

Yesco HR WEB개발 표준 정의서

1. 개요

프로젝트의 개발 시 개발생산성 향상 및 운영의 효율화를 위해 반드시 준수되어야 하는 준수 사항을 정의함

- * 제시된 표준은 철저히 준수한다.
- * 새로운 정의가 추가로 필요한 경우는 본 자료를 작성한 담당자와 협의하여 해당 내용을 정의하고, 본 표준서에 수록한 후 프로젝트팀 전체에 공지한다.
- * 모든 화면은 최대 응답속도 7초 이내를 목표로 Design 되어야 한다.
- * 표준을 준수하지 못할 경우에는 PM 및 인수책임자의 허가를 받아야 하며, 그럴 경우에도 해당 시스템 내에는 일정한 표준을 정의하여 적용시켜야 한다.(품질계획서 기준을 따른다)
- * 본 표준안은 최소한의 필수사항을 지키도록 유도하고 해서는 안되는 것들을 지적하는 내용이며, 개개인의 다양한 능력과 개념들로 본 표준을 확장시켜 시스템에 적용시키도록 한다.

2. Coding Convention

아래의 Airbnb, SAPUI5 스타일을 준수한다.

2.1 Airbnb JavaScript 스타일 가이드

- <https://github.com/tipjs/javascript-style-guide>

2.2 SAPUI5 Coding Guidelines

- <https://blog.sap-press.com/sapui5-coding-guidelines>

3. 명명규칙

3.1 폴더 구조

```
/webapp <ROOT>
  /common <공통 Script>
  /control <Custom UI5 Object>
  /controller <SAPUI5 Controller>
  /css <Page Style>
  /fragment <SAPUI5 Fragment>
  /libs <External Library>
  /view <SAPUI5 View>
```

3.2 파일

다음과 같은 파일 네이밍 규칙을 준수한다.

- Controller [파스칼 표기법]

업무^의미^상태.controller.js
조합 예: MedicalApprovalDetail.controller.js <의료비 신청 상세>

- View [파스칼 표기법]

업무^의미^상태.controller.js
조합 예: MedicalApprovalDetail.view.xml <의료비 신청 상세>

- Fragment [파스칼 표기법]

업무^의미^상태.fragment.js
조합 예: CommonEmployeeList.fragment.xml <공통 사원 검색>

- Control [파스칼 표기법]

기능.js
조합 예: Placeholder.js

- CSS [카멜 표기법]

```
업무.css  
조합 예: medical.css
```

3.3 변수

- * 모든 이름은 영어로 작성한다.
- * 컬렉션의 이름은 반드시 복수형으로 사용한다.
- * 변수의 이름은 대소문자를 혼용할 수 있지만 반드시 소문자로 시작한다.
- * 상수를 표현하는 이름은 반드시 모두 대문자로 지정하되 '_' 사용하여 단어들을 구분한다.
- * 축약형 이름의 사용은 피한다.
- * Boolean 변수 이름은 부정적인(거짓인) 이름을 사용하지 않는다.

- **String**

```
const sId = "approvalTable";
```

- **Object**

```
const oTable = sap.ui.getCore().byId("approvalTable");  
const oPayload = {  
  name: "Kim",  
  age: 31,  
};
```

- **jQuery Object**

```
const $Table = $('#approvalTable');
```

- **Int**

```
const iCount = 0;
```

- **Map associative array**

```
const mDatas = [  
  {  
    idx: 1,  
    title: "First subject",  
    content: "It's First content",  
  },  
  {  
    idx: 2,  
    title: "Second subject",  
    content: "It's Second content",  
  },  
];
```

- **Array**

```
const aMenus = [  
  "first menu",  
  "second menu",  
  "third menu",  
];
```

- **Date**

```
const dToday = new Date();  
const dToday = moment();
```

- **Float**

```
const fDecimal = 1.25;
```

- **Boolean**

```
const bIsNew = true;
```

- **RegExp**

```
const rPattern = /\w+/;
```

- **Function**

```
const fnSuccess = (res) => {  
  // do something  
  console.log(res);  
}
```

- **Variant types**

```
let vDate;  
vDate = "2021-09-01";  
...  
vDate = moment();
```

4. 주석

- * 프로그램의 이해를 도와 향후 유지보수가 용이하게 한다.
- * 전체적인 프로그램 설명은 프로그램 앞부분에 위치시킨다.
- * 각 Function 앞에 Function의 Parameter, Return Value의 의미와 Function의 간략한 설명을 기술한다.
- * 해당 프로그램의 핵심적인 중요Logic에 대해서는 반드시 그에 따른 주석을 추가한다.
- * JSDoc 표준을 준수한다.
- * JSDoc에 의해 처리될 주석은 `/**`으로 시작해 `*/`으로 종결된다. 각 라인마다 `/*`로 시작한다.
- * 사양서에 기술할 필요가 없는 주석은 자유롭게 기술한다. (`/* */`, `//`사용)

- [JSDoc Guidelines](#)

```
/**
 * Compute average
 *
 * @author hhsung <hhsung@lsitc.com>
 * @version 1.0.0
 * @this AppHandler
 * @throws 인자값이 없으면 null 리턴
 * @param {number} a 국어 점수
 * @param {number} b 수학 점수
 * @returns {number} Average
 * @example
 *     computeAverage(80, 90);
 */
function computeAverage(a, b) {
    return (a + b) / 2;
}
```

- 상수

```
/**
 * @constant {number} 페이지 별 콘텐츠 개수
 */
const PER_PAGE_COUNT = 10;

/**
 * @constant {string} blue 색상코드
 */
const YESCO_BLUE = "#0072c6";
```