La documentation JavaScript : JSDoc

Achref El Mouelhi

Plan

1. Introduction
2. Syntaxe
3. `JSDoc` : installation et génération de documentation

Documentation JavaScript

Définition

Il s'agit d'un document `PDF`, `HTML` ou autre qui permet de décrire les scripts JavaScript.

Documentation JavaScript

Définition

Il s'agit d'un document `PDF`, `HTML` ou autre qui permet de décrire les scripts JavaScript.

Ce n'est pas un rapport à rédiger.

Documentation JavaScript

Définition

Il s'agit d'un document `PDF`, `HTML` ou autre qui permet de décrire les scripts JavaScript.

Ce n'est pas un rapport à rédiger.

C'est plutôt un rapport à générer à partir de notre code-source.

Documentation JavaScript

Pourquoi ? !

- Avoir une idée sur la structure de l'application
- Permettre de bien comprendre le fonctionnement
- Facilité la maintenance
- Éviter de lire plusieurs (milliers voire plus) de lignes de code pour comprendre le fonctionnement de l'application

Documentation JavaScript

Pourquoi ? !

- Avoir une idée sur la structure de l'application
- Permettre de bien comprendre le fonctionnement
- Facilité la maintenance
- Éviter de lire plusieurs (milliers voire plus) de lignes de code pour comprendre le fonctionnement de l'application

C'est encore plus important pour un langage faiblement typé comme JavaScript

Documentation JavaScript

Principe

- Bien commenter le code (respecter certaines règles)
- Utiliser un générateur de documentation pour extraire ses commentaires et créer la documentation

Documentation JavaScript

Plusieurs générateurs de documentations pour PHP

- `JSDoc`
- `ESDoc`
- `YUIDoc`
- `Slate`
- `Doxygen`

Documentation JavaScript

3 types de commentaires

- Commentaire mono-ligne

```
// commentaire sur une seule ligne
```

- Commentaire multi-ligne

```
/* commentaire sur
plusieurs lignes */
```

- Commentaire pour annotation et documentation

```
/** commentaire pour
 documentation */
```

Documentation JavaScript

Règles

- Un commentaire de documentation concerne un élément de notre code source
  - une classe
  - un attribut de classe
  - une variable
  - une fonction ou une méthode
  - ...
- Pas de ligne vide entre un commentaire de documentation et l'élément commenté
- Un commentaire de documentation peut avoir un (ou même plusieurs) attributs (appelés aussi balises)

Documentation JavaScript

Les attributs `JSDoc`

- `JSDoc` possède son propre format pour les attributs `@attribut`
- Ce format est inspiré par le format `JavaDoc`

Documentation JavaScript

Attributs

- `@author` : permet d'indiquer l'auteur (ou les auteurs)

```
/** @author Wick */
```

- `@method` : permet de décrire une fonction/méthode

```
/** @method Cette fonction permet de ... */
```

- `@deprecated` : indique qu'une fonction/méthode est dépréciée

```
/** @deprecated */
```

- `@see` : invite l'utilisateur à consulter un autre élément JavaScript

```
/** @see method somme() */
```

Documentation JavaScript

Attributs

- `@version` : permet de définir la version d'un fichier/classe...

```
/** @version 3.2beta */
```

- `@class` : permet de marquer la déclaration d'une classe

```
/** @class Voiture */
```

Documentation JavaScript

Attributs

- `@return(s)` : permet de décrire la valeur de retour d'une fonction/méthode.

```
/** @return {integer} renvoit la somme de deux
 parametres. */
```

- `@param` : permet de lister et décrire les paramètres d'une fonction/méthode

```
/**
 * @param {integer} a premier parametre
 */
```

Documentation JavaScript

Autres @attributs

- @exception
- @type, @constant
- @this
- @override
- @exports, @extends, @throws, @requires...
- @constructor
- ...

Documentation JavaScript

Téléchargement

- Télécharger JSDoc `npm install -g jsdoc`
- Générer la documentation avec la commande `jsdoc chemin/vers/vos/scripts`

Documentation JavaScript

Exemple

- Créer un répertoire jsDoc
- Dans ce répertoire, créer les deux fichiers suivants :

Documentation JavaScript

Premier fichier

```
/**
* @param  a {integer} premier parametre
* @param b {integer} deuxieme parametre
* @function computeSomme qui calcule la somme
* @return {integer} la somme des deux parametres a et b
*/
function computeSomme(a,b) {
    var  somme =  a + b;
    return somme;
}
/**
* @param  a {integer} premier parametre
* @param b {integer} deuxieme parametre
* @function computeProduit qui calcule le produit
* @return {integer} le produit des deux parametres a et b
*/
function computeProduit(a,b) {
    var produit =  a * b;
    return produit;
}
```

Documentation JavaScript

Deuxième fichier

```
/**
* @param {integer} a premier parametre
* @param {integer} b deuxieme parametre
* @method findMaximum qui cherche et retourne le max
* @return {integer} le max entre deux parametres a et b
* @deprecated @see findMax
* @version 1
* @author Achref
*/
function findMaximum(a,b) {
        if (a<b)
                return b;
        else
                return a;
}
```

Documentation JavaScript

La suite du fichier précédent

```
/**
* @param {integer} a premier parametre
* @param {integer} b deuxieme parametre
* @method findMax qui cherche et retourne le max
* @return {integer} le max entre deux parametres a et b
* @version 2
* @author El Mouelhi
*/
function findMax(a,b) {
        if (a<b)
                return b;
        return a;
}
```