

Automatisation du développement

Points de la semaine

- Génération de données
- Linters & git hooks
- Utilisation de Vite

Génération de données



Pourquoi générer des données ?

- Rendre l'application plus réaliste
- Tester les limites de l'application
- Tester les performances, les erreurs, les comportements
- Indépendance vis-à-vis de la production

Pourquoi éviter « test » 15x

- Ce n'est pas réaliste
- Ne teste pas différents cas
- Les affichages ne sont pas réalistes

Génération de données

Peupler la base de données ?

- Récup la base de prod > à eviter
- Construire un jeu de données statique > *intéressant mais limité*
- Commande générant automatiquement des données > *l'idéal*

Datafixture / Seeders /

- outils pour peupler la base de donnée via une commande
- Symfony > Datafixtures
- Laravel > Seeders
- Commandes Custom si pas sur ces frameworks

Faker

fakerphp.github.io

- Librairie PHP
- Génère des données aléatoires
- Plusieurs langues
- Nombreux types de données

Exemple d'utilisation Faker

```
$faker = Faker\Factory::create('fr_FR');
echo $faker->name();
// "M. Jean Dupont"
```

Interlude

- Dans le repo des exercices, répertoire « cours 2 INTERLUDES »
 - Allez dans le dossier `faker`
 - Installez les dépendances avec Composer
 - Affichez un utilisateur aléatoire à chaque rafraîchissement de la page

Linters



Qu'est-ce qu'un linter ?

- Outil d'analyse de code
- Vérifie la syntaxe
- Vérifie les bonnes pratiques
- Déetecte les erreurs courantes

Pourquoi utiliser un linter ?

- Améliorer la qualité du code
- Favoriser la collaboration
- Faciliter la maintenabilité
- Améliorer la lisibilité

Exemple de mauvais code JS



```
function test() {  
    var a = 1; var b = 2  
    var c = a + b;  
    return c;  
}
```

```
function test() {  
    var a = 1; var b = 2  
    var c = a + b;  
    return c;  
}
```

2:1	error	Expected indentation of 2 spaces but found 4	indent
2:5	error	Unexpected var, use let or const instead	no-var
2:16	error	Unexpected var, use let or const instead	no-var
2:25	error	Missing semicolon	semi
3:1	error	Expected indentation of 2 spaces but found 4	indent
3:5	error	Unexpected var, use let or const instead	no-var
4:1	error	Expected indentation of 2 spaces but found 4	indent
7:1	warning	This line has a length of 208. Maximum allowed is 120	max-len
7:1	warning	Unexpected console statement	no-console
7:13	error	Unexpected string concatenation	prefer-template
7:209	error	Newline required at end of file but not found	eol-last
7:209	error	Missing semicolon	semi

✗ 12 problems (10 errors, 2 warnings)

Les linters PHP

PHP Code Sniffer



- Outil d'analyse PHP
- Vérifie la syntaxe
- Contrôle le respect des standards PSR-12
- Documentation PSR-12

PHPStan

- Outil d'analyse de code PHP
- Déetecte erreurs et incohérences
- Vérifie types, variables, retours, etc.
- Complémentaire à PHP Code Sniffer

Les linters JS

ESLint

- Outil d'analyse de code JS
- Vérifie syntaxe et bonnes pratiques
- Comparable à phpcs mais pour le JS

Interlude

- Toujours sur le répertoire « cours 2 INTERLUDES »
- Allez dans le dossier `linter`
- Installez les dépendances PHP et JS comme indiqué dans le README
- Exécutez les commandes des linters
- Corrigez les erreurs indiquées

Git Hooks

Git Hooks

- Automatisation d'actions lors d'événements Git
- Exécutés à chaque action clé (commit, push...)
- Partage faciles entre membres de l'équipe

Git Hooks

Exemples d'utilisation

- `pre-commit` : Vérifier le code avant le commit
- `post-pull` : Installer les dépendances après un pull
- `post-push` : Envoyer une notification après un push

Utilisation de Vite

Vite

- Outil de build JS performant
- Très simple à utiliser
- Compatible JS vanilla, SCSS, TypeScript, etc.

Vite – principe

- Prend les fichiers source d'un dossier
- Les transforme selon les besoins/modules chargés
- Les build ou les sert via un serveur de développement

Vite — Exemple de configuration simple

```
import { defineConfig } from 'vite'
export default defineConfig({
    root: './assets',
    base: '/build/',
    server: { port: 3000 },
    build: {
        manifest: true,
        outDir: './public',
    }
});
```

Interlude

- Toujours sur le répertoire « cours 2 INTERLUDES »
- Allez dans le dossier `vite`
- Installez les dépendances JS comme indiqué dans le README
- Démarrez le serveur comme indiqué et accédez à l'application
- Lisez le fichier de config `vite` pour le comprendre

À votre tour 💪