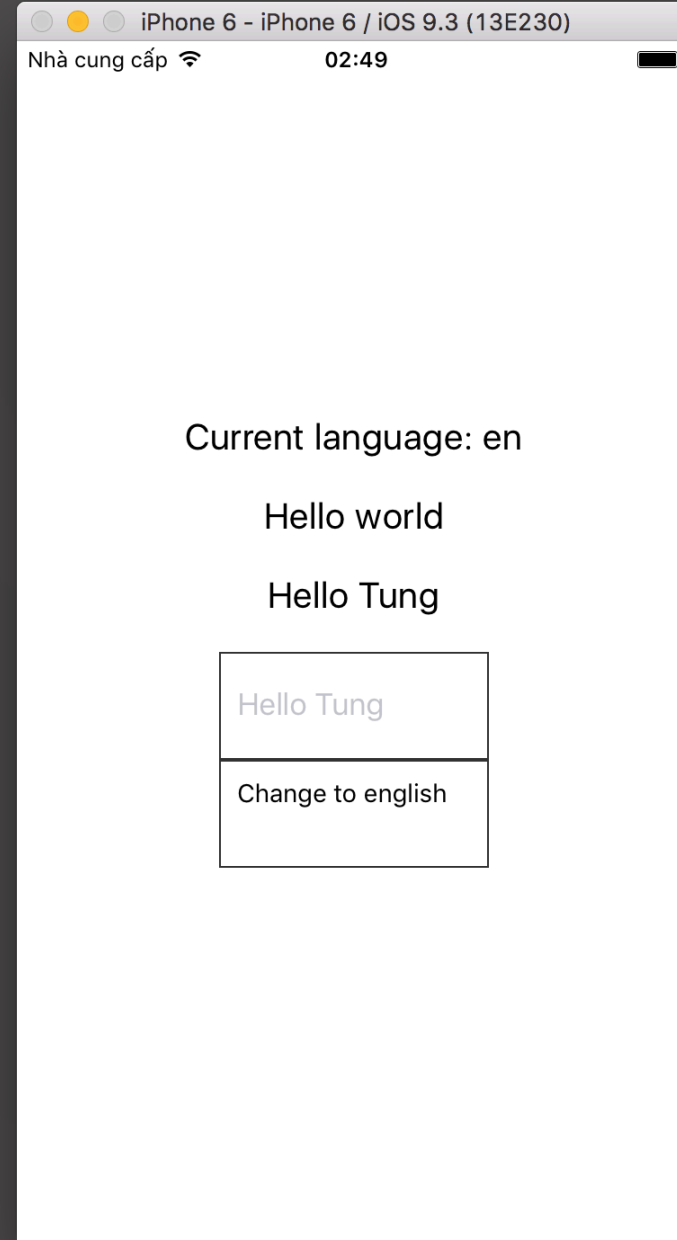
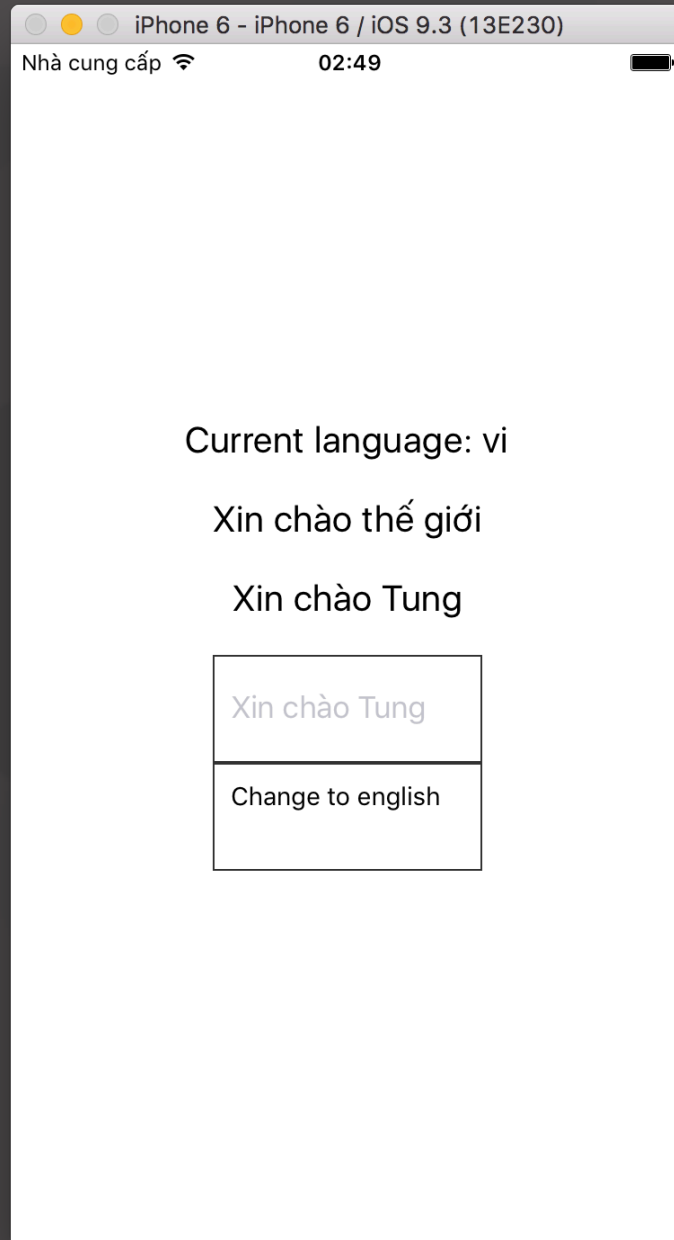
const mapStateToProps = (state) => {
  return {
    currentLocale: state.i18n.locale
  }
}

const mapDispatchToProps = (dispatch) => {
  return bindActionCreators({setLocale}, dispatch);
}

export default connect(mapStateToProps, mapDispatchToProps)(ChangeLocaleTest);
```



# Kết quả



<http://github.com/thanhtungdp/react-native-redux-i18n>



# React Native + Redux

Xây dựng ứng dụng đa ngôn ngữ



Trình bày: Phan Thanh Tùng  
Co-founder [tungtung.vn](http://tungtung.vn) – [vndev.co](http://vndev.co)