

# React Guideline Syntax

## Table of contents

<b>Text Editor</b>	<b>2</b>
<b>Class Name</b>	<b>2</b>
<b>Create New File</b>	<b>2</b>
<b>Import</b>	<b>3</b>
<b>Variable</b>	<b>4</b>
<b>Function</b>	<b>5</b>
<b>API Service</b>	<b>5</b>
<b>Pass Props Variable</b>	<b>6</b>
<b>Conditional Operator</b>	<b>7</b>
<b>Render Value</b>	<b>7</b>

## Text Editor

1. Indent using spaces = 2

## Class Name

1. Components
  - 1.1. เริ่มต้นด้วยตัวใหญ่
  - 1.2. ใช้ CamelCase
  - 1.3. การตั้งชื่อควรทำให้ Components ที่เกี่ยวข้องกันอยู่ใกล้กัน
  - 1.4. Ex.
    - 1.4.1. TableBody
    - 1.4.2. TableContent
    - 1.4.3. TableHeader
2. Containers
  - 2.1. เริ่มต้นด้วยตัวใหญ่
  - 2.2. ใช้ CamelCase
  - 2.3. ชื่อตามด้วย Container
  - 2.4. Ex.
    - 2.4.1. UserManagementContainer

## Create New File

1. การสร้าง File React ใหม่

```
import React from 'react'

export class UserManagementContainer extends React.Component {
  render() {
    return (
      <div/>
    )
  }
}
```

# Import

1. การเรียง Import
  - 1.1. Plugin modules
  - 1.2. Components
  - 1.3. Style
  - 1.4. Theme helpers
  - 1.5. Containers
  - 1.6. Api service
  - 1.7. Context
  - 1.8. Script helpers
2. Import แบบ Value
  - 2.1. ให้ขึ้นบรรทัดใหม่ทุก Variable
  - 2.2. ตัวสุดท้ายไม่ต้องมี Comma

```
import React from 'react'
import {
  connect
} from 'react-redux'
import {
  Segment
} from 'semantic-ui-react'
import {
  HeaderText,
  HeadBar,
  RecentUserTable,
  ModalDialog,
  ViewCard,
  ViewProfile,
  Profile,
  Pagination
} from 'components'
import {
  cssVariable
} from 'assets'
import {
  userService
```

```

    } from 'apiService'
import {
    ORDER_TABLE,
    SORT_TABLE_USER,
    isValidResponse,
    redirect
} from 'helpers'

```

## Variable

1. การตั้งชื่อ
  - 1.1. ใช้ CamelCase
  - 1.2. ขึ้นต้นด้วยตัวเล็ก
  - 1.3. ตั้งชื่อที่อ่านแล้วเข้าใจง่าย
    - 1.3.1. เกี่ยวกับ Element ที่จะเอาไปใช้
    - 1.3.2. เกี่ยวกับ Value ของ Variable
    - 1.3.3. Ex. Boolean
      - 1.3.3.1. ขึ้นต้นด้วย is
      - 1.3.3.2. Ex. isOpenModal
    - 1.3.4. Ex. Array
      - 1.3.4.1. ตามด้วย List
      - 1.3.4.2. Ex. userList
  - 1.4. Value ที่เป็น String ให้ใช้ Single Quote ( ' )
  - 1.5. Destructuring และ Object ให้ขึ้นบรรทัดใหม่ทุก Variable

```

const {
  user
} = this.props
const {
  isOpenEditAccount,
  oldPasswordStatus
} = this.state
const params = {
  'Password': oldPassword
}

```

## Function

1. ใช้ CamelCase
2. ขึ้นต้นด้วยตัวเล็ก
3. สร้างเป็น Group ของ Function
4. คำแรกเป็นกริยาให้ขึ้นต้นด้วย on
  - 4.1. Ex.
    - 4.1.1. onClickButton
    - 4.1.2. onChangeInput
    - 4.1.3. onOpenModal
    - 4.1.4. handleLogin
    - 4.1.5. toggleSideMenu

```
topbar = {
  onClickLogo: () => {
    redirect({
      pathname: ROUTE_PATH.DASHBOARD.LINK,
      query: {
        refresh: true
      }
    })
  },
  onClickLogout: () => {
    removeLocalStorage(LOCAL_STORAGE.USER_AUTH)
    redirect('/')
  },
  onOpenEditAccountModal: () => {
    this.setState({
      isOpenEditAccount: true
    })
  },
},
}
```

## API Service

1. สร้าง File service ที่สอดคล้องหน้าตาม backend

- 1.1. Ex.
  - 1.1.1. userService
  - 1.1.2. staffService
- 2. ใช้ async กับ await ในการเรียก api แบบ synchronous

```
onClickSaveChangePassword: async () => {
  const {
    oldPassword,
    newPassword,
    oldPasswordStatus
  } = this.state
  if (oldPasswordStatus === 2 && newPassword) {
    const params = {
      'OldPassword': oldPassword,
      'NewPassword': newPassword
    }
    const res = await staffService.changePassword(params)
    if (isValidResponse(res)) {
      this.setState({
        oldPasswordStatus: 1
      })
      this.changePassword.onCloseModal()
      alert.success(MESSAGE.CHANGE_PASSWORD_COMPLETE,
        imageIconMainLayout['success.svg'])
    }
  }
}
```

## Pass Props Variable

- 1. หลังชื่อ Component ให้ส่ง props เรียงจาก
  - 1.1. Boolean
  - 1.2. String
  - 1.3. Variable
    - 1.3.1. ให้ขึ้นบรรทัดใหม่ทุก Variable
    - 1.3.2. เรียง Variable จาก
      - 1.3.2.1. Value
      - 1.3.2.2. Function

#### 1.4. ถ้ามี Props value ให้ Tag ปิด ขึ้นบรรทัดใหม่

```
<HeadBar showText headtext='User Statistics'  
  onlineList={onlineUserStaticList}  
  headerActions={this.userStatistics.headBarActions()}  
>  
</>
```

## Conditional Operator

1. สำหรับ Render Element หรือ Function ให้ขึ้นบรรทัดใหม่ทั้งหมด

```
{  
  isEmpty ?  
    this.header.renderHeaderBar()  
    :  
    <div/>  
}
```

2. สำหรับ Select Value ให้เขียนบรรทัดเดียวกันและติดกับปีกกา

```
{contentId === 0 ? ROUTE_PATH.CREATE_CONTENT : ROUTE_PATH.EDIT_CONTENT}
```

## Render Value

1. การแสดง Value หรือ Function ให้ติดกับปีกกา

```
{content}
```

2. การแสดง Value และ String ให้ใช้ Back single code ( ` ` )

```
{`Title: ${titleText}`}
```