

CHARTRE DU DEVELOPPEUR

BONNEFETE

Créé avec
OfficeSuite

BONNE FETE

Charte du développeur

Sommaire

- Qualité du Code1
- Avoir un code propre, lisible et maintenable.....2
- Respecter les conventions de codage.....3
- Utiliser des noms de variables, de fonctions et de classes explicites et significatifs.....4
- Éviter les doublons de code et favoriser la réutilisation du code existant.5
- Documenter le code avec des commentaires clairs et concis6
- Test et validation7
- Ecrire des tests unitaires exhaustifs pour chaque fonctionnalité développée8
- Effectuer des tests de validation pour s'assurer du bon fonctionnement global de l'application.9
- S'assurer que les tests sont exécutés régulièrement et que les erreurs sont corrigées rapidement.....10
- Sécurité11
- Appliquer les meilleures pratiques de sécurité lors de la conception et de la mise en œuvre de l'application.12
- Éviter les vulnérabilités courantes, telles que les injections SQL et les failles de sécurité XSS.....13
- Protéger les données sensibles et respecter la confidentialité des utilisateurs.....14
- Performances15
- Optimiser le code pour améliorer les performances de l'application.16
- Éviter les requêtes excessives à la base de données et les boucles inefficaces.17
- Gestion de versions et collaboration18
- Utiliser un système de contrôle de versions pour gérer le code source19
- Travailler sur des branches distinctes pour chaque fonctionnalité et fusionner les modifications de manière appropriée.....20
- Apprentissage continu21
- Suivre les évolutions des technologies et des bonnes pratiques de développement.....22
- Partager les connaissances et les découvertes avec les membres de l'équipe.23

Qualité du Code Avoir un code propre, lisible et maintenable

En effet le code doit être aéré et facilement lisible afin de permettre à toute l'équipe de s'y retrouver, ainsi il faudra respecter l'utilisation de fonctions et la hiérarchie de dossier mis en place pour le projet. Le code doit avoir une indentation parfaite.

Respecter les conventions de codage

Les conventions de codage à respecter sont les conventions de nommage expliquer ci-dessous, gérer correctement les erreurs et les exceptions en utilisant des blocs try-catch appropriés.

Éviter de masquer les erreurs en utilisant des blocs catch vides ou en ignorant les exceptions.

Enregistrer les erreurs et les exceptions de manière appropriée pour faciliter le débogage.

Utiliser des constantes pour les valeurs qui ne devraient pas être modifiées.

Priorisé le développement en programmation orienté objet pour le langage back

Utiliser des noms de variables, de fonctions et de classes explicites et significatifs

Utiliser des noms de variables, de fonctions et de classes significatifs et descriptifs, celles-ci devrait-être en anglais. Il vous faudra aussi utiliser la convention de nommage camelCase pour les variables et les fonctions (ex : maVariable, maFonction). Puis utiliser la convention de nommage PascalCase pour les noms de classes (ex : MaClasse). De même éviter les noms de variables génériques tels que "var1" ou "temp". Enfin éviter les abréviations obscures et privilégier la clarté.

Éviter les doublons de code et favoriser la réutilisation du code existant.

Éviter les fonctions ou les classes surchargées qui effectuent trop de tâches différentes.

Diviser le code en fonctions ou en classes logiques et cohérentes afin de limiter la répétition de code. Mise en place de librairie ou package possible afin de permettre la réutilisation de code dans divers projets.

Documenter le code avec des commentaires clairs et concis

Il faut commenter le code avec une phrase simple et concise afin d'expliquer les morceaux de code un peu plus complexe ou les noms de variables qui sont du langage métier du projet. A contrario ne pas laisser de code commenté (code mort), inutile dans le projet.

Utiliser des commentaires ou des en-têtes de fonction pour décrire la fonctionnalité, les paramètres et la valeur de retour.

Sécurité

Tenez compte des bonnes pratiques de sécurité lors du développement de vos applications.

Des lors que le projet commence une réflexion sur la sécurité, surtout sur la sécurité des données stocké doit-être fait. Ainsi un mot de passe conséquent doit-être mis en place pour la base de données.

Éviter les vulnérabilités courantes, telles que les injections SQL et les failles de sécurité XSS.

Pour éviter les injections SQL il nous faut respecter certaines règles dans le développement de notre application en utilisant des bibliothèques qui bloque ce genre d'attaque (ex : PDO)

Protéger les données sensibles et respecter la confidentialité des utilisateurs.

A l'aide de notre DPO, nous devons toujours vérifier la protections des données sensibles et respecter la confidentialité des utilisateurs. Ainsi les données personnelles non utile ne doivent pas être-stocker sauf si utile au fonctionnement de l'application.

Performances

Optimiser le code pour améliorer les performances de l'application.

En plus d'être simple, le code doit-être optimal pour une plus grande performance de l'application.

Utiliser des techniques de mise en cache pour réduire la charge sur le serveur.

La mise en cache surtout pour les projets CMS (WordPress) est très importantes grâce a ce système le site fonctionnera plus rapidement lorsque la personne aura déjà navigué sur le site.

BONNEFETE

Gestion de versions et collaboration

Utiliser un système de contrôle de versions pour gérer le code source

Pour chaque feature (fonctionnalité), un push sur git sur la branche respective doit-être fait avec un commentaire au commit qui est assez clair pour savoir ce qui a été fait.

Un système de version doit-être mis en place pour les nouveaux projets de développement personnalisé sous forme de trois chiffres X.X.X, avec le premier en partant de la droite correspondant à des résolutions de problème (X.X.1), le deuxième chiffre pour de nouvelles fonctionnalité peu conséquentes (X.1.X) et enfin le dernier chiffre pour les changements conséquent de l'application.

Travailler sur des branches distinctes pour chaque fonctionnalité et fusionner les modifications de manière appropriée.

Lors d'un projet une personne est désigné chef de projet, c'est cette personne qui gère les merges afin de mieux gérer les conflits de code entre les développeurs. Les conflits sont résolus ensemble lors des commit ou d'une réunion le matin.