

Documenter son code

Documentation des fichiers du projet

Chaque fichier **doit** avoir sa documentation au début de celui-ci

Exemple

```
/**
 * @file Nom du fichier
 * @brief Résumé du rôle du module/classe
 * @brief On peut continuer le résumé sur plusieurs lignes
 * @author Auteur du fichier
 * @version v1.0
 * @class Nom de la classe (si POO)
 * @date 01/01/1900
 */
```

■

Documentation des fonctions ou méthodes

- La documentation doit être écrite dans les fichiers `.h`
- Chaque fonction/méthode **doit** avoir sa documentation avant son prototype

Cas général

```
/**
 * @fn type nomFonction(type parametre1, type parametre2);
```

```
* @brief Résumé de la fonction
* @param type parametre1 : Le rôle du paramètre 1
* @param type parametre2 : Le rôle du paramètre 2
* @return type : le rôle de la valeur de retour
*/

type nomFonction(type parametre1, type parametre2);
```

■

Exemple : méthode `string findClientName(int idClient);`

```
/**
 * @fn string findClientName(int idClient);
 * @brief trouve le nom du client en fonction de son id
 * @param int idClient : l'id du client à rechercher
 * @return string : nom du client
 */

string findClientName(int idClient);
```

■

Attribut des classes

- Chaque attribut doit être documenté ainsi

Cas général

```
type nom; //!< rôle de l'attribut
```

■

Exemple

```
QSqlDatabase _dbParking; //!< base de données
```

■

Balises supplémentaires

@code

Cette balise permet d'insérer un exemple de code dans la documentation

```
/**
 * @code
 * #include <iostream>
 *
 * int main()
 *{
 * //Your code
 *
 * return 0;
 *}
 * @endcode
 **/
```

■

Doxyfile

On peut ensuite générer la documentation grâce à [doxygen](#)

Si doxygen n'est pas installé sur le PC, exécuter la commande suivante

```
apt install doxygen graphviz doxygen-gui
```

■

Ensuite, on peut générer la documentation

- Soit, en suivant le doxywizard dans le dossier du projet

```
doxywizard
```

-
- Soit en lançant directement la commande (il faut alors avoir un doxyfile cf-ci-après)

```
doxygen Doxyfile
```

■

Exemple de doxyfile C

Exemple de doxyfile cpp

Générer automatiquement la documentation avec le pipeline Gitlab CI/CD

- Ajouter le stage ci-dessous à votre fichier .gitlab-ci.yml

```
stages:
  - deploy

pages:
  stage: deploy
  script:
    - doxygen Doxyfile
    - mv html/ public/
  artifacts:
    paths:
      - public
```

■

ToDo

- La documentation générée n'est accessible que depuis notre réseau local

Doc gitlab et doxygen

Revision #2

Created 28 May 2024 11:34:06 by Nicolas

Updated 31 January 2025 23:10:10 by Nicolas