

A5.2.4 Étude d'une technologie, d'un composant, d'un outil ou d'une méthode

- C5.2.4.1 Se documenter à propos d'une technologie, d'un composant, d'un outil ou d'une méthode
- C5.2.4.2 Identifier le potentiel et les limites d'une technologie, d'un composant, d'un outil ou d'une méthode par rapport à un service à produire

PhpStorm :

Type -> IDE (Integrated Development Environment)

Editeur -> JetBrains

License-> Propriétaire

OS compatibles : Microsoft Windows, Mac OS X, Linux

Ecrit en Java.

ANTÉRIORITÉ:

Première version : PhpStorm 1 -> ~ 2009

	PREMIÈRE VERSION	DERNIÈRE VERSION
PHPSTORM 1	~	1.0.2 (8 Aout 2010)
PHPSTORM 2	~	2.1.5 (16 Septembre 2011)
PHPSTORM 3	12 Janvier 2012	3.0.3 (28 Février 2012)
PHPSTORM 4	24 Avril 2012	4.0.3 (13 Juillet 2012)
PHPSTORM 5	12 Septembre 2012	5.0.4 (2 Novembre 2012)
PHPSTORM 6	12 Mars 2013	6.0.4 (11 Mai 2016)
PHPSTORM 7	22 Octobre 2013	7.1.5 (11 Mai 2016)
PHPSTORM 8	16 Septembre 2014	8.0.4 (11 Mai 2016)
PHPSTORM 9	08 Juillet 2015	9.0.3 (11 Mai 2016)
PHPSTORM 10	02 Novembre 2015	10.0.4 (11 Mai 2016)



Versions n'étant plus maintenues.

PRESENTEMENT :

Dernière version : PhpStorm 2016.1 Première sous-version : 17 Mars 2016

Dernière sous-version : 2016.2.2 (24 Octobre 2016)

PRIX :

	Entreprise et Organisation	Utilisateur Individuel
Par an	199€ la 1 ^{ère} année 159€ la 2 ^{ème} année 119€ la 3 ^{ème} année	89€ la 1 ^{ère} année 71€ la 2 ^{ème} année 53€ la 3 ^{ème} année
Par mois	19,90€ par mois la 1 ^{ère} année <i>Si abonnement ininterrompu :</i> -20% la 2 ^{ème} année -40% la 3 ^{ème} année	8,90€ par mois la 1 ^{ère} année <i>Si abonnement ininterrompu :</i> -20% la 2 ^{ème} année -40% la 3 ^{ème} année

LES FEATURES (Caractéristiques) EN BREF :

- Assistance intelligente à la programmation :
 - Editeur de code PHP Intelligent :
 - Auto complétion
 - Coloration et formatage syntaxique
 - Vérification et prévention d'erreurs à la volée
 - Prise en charge de mélange des langages
 - Outils de refactoring automatisés.
 - Analyse du code :
 - Des 100èmes d'inspection durant la saisie
 - Contrôle des erreurs sur l'ensemble du projet
 - Suggestion de « Réparations rapides »
 - Test instantanée des modifications
 - Recherche et navigation simple :
 - Déplacements rapides dans le projet
 - Sauts aux méthodes, définitions, usages en un clic
 - Intègre PHPDoc, aide à sa génération
- Debugging, Testing et Profiling :
 - Debugging :
 - Préconfigurations de débogage
 - Nombreuses options de débogage avec Visual Debugger
 - Usage de breakpoints
 - Débogage d'application distantes
 - Débogage de page en sessions multiple simultanément.
 - Testing :
 - Integration de PHPUnit
 - Tests directement sur le dossier, fichier ou classe
 - PHPUnit montre la proportion du code impliqué dans le test.
 - Profiling :
 - Profilage de l'application avec Xdebug ou Zend Debugger
 - Edition de rapport agrégés propre à l'application
- Editeur HTML/CSS/JavaScript :

- Editeur HTML et CSS :
 - Supporte toutes les technologies à la pointe du web (Ex : SASS, SCSS, LESS, Jade templates, etc.)
 - Aperçu des changements instantané dans le navigateur
- Editeur JavaScript :
 - Editeur fourni avec l'IDE
 - Auto complétion, coloration et formatage syntaxique...
 - Support des annotations type JSDoc
 - Débogages et test unitaires
 - Support des Framework JavaScript
- Environnement de développement :
 - Système de contrôle de version (VCS – Version Control System) :
 - UI unifiée pour travailler avec différents VCS
 - Support de git, GitHub, Mercurial, etc.
 - Base de données (BDD) et SQL :
 - Connection direct à la BDD sur l'IDE
 - Outils d'édition des tables (Structure et données)
 - Schématisation et analyse de diagrammes UML
 - Intégration de SQL Editor.

PRISE DE NOTE GRAFIKART :

LA BASE

Apparence :

Editeur à des thèmes (obscur, clair)

Coloration syntaxique (plusieurs aussi) -> FICHER-> SETTINGS : Colors and Fonts

Des thèmes sont préinstallés, on peut en télécharger d'autres (Depot github [daylerees.github.io](https://github.com/daylerees)) et les déposer dans User->Nom->.WebIde80->config->colors. Création d'un fichier .icls et on copie le code.

On peut changer le font indépendamment de la couleur.

Toutes modifications peuvent être intégrées au thème ou faire l'objet d'un nouveau thème

Nombreux raccourcis clavier

Plusieurs parties :

Barre de menu

Barre de navigation (chemin du fichier actuel)

Barre à gauche contenant le projet

Status barre (About fichier actuellement en visu + les erreurs) Plus icône permettant d'accéder rapidement à certains outils

LA

VIEW dans la barre de menu permet de paramétrer l'apparence de l'IDE

FICHER -> SETTINGS pour accéder aux différents paramètres de l'éditeur IL Y EN A TRES BEAUCOUP (IDE oblige)

C'est pourquoi un petit moteur de recherche a été intégré pour filtrer les différentes options sur la barre de gauche

On peut séparer les vues : WINDOW->EDITOR TABS -> Split Vertically/Horizontally

Les raccourcis :

Modifier les raccourcis : FILE->SETTINGS , Keymap

Recherche [Nom du raccourcis] OU Appuyer sur icône « loupe » et taper raccourcis

On peut alors modifier le raccourcis, l'IDE indique si nouveau raccourcis déjà utilisé ou non et lequel

*Ouvrir rapidement un fichier NAVIGATE->FILES Par défaut Ctrl+Maj+N

*Fermer un onglet raccourcis : Editor Tabs->Close Mettre Ctrl w

*Accéder au panneau des options HELP->FIND ACTION... Par défaut Ctrl+Maj+A

*Reformater le code : Code->ReformatCode Par défaut Ctrl+Maj+L

*Naviguer dans les différentes méthodes d'une classe : NAVIGATE->FILE STRUCTURE Par défaut Ctrl+F12

*Naviguer globalement pour chercher un symbole, on cherche n'importe quoi : NAVIGATE->SYMBOL... Par défaut Ctrl+Alt+Maj+N

*Étendre la sélection : Edit->Selection Par défaut Ctrl+W

'Sélectionner une chaîne ou un ensemble et trouver toutes ses occurrences : Navigate->Previous/Next Occurrence Par défaut Ctrl+Alt+Bas et Ctrl+Alt+Haut ET DANS Find->Select All Occurrences Par défaut Ctrl+Alt+Maj+J

*Dupliquer des lignes : Editor Actions->Duplicate Line or Block

NECESSITE D'UNE GROSSE CONFIGURATION AU DEPART, Y PASSER DU TEMPS ! Essayer, agencer au mieux pour être à l'aise au clavier

Création de fichiers :

Side Barre du projet permet d'un clic droit sur un Folder (Dossier) -> New-> Toute une ensemble de fichiers possible (soit vierge soit des TEMPLATES) ou dossier.

On peut se servir du raccourci par défaut Alt+Origin (touche à droite de la touche Insert ou Ins) pour sortir la barre de navigation, naviguer, faire une action et la fermer quand on veut.

On peut également se servir de la navigation barre en haut de l'interface de l'IDE (L'afficher ou la faire disparaître : VIEW->NAVIGATION BAR) et naviguer dessus toujours avec le raccourci Alt+Origin

Naviguer est une chose encore faut-il pouvoir faire des actions :

L'action « New » raccourci par défaut : Alt+Insert | NB : Créer un nouveau fichier directement dans le même dossier que le fichier actuellement ouvert ne nécessite pas de navigation, directement Ctrl+N suffit

Avantage d'un IDE est qu'il propose un panel de TEMPLATES qui permettent de créer rapidement des fichiers spécifiques (ex : une classe php) Une fenêtre s'ouvre alors et demande les détails du template à générer (ex : nom de la classe, extension, namespace, etc.)

LES TEMPLATES : On peut personnaliser les templates générer : FILES->SETTINGS et on cherche avec le mot clef « template », ensuite sélectionner Editor->File and Code Templates Et toute une Atelier de création et de modification de templates et d'Includes s'offre à nous ! #[Joindre des screens grafikart]#

Au-delà des templates, ce sont directement certains fichiers ne changeant que très rarement ou très peu dans leur contenu que l'on peut créer très rapidement en les renseignant (en les créant dans

l'IDE) et ainsi également pouvoir les créer en faisant FILE->NEW ou Alt+Insert et ainsi choisir directement dans la liste)

Live templates :

Dans le cas des parties de codes récurrentes, très répétitive, on peut sauvegarder le bout de code sous la forme d'un template. Pour cela on surligne le code en question dans son fichier, on accède au panneau des actions Ctrl+Maj+A on cherche « save as live templates ». On voit alors sur le panneau des LIVE TEMPLATE qu'il y déjà de nombreux template déjà inclus dans PhpStorm.

On voit le dossier « user », dans lequel est pré renseigné <abbreviation> le bout de code que vous souhaitez enregistrer, vous pouvez le changer l'abréviation, modifier son contenu, on peut rendre les variables dynamiques en les mettant entre des « \$ ». On peut éditer ces mêmes variables pour leur donner une valeur par défaut.

➔ Tapez l'abréviation, taper sur TAB, le Template apparaît

Refactoring :

Sélection d'un bout de code récurrent que l'on veut isoler dans une fonction : REFACTOR->REFACTOR THIS... ou Ctrl+Alt+Maj+T et choisir Method... une fenêtre propose de paramétrer la fonction à créer. REFACTOR THIS propose plusieurs options comme le renommage, l'extraction de variables etc...

Dans le cas d'un renommage, une fenêtre apparaît avant d'opérer les changements souhaités montrant tous les fichiers impactés par la refonte laissant le choix entre continuer ou annuler.

TOUJOURS SE MEFIER AVEC L'AUTOMATISME, DES ERREURS PEUVENT ETRE CREEES.

REFACTOR THIS-> Pull member up, permet de remonter une classe dans une classe parente.

Automatisme, si méthode est privée, passe automatiquement en protected.

Maintenir touche contrôle permet de rendre les fonctions cliquables pour se rendre directement à leur définition

Ctrl+E permet d'aller au dernier fichier consulté, permet de rapides aller/retour entre fichiers

Placer le curseur sur nom de classe, REFACTOR THIS, Interface, permet de créer une interface pour la classe. Une fenêtre s'ouvre proposant un nom d'interface et les membres à renseigner.

Indentation : automatisé, modifiable dans les Settings, rechercher « Code Style », on y trouve tous les codes et les styles qu'il adoptent automatiquement 100% personnalisable.

On y trouve également tous les automatismes concernant la documentation, les arrangements, etc.

Sans y toucher, les pratiques adoptées sont normalement compatibles **PSR**.

Utiliser les suggestions matérialisées par une ampoule devant la ligne concernée lorsque qu'un soulignement en jaune (Warning) ou rouge (Alert) apparaît avec Alt+Entrer pour automatiquement générer la solution

CCL : Intérêt de l'IDE sur l'Editeur : très intelligent, capable de comprendre le code, capable de proposer des solutions à vos problèmes, capable de réorganiser et générer du code à notre place.
GROS GAIN DE TEMPS ET D'ORGANISATION

OUTILS AVANCES :

Composer :

Composer est directement installé sur php storm (Composer sert à la gestion des dépendances).

Panneau d'action->Composer et on peut initialiser composer pour le projet.

On peut télécharger directement le fichier composer.phar dans le projet. Automatiquement composer.json est ouvert pour modification.

Une fois fait on peut réeffectuer Panneau d'action->Composer et est proposé Add dependency... on peut alors rechercher un Package.

ATTENTION ! PHP Storm est capable d'exécuter du PHP en ligne mais pas dans le cadre de Composer, il faut un PHP en local. Pour cela, télécharger directement les fichiers binaires de PHP (cela permet de ne pas avoir à installer PHP) Placer tout le contenu du dossier dans un dossier de votre système. Puis dans l'IDE renseignez l'emplacement de votre php.exe.

Cela fait, on peut installer n'importe quel package dans notre application/projet.

S'il n'existe pas, le dossier vendor est alors créé à la racine du projet contenant toutes les librairies importées via composer.

Xdebug :

Lancer l'écoute de Xdebug : RUN->START LISTENING FOR PHP DEBUG CONNECTIONS

A chaque fois que l'on va recharger l'onglet écouté par Xdebug sur le navigateur, PhpStorm va réagir et s'arrêter sur les breakpoints placé devant les lignes de notre code et produire des rapports.

Permet de ne plus avoir à faire de « die(var_dump(\$variable)) », etc.

Comme le mapping entre fichiers distant et fichiers locaux peut ne pas être fait. On peut créer un nouveau serveur web PHP :

RUN->EDIT CONFIGURATION, Clic sur le « + » puis PHP Web Application(renseigner Nom, Server, Start Url, etc.).

On peut alors de la même manière créer un nouveau serveur. Lors de sa création cliquer sur « Use path mappings (select if the server or symlinks are used) » permet de mapper le fichier du projet en renseignant son chemin absolu sur le server.

On peut alors faire RUN->Debug '*Nom du serveur*' : Automatiquement il lance la page sur le navigateur et stop le script au premier point d'arrêt.

AVANTAGES : Accéder aux différentes variables pour constater leur valeur, les modifier, faire du pas à pas (step over) ou avancer jusqu'au prochain breakpoint.

PHPUnit :

- ➔ Permet de faire des tests unitaires directement dans l'éditeur sans beaucoup de configuration

Pour cela, créer la dépendance avec composer (chercher « phpunit/phpunit »),

Puis dans composer.json taper : « require-dev : { « phpunit/phpunit » : « [numero de version] » },

Configurer l'interpréteur Php dans PhpStorm : FILE->SETTING->PHP et on peut ainsi renseigner soit un interpréteur local ou distant.

Ensuite allez sur FILE->SETTING->PHPUnit et renseigner également l'interpréteur pour cet outil (ainsi que l'autoloader utilisé par l'application.

Mapper où se trouve le dossier du projet : TOOLS->DEPLOYEMENT. Et créer le nouvel environnement de déploiement.

LANCER LE FICHIER DE TEST : RUN->RUN-> '*Nom du fichier*'.

LANCER CE GENRE DE TEST est très efficace quand il s'agit de déboguer de gros fichiers applicatifs ou un Framework entier lorsqu'on y a amené des modifications.

Deploiement :

PhpStorm capable de gérer le FTP

Espace de déploiement à créer : TOOLS->DEPLOYEMENT->Configuration... On peut ainsi configurer un nouveau FTP

DEPLOYER: TOOLS->DEPLOYEMENT->Upload to... (On peut configurer la machine par défaut dans Configuration qui sera affiché dans la barre de menu

On peut choisir d'uploader le fichier ou le répertoire directement en faisant un clic droit dessus la barre du projet (Clic droit, Upload to...)

On peut également synchroniser le fichier à déployer. Dans le cas d'un travail en équipe, cela permet de pré visualiser les modifications faites par un autre sur le fichier à uploader qui pourrait interférer et synchroniser le fichier sur lequel on travail localement, le fichier distant, sans risque d'écrasement ou de perte de travail.

Dans le cas où l'on travaillerait, seul on peut activer TOOLS->DEPLOYEMENT->Automatic Upload qui upload les modifications du fichier dès qu'on y procède (ATTENTION ! Par défaut, PhpStorm y procède sans vérification ! Il écrasera automatiquement l'ancien fichier !) Mais encore une fois tout cela est configurable dans TOOLS->DEPLOYEMENT->OPTIONS.

Ex : si on configure une vérification du timestamp d'un fichier, opérer une modification sur un fichier ou une modification aurait déjà été opéré par un autre en ligne produira une fenêtre d'alerte proposant un Overwrite (réécrire en écrasant) OU de gérer la manière de Merge (fusionner) qui propose une interface permettant de conjuguer manuellement la modification des deux mêmes fichiers sur un seul.

On peut directement parcourir l'Hôte distant : TOOLS->DEPLOYEMENT->Browse Remote Host et ainsi procéder à de petites modifications directement sur l'hôte. Cette interface fonctionne exactement comme un client FTP

Versioning :

VCS->LOCAL HISTORY->Show History : Interne à l'IDE, ce dernier sauvegarde automatiquement les actions passées auxquelles on a pu procéder (Suppression, déplacement, etc.) et permet des retours en arrière en cas d'erreurs (remonte jusqu'à 7 jours !!).

Lancer le versioning sur un projet : VCS->ENABLE VERSION CONTROL IINTEGRATION puis renseignez l'outil de versioning souhaité. (Dans le cas de git, créé alors un Git repository dans votre projet).

Dans le cas de git : le dossier vendor de votre projet possède un fichier « .gitignore » dans lequel sont précisés tous les fichiers ignorés par le versioning (ne pas oublier de le renseigner).

Au démarrage, pour versionner les différents dossiers ou fichier il faut les sectionner dans la Barre du projet et faire VCS->Git->Add (Ou directement faire cette opération sur le dossier racine de votre projet pour versionner tout son contenu).

Une fois le versioning appliqué, on peut commit : VCS->COMMIT CHANGES.

On peut également gérer les branches VCS->GIT->BRANCHES

➔ VCS->VCS OPERATIONS POPUP... rassemble toutes les opérations que l'on peut effectuer

Historique de versioning également visible via cet outil

Gros avantage sur l'IDE c'est lors d'un commit, l'upload se fera également automatiquement sur le serveur, gros gain de temps.

File Watchers :

Système qui va inspecter la modification d'un fichier et qui va lancer automatiquement des commandes suite à ces modifications.

Pour le CSS par défaut, Emmet (Plugin améliorant grandement le workflow HTML et CSS) est installé. Pour le CSS, utiliser une règle de couleur fait apparaître un « Color picker » devant la ligne.

Si on ne travaille pas directement avec du CSS mais par exemple avec du SASS, l'éditeur peut directement compiler ce langage en CSS. Si un fichier scss est détecté il va suggérer son system de File Watcher pour le compiler. Pour utiliser un autre file watcher (Par exemple celui de Compass scss): Setting-> taper compass et activer. On peut alors relancer les fichier scss et ajouter le watcher compass sur l'ensemble du dossier contenant les .scss. Lorsque vous retapez du sass, son contenu est bien compilé automatiquement en css dans le fichier correspondant.

PhpStorm par rapport à des IDE plus génériques :

Par rapport à des IDE plus génériques (Netbeans, UltraEdit, Eclipse, Komodo Edit, etc.) qui sont gratuits et supportent de très nombreux langages, PhpStorm est spécifiquement dédié aux développeurs PHP et pensé pour les développeurs PHP. Il intègre de très nombreux outils conçus pour faciliter grandement diverse tâche et met à disposition des outils pour faciliter un travail collaboratif et/ou distant. Il propose beaucoup plus de fonctionnalités qui sont bien organisées, et de moyens de recherche qui font gagner énormément de temps lorsque l'on travaille sur un gros projet ou avec des nouvelles technologies webs. L'intégration de différents packages, plugins, etc. est très bien supporté.

AVANTAGES :

PhpStorm est véritablement 100% configurable !

Facilite grandement l'utilisation de Frameworks MVC (Symfony, Laravel, CakePHP, etc.)

Par rapport à un simple éditeur, cet IDE comprend votre code.

PHP tend à se « complexifier » ; il devient plus strict, utilise de plus en plus les namespaces, l'autoloading, composer, etc. et automatiser certain processus va venir alléger le workflow.

INCONVENIENT :

Interface complexe, pop up au démarrage

On peut être effrayé si l'on passe d'un simple éditeur de texte minimaliste comme SublimeText 2. PhpStorm la première fois peut faire peur comme se retrouver face à un cockpit d'avion. Les premiers pas avec le logiciel seront difficiles et les différentes configurations prendront du temps.

Si de petites modifications doivent être opérés sur un projet, un Editeur de texte, plus rapide et léger, tel que SublimeText peu suffire.

SOURCES :

<https://laracasts.com/series/how-to-be-awesome-in-phpstorm>

<https://www.jetbrains.com/phpstorm/>

<https://confluence.jetbrains.com/display/PhpStorm/Welcome>

<https://www.grafikart.fr/formations/phpstorm>

<http://buddyweb.fr/blog/pourquoi-jai-lache-sublime-text-pour-phpstorm/>

<https://en.wikipedia.org/wiki/PhpStorm>

