**CACHE :**

**/\*\***

**\* Gère le cache via APC.**

**\***

**\* Utilisation:**

**\***

**\* $cache = new APC('IDENTIFIER');**

**\* $cache->set('count', 1);**

**\* $count = $cache->get('count', 0); // 1**

**\* $cache->incremente('count', 2);**

**\* $count = $cache->get('count', 0); // 3**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Gère le cache via un système de fichiers.**

**\***

**\* Utilisation:**

**\***

**\* $cache = new File('IDENTIFIER', \_\_DIR\_\_ . '/../../cache');**

**\* $cache->set('key', 'value', 3600);**

**\* $key = $cache->get('key');**

**\* $cache->clear();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Gère le cache via Memcached.**

**\***

**\* Utilisation:**

**\***

**\* $cache = new Memcached('IDENTIFIER');**

**\* $cache->set('count', 1);**

**\* $count = $cache->get('count', 0); // 1**

**\* $cache->incremente('count', 2);**

**\* $count = $cache->get('count', 0); // 3**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**CLASSLOADER :**

**/\*\***

**\* Gestion du chargement automatique de classes.**

**\***

**\* Utilisation:**

**\***

**\* $loader = new Loader();**

**\* $loader->register();**

**\***

**\* $loader->addPrefix('My\_Application', \_\_DIR\_\_ . '/modules/My/Application/');**

**\* $loader->addNamespace('MyApplication', \_\_DIR\_\_ . '/modules/MyApplication/');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**CONFIG :**

**/\*\***

**\* Permet de stocker des données de configuration à partir de fichiers INI, XML ou un simple tableau PHP.**

**\***

**\* Utilisation:**

**\***

**\* $config = new Config('production');**

**\* $config->load(array(\_\_DIR\_\_ . 'config.php',**

**\* \_\_DIR\_\_ . 'config.xml'));**

**\***

**\* // Ajout d'un processeur**

**\* $processor = new Filter(new Replace(), array('regex' => '/{SCHEME}/', 'by' => 'https'));**

**\* $config->addProcessor($processor);**

**\***

**\* // Lecture**

**\* $value = $config->get('key'); // $config->key; // $config['key'];**

**\***

**\* // Ecriture**

**\* $config->set('key', 'new value'); // $config->key = 'new value'; // $config['key'] = 'new value';**

**\* $config->set('url', '{SCHEME}://www.example.com'); // https://www.example.com**

**\***

**\* // Utilisation de la notation pointée**

**\* $config->set('multi:key', 'value');**

**\***

**\* // Utilisation de la notation sous forme de tableau**

**\* $value = $config->get(array('multi', key'));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**DB :**

**/\*\*** **/\*\***

**\* Fait la passerelle avec une ligne d'une table dans une base de données.**

**\***

**\* Permet les opérations les plus courantes sur une table de base de données qui sont "INSERT", "UPDATE et "DELETE"**

**\* mais aussi de faire des opérations de type "SELECT" afin d'hydrater le modèle**

**\* ou de retourner un tableau de modèles correspondant à la recherche.**

**\***

**\* On accède à ses propriétées de manière dynamique (ex: $model->property), en interne s'il existe**

**\* une méthode "GETTER"/"SETTER" correspondante à la propriétée, elle sera executée à la place de la valeur brut.**

**\***

**\* Le nom de la table du modèle prend par défaut son nom sans namespace.**

**\* La clé primaire vaut "id" (il est important de noter que toutes les opérations se base sur la/les clé(s) primaire du modèle).**

**\* Si la clé primaire est de type "AUTO INCREMENT", alors définissez la propriétée "\_autoIncrement" à true. Cela permet**

**\* de mettre à jour la clé primaire après chaque opérations de type "INSERT".**

**\***

**\* Configuration:**

**\***

**\* class User extends ModelAbstract**

**\* {**

**\* public function \_\_construct(ContainerInterface $pManager = null, array $pData = array())**

**\* {**

**\* $this->\_tableName = 'tbl\_user';**

**\* $this->\_autoIncrement = true;**

**\* $this->\_primaryKey = 'id';**

**\***

**\* parent::\_\_construct($pManager, $pData);**

**\* }**

**\***

**\* // Définis les propriétées liées à la table**

**\* protected function defineColumns()**

**\* {**

**\* $this->id = null;**

**\* $this->name = null;**

**\* $this->firstname = null;**

**\* }**

**\***

**\* // Définis les relations etc...**

**\* protected function defineRelations()**

**\* {**

**\* $this->picture = null;**

**\* }**

**\***

**\* public function setName($pValue)**

**\* {**

**\* $this->setRaw('name', ucfirst($pValue));**

**\* }**

**\***

**\* public function setFirstname($pValue)**

**\* {**

**\* $this->setRaw('firstname', ucfirst($pValue));**

**\* }**

**\***

**\* // Astuce afin d'executer le chargement de la relation dynamiquement**

**\* public function getPicture($pForce = false)**

**\* {**

**\* if($pForce || null === $this->getRaw('picture'))**

**\* {**

**\* $this->loadPicture();**

**\* }**

**\***

**\* return $this->getRaw('picture');**

**\* }**

**\***

**\* // Etends le modèle (s'effectue après la requête)**

**\* public function loadPicture()**

**\* {**

**\* $picture = new Picture();**

**\* $picture = $picture->select()->where('idUser = ?', $this->id)->one();**

**\***

**\* $this->picture = $picture;**

**\* }**

**\***

**\* // Etends la requête**

**\* public function scopePicture(Select $pSelect)**

**\* {**

**\* $pSelect->join('`tbl\_picture`',**

**\* '`tbl\_picture`.`idUser` = `tbl\_user`.`id`',**

**\* null,**

**\* Picture::columns());**

**\* }**

**\* }**

**\***

**\* Utilisation:**

**\***

**\* $user = new User($DBManager);**

**\* $user->name = 'my name';**

**\* $user->firstname = 'my firstname';**

**\* $user->insert();**

**\* $id = $user->id;**

**\* $user = null;**

**\***

**\* $user = new User($DBManager);**

**\* $select = $user->select()->primary($id);**

**\***

**\* // Charge la relation picture**

**\* $select->load('picture');**

**\***

**\* // Ou directement dans la même requête**

**\* $select->scope('picture');**

**\***

**\* $user = $select->one();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Permet de construire une requête de type "DELETE" dynamiquement.**

**\***

**\* Utilisation:**

**\***

**\* $delete = new Delete('`tbl\_name`');**

**\* $delete->where('`type` = ?', 'my type')**

**\* ->render();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Permet de construire une requête de type "INSERT" dynamiquement.**

**\***

**\* Utilisation:**

**\***

**\* $insert = new Insert('`tbl\_name`');**

**\* $insert->ignore(true)**

**\* ->values(array('`name`' => 'my name',**

**\* '`firstname`' => 'my firstname'))**

**\* ->render();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Permet de construire une requête de type "SELECT" dynamiquement.**

**\***

**\* Utilisation:**

**\***

**\* $select = new Select('`tbl\_name`');**

**\* $select->where('`tbl\_name`.`name` = ?', 'my name')**

**\* ->join('`tbl\_name2`', function($pSQL)**

**\* {**

**\* $pSQL->on('`tbl\_name2`.id = ?', 6);**

**\* })**

**\* ->having(function($pSQL)**

**\* {**

**\* $pSQL->where('`tbl\_name`.`firstname` = :firstname', array('firstname' => 'my firstname'));**

**\* })**

**\* ->orderBy('`tbl\_name2`.id')**

**\* ->quantifier(Select::QUANTIFIER\_DISTINCT)**

**\* ->limit(1)**

**\* ->offset(6)**

**\* ->render();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Permet de construire une requête de type "UPDATE" dynamiquement.**

**\***

**\* Utilisation:**

**\***

**\* $update = new Update('`tbl\_name`');**

**\* $update->where('`type` = ?', 'my type')**

**\* ->set(array('`type`' => 'my new type'))**

**\* ->orderBy('RAND()', null)**

**\* ->limit(1)**

**\* ->render();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Fabrique de connexions aux bases de données.**

**\***

**\* Utilisation:**

**\***

**\* $DB = DBFactory::create(array('type' => 'pdo\_mysql',**

**\* 'host' => 'localhost',**

**\* 'username' => 'root',**

**\* 'password' => '',**

**\* 'dbname' => 'my\_database'));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Connexions à une base de données via l'extension PDO.**

**\***

**\* Utilisation:**

**\***

**\* $DB = new PDO('mysql:dbname=my\_database;host=localhost', 'root', '');**

**\* $DB->begin();**

**\* $DB->exec('INSERT INTO `tbl\_name` ('`name`') VALUES ("user");');**

**\* $DB->commit();**

**\***

**\* $statement = $DB->query('SELECT \* FROM `tbl\_name` where `name` = :name', array('name' => 'user'));**

**\* $row = $statement->fetch(SetAbstract::FETCH\_ASSOC);**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**DI :**

**/\*\*** **/\*\***

**\* Container d'injection de dépendance.**

**\***

**\* Objet qui est capable de créer des objets sur demande et de gérer le "câblage", ou l'injection des dépendances requises, pour les objets demandés.**

**\***

**\* Utilisation:**

**\***

**\* $container = new Container();**

**\***

**\* // Configuration via un fournisseur de services**

**\* $container->load(new DIProvider());**

**\***

**\* // Génère une valeur aléatoire à chaque appel**

**\* $container->set('param-1', function($pContainer)**

**\* {**

**\* return rand(0, 10);**

**\* });**

**\***

**\* // Génère le résultat une seule et unique fois**

**\* $container->singleton('param-2', function($pContainer)**

**\* {**

**\* return $pContainer->get('param-1');**

**\* });**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**DISPATCHER :**

**/\*\*** **/\*\***

**\* Gère le système événementiel, permet à tout objet de servir de cible d'événement.**

**\***

**\* Utilisation:**

**\***

**\* $dispatcher = new Dispatcher();**

**\***

**\* // Configuration via un souscripteur d'événements**

**\* $container->addSubscriber(new ListenerSubscriber());**

**\***

**\* $dispatcher->addListener(Event::COMPLETE, function(Event $e)**

**\* {**

**\* echo 'matched !';**

**\* });**

**\***

**\* $dispatcher->dispatch(new Event(Event::COMPLETE));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**FILTER :**

**/\*\*** **/\*\***

**\* Filtre un boolean.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Boolean();**

**\* $result = $filter->filter('true');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Filtre via une fonction de rappel.**

**\***

**\* Utilisation:**

**\***

**\* $callback = function($pValue)**

**\* {**

**\* return ucfirst($pValue);**

**\* }**

**\***

**\* $filter = new Callback();**

**\* $result = $filter->filter('username', array('options' => $callback));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Permet le chainage de filtres en un seul.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Chain();**

**\* $filter->addFilter(new Trim());**

**\* $filter->addFilter(new Escaper(), array('strategy' => 'html'));**

**\***

**\* $result = $filter->filter('My text to be filtered');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Recadre une image.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Crop();**

**\* $result = $filter->filter($\_FILES['file-uploaded'], array('wEnd' => '100', 'hEnd' => '100'));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Permet de formatter une date.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Date();**

**\* $result = $filter->filter('1970-01-01', array('input' => 'Y-m-d', 'output' => 'd/m/Y')); // 01/01/1970**

**\***

**\* // Avec formatage via la locale via l'extension "intl"**

**\* // Patterns disponible : http://userguide.icu-project.org/formatparse/datetime**

**\* $result = $filter->filter(new \DateTime(), array('locale' => 'fr\_FR',**

**\* 'ICU' => true,**

**\* 'output' => 'eeee dd LLLL yyyy')); // mercredi 07 août 2013**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Duplique un fichier et lui applique une succession de filtres.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Duplicate();**

**\* $filter->addFilter(new Rename(), array('name' => 'nameOfThumb', 'folder' => 'medias/upload/thumbs'));**

**\* $filter->addFilter(new Resize(), array('wMax' => '100', 'hMax' => '100');**

**\***

**\* $result = $filter->filter($\_FILES['file-uploaded']);**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Filtre un email.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Email();**

**\* $result = $filter->filter('use:r@ex;ample.com');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Redimentionne une image en s'assurant qu'elle prenne tout l'espace qui lui est attribué.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Enlarge();**

**\* $result = $filter->filter($\_FILES['file-uploaded'], array('wMin' => '100', 'wMin' => '100'));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Filtre d'échappement de toute sortie utilisateur.**

**\***

**\* Ceci est un portage de "Zend/Escaper/Escaper.php" du framework Zend.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Escaper('UTF-8');**

**\* $result = $filter->filter('My text to be filtered', array('strategy' => 'html'));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Filtrage statique.**

**\***

**\* Utilisation:**

**\***

**\* // Si on possede un container d'injection de dépendance, la recherche ce basera sur celui-ci en priorité**

**\* Filter::setContainer($container);**

**\***

**\* $result = Filter::filter('\Elixir\Filter\Email', 'use:r@ex;ample.com');**

**\* $result = Filter::filter('\Elixir\Filter\Date', '1970-01-01', array('input' => 'Y-m-d', 'output' => 'd/m/Y'));**

**\***

**\* @see http://fr.wikipedia.org/wiki/Fa%C3%A7ade\_(patron\_de\_conception)**

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Filtre un nombre à virgule flottante.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Float();**

**\* $result = $filter->filter(10.45F);**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Filtre un nombre entier.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Int();**

**\* $result = $filter->filter(10e);**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Supprime/replace tous les caractères sauf les lettres, chiffres et ".\_-()".**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Protect();**

**\* $result = $filter->filter('&éhe llo-@,world'); // hello-world**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Renomme un fichier uploadé.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Rename();**

**\* $result = $filter->filter($\_FILES['file-uploaded'], array('name' => 'nameOfFile', 'folder' => 'medias/upload'));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Recherche et remplace via expression régulière.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Replace();**

**\* $result = $filter->filter('hello {WORLD}', array('regex' => '/\{WORLD\}/i', 'by' => 'world'));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Redimentionne une image en s'assurant qu'elle rentre entièrement dans l'espace qui lui est attribué.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Resize();**

**\* $result = $filter->filter($\_FILES['file-uploaded'], array('wMax' => '100', 'hMax' => '100'));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Supprime les espaces (ou d'autres caractères) en début et fin de chaîne.**

**\***

**\* Utilisation:**

**\***

**\* $filter = new Trim();**

**\* $result = $filter->filter(' My text to be filtered '); // My text to be filtered**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Fabrique de formulaires.**

**\***

**\* Utilisation:**

**\***

**\* $form = FormFactory::createForm(array('name' => 'my-form',**

**\* 'attributes' => array('id' => 'formID'),**

**\* 'errorMessage' => 'An error has been detected',**

**\* 'items' => array('item-input' => array('type' => 'Elixir\Form\Field\Input',**

**\* 'required' => true,**

**\* 'validators' => array(new NotEmpty())))));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Création de formulaires, permet d'intéragir avec les formulaires HTML.**

**\***

**\* Un formulaire contient des éléments de formulaire de type "Input", "Select", "Checkox", etc... Mais peut aussi**

**\* aussi contenir d'autres sous-formulaires. Un élément de formulaire peut contenir des filtres et des validateurs afin**

**\* de gérer la cohérence des données soumises.**

**\***

**\* Un formulaire ne gère pas automatiquement son affichage HTML,**

**\* pour cela vous pouvez utiliser l'aide "\Elixir\Helper\Form".**

**\***

**\* Utilisation:**

**\***

**\* // Création basique**

**\* $form = new Form();**

**\* $form->setName('my-form');**

**\* $form->setAttribute('id', 'formID');**

**\* $form->setErrorMessage('An error has been detected');**

**\***

**\* $input = new Input('item-input');**

**\* $input->addValidator(new NotEmpty());**

**\* $input->setRequired(true);**

**\* $form->add($input);**

**\***

**\* // Création via la fabrique**

**\* $form = FormFactory::createForm(array('name' => 'my-form',**

**\* 'attributes' => array('id' => 'formID'),**

**\* 'errorMessage' => 'An error has been detected',**

**\* 'items' => array('item-input' => array('type' => 'Elixir\Form\Field\Input',**

**\* 'required' => true,**

**\* 'validators' => array(new NotEmpty())))));**

**\***

**\* if($\_POST)**

**\* {**

**\* if($form->submit($\_POST))**

**\* {**

**\* $values = $form->values();**

**\* }**

**\* else**

**\* {**

**\* $errors = $form->errors();**

**\* }**

**\* }**

**\* else**

**\* {**

**\* $form->bind(array('item-input' => 'hello'));**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**HTTP :**

**/\*\*** **/\*\***

**\* Crée un cookie HTTP.**

**\***

**\* Utilisation:**

**\***

**\* $cookie = new Cookie('cookie', 'my value', time() + 3600);**

**\* $cookie->send();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Contient les informations concernant le client.**

**\***

**\* Utilisation:**

**\***

**\* // Création basique**

**\* $request = new Request(array(),**

**\* $\_GET,**

**\* $\_POST,**

**\* $\_SESSION,**

**\* $\_COOKIE,**

**\* $\_FILES,**

**\* $\_SERVER,**

**\* $\_ENV);**

**\***

**\* // Via la fabrique**

**\* $request = RequestFactory::create();**

**\***

**\* $p1 = $request->get('p1');**

**\* $p2 = $request->getPost('p2');**

**\* $pathInfo = $request->getPathInfo();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Permet la création de requête correctement configurée de manière simplifié.**

**\***

**\* Comprend l'utilsation du composant "\Elixir\Session\Session",**

**\* l'ajout d'headers via la fonction "apache\_request\_headers", etc...**

**\***

**\* @param Sanitizer $pSanitizer**

**\* @return Request**

**\*/**

**/\*\***

**\* Contient toutes les informations utiles à l'envoi de la réponse au client.**

**\***

**\* Utilisation:**

**\***

**\* // Création basique**

**\* $response = new Response('Hello world !', 200, 'HTTP/1.1', array('Content-Type' => 'text/html; charset=UTF-8'));**

**\***

**\* // Via la fabrique**

**\* $response = ResponseFactory::create('Hello world !',**

**\* 200,**

**\* 'HTTP/1.1',**

**\* array('Content-Type' => 'text/html; charset=UTF-8'));**

**\***

**\* $response->send();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**HTTP :**

**/\*\***

**\* Gère l'affichage de formulaire HTML.**

**\***

**\* Utilisation:**

**\***

**\* $formHelper = new Form(new Escaper('UTF-8'));**

**\***

**\* // Génère un formulaire complet**

**\* echo $formHelper->formAll($form);**

**\***

**\* // Permet un affichage plus subtil**

**\* echo $formHelper->openFormTag($form);**

**\* echo $formHelper->field($form->get('input-field');**

**\***

**\* echo $formHelper->openLabelTag();**

**\* echo $formHelper->input($form->get('input-field');**

**\* echo $formHelper->closeLabelTag();**

**\* echo $formHelper->fieldErrors($form->get('input-field');**

**\***

**\* echo $formHelper->closeFormTag();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Simule une redirection sans recharger la page.**

**\***

**\* Utilisation:**

**\***

**\* $forward = new Forward($application);**

**\* $response = $forward->to('default', 'index', 'index');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Simule une redirection sans recharger la page.**

**\***

**\* Utilisation:**

**\***

**\* $forward = new Forward($application);**

**\* $response = $forward->to('default', 'index', 'index');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Traduit une chaîne de caractères selon la locale demandée.**

**\***

**\* Utilisation:**

**\***

**\* $I18N = new I18N($translator);**

**\***

**\* $result = $I18N->translate('Hello world');**

**\* $result = $I18N->plurialize(array('are no dogs', 'is 1 dog', 'are {COUNT} dogs'), 0);**

**\* $result = $I18N->transPlural('There [are no dogs|is 1 dog|are {COUNT} dogs]', 2, 'fr-FR');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Localise une classe ou un fichier dans un module.**

**\***

**\* On utilise la nomenclature "(@NomDuModule)", si ce module est parent d'un autre module,**

**\* celui-ci sera utilisé à la place.**

**\***

**\* Utilisation:**

**\***

**\* $locator = new Locator($application);**

**\***

**\* $class = $locator->locateClass((@Namespace)\Hello\World);**

**\* $file = $locator->locateFile(medias/(@Namespace)/styles/hello.css);**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Rends une pagination.**

**\***

**\* Configuration du fichier "pagination.phtml":**

**\***

**\* <?php if(count($this->range) > 1): ?>**

**\* <ul class="pagination pagination-sm">**

**\* <?php if($this->pagination->hasPreviousItem()): ?>**

**\* <li><a href="?page=<?php echo ($this->pagination->getCurrentItem() - 1); ?>">&laquo;</a></li>**

**\* <?php endif; ?>**

**\***

**\* <?php foreach($this->range as $item): ?>**

**\* <li class="<?php echo $item->isSelected() ? 'active' : ''; ?>">**

**\* <a href="?page=<?php echo $item->getIndex(); ?>"><?php echo $item->getIndex(); ?></a>**

**\* </li>**

**\* <?php endforeach; ?>**

**\***

**\* <?php if($this->pagination->hasNextItem()): ?>**

**\* <li><a href="?page=<?php echo ($this->pagination->getCurrentItem() + 1); ?>">&raquo;</a></li>**

**\* <?php endif; ?>**

**\* </ul>**

**\* <?php endif; ?>**

**\***

**\* Utilisation:**

**\***

**\* $pagination = new Pagination();**

**\* $pagination->setTemplating($view);**

**\***

**\* echo $pagination->paginate($paginator, 'pagination.phtml');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Rends une sous partie d'un template.**

**\***

**\* Utilisation:**

**\***

**\* $partial = new Partial();**

**\* $partial->setTemplating($view);**

**\***

**\* echo $partial->render('file.phtml', array('id' => 6));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Rends un template.**

**\***

**\* Utilisation:**

**\***

**\* $render = new Render();**

**\* $render->setController($controller);**

**\***

**\* // Retourne une chaine de caractères**

**\* echo $render->render('file.phtml', array('id' => 6));**

**\***

**\* // Retourne une réponse HTTP**

**\* echo $render->renderResponse('file.phtml', array('id' => 6));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Construit une URL.**

**\***

**\* Utilisation:**

**\***

**\* $urlGenerator = new URL($request);**

**\* $urlGenerator->setRouter($router);**

**\***

**\* // Retourne une URL absolue**

**\* $URL = $urlGenerator->baseURL('medias/styles/hello.css');**

**\***

**\* // Retourne une URL vie une route**

**\* $URL = $urlGenerator->generate('nameOfRoute', array('id' => 9));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**I18N :**

**/\*\*** **/\*\***

**\* Gestion de l'internationalisation.**

**\***

**\* Se base sur l'extension "Intl" ou la classe "\Elixir\Locale" pour définir une langue par défaut.**

**\***

**\* Utilisation:**

**\***

**\* $translate = new I18N();**

**\* $translate->setLocale('fr-FR');**

**\***

**\* $translate->load(array(**

**\* \_\_DIR\_\_ . '/../../languages/translations.mo',**

**\* \_\_DIR\_\_ . '/../../languages/translations.json',**

**\* \_\_DIR\_\_ . '/../../languages/translations.csv'**

**\* ));**

**\***

**\* echo $translate->translate('Hello world !');**

**\* echo $translate->pluralize('I have [no fingers|one finger|{COUNT} fingers]', 10);**

**\* echo $translate->transPlural('There [are no dogs|is 1 dog|are {COUNT} dogs]', -5);**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**Logging :**

**/** **/\*\***

**\* Permet la journalisation des données.**

**\***

**\* Utilisation:**

**\***

**\* $logger = new Logger();**

**\***

**\* // Ajout d'un rédacteur de type "File"**

**\* $logger->addWriter(new File(logs.txt'));**

**\***

**\* // Log de type "INFO"**

**\* $logger->info('value info');**

**\***

**\* // Log de type "ERROR"**

**\* $logger->error('value error');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**MVC :**

**/\*\*** **/\*\***

**\* Controlleur de base.**

**\***

**\* Un controlleur doit posséder le suffix "Controller" et une action le suffix "Action".**

**\* Une action doit retourner soit une réponse ou plus simplement**

**\* une chaine de caractères (qui sera automatiquement converti en réponse).**

**\***

**\* Utilisation:**

**\***

**\* class IndexController extends ControllerAbstract**

**\* {**

**\* // Ajout d'aides d'actions à usage unique**

**\* // Identique à la méthode "getTemplatingHelpers" si ce n'est que cette dernière sert aux aides de vues**

**\* protected function getControllerHelpers()**

**\* {**

**\* $helpers = array();**

**\***

**\* $helpers['helper.forward'] = function()**

**\* {**

**\* return new Forward();**

**\* };**

**\***

**\* return $helpers;**

**\* }**

**\***

**\* public function indexAction()**

**\* {**

**\* // Effectue une redirection sans changement d'URL via une aide d'action**

**\* return $this->helper('helper.forward')->to('(@NomDuModule)', 'index', 'login');**

**\* }**

**\***

**\* public function loginAction()**

**\* {**

**\* // Génère et retourne une réponse via une aide d'action**

**\* return $this->helper('helper.render')->renderResponse(null, array('param' => 'value'));**

**\* }**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Controlleur de type REST.**

**\***

**\* Identique au controlleur de base si ce n'est que l'appel des actions se base en plus sur**

**\* les méthodes de requête du protocol HTTP.**

**\***

**\* Si l'action ne possède pas le préfix adéquat, l'action la plus propice**

**\* sera utilisée (HEAD devient GET, DELETE devient POST, etc...).**

**\***

**\* Utilisation:**

**\***

**\* class IndexController extends RESTFulControllerAbstract**

**\* {**

**\* // Appel de l'action "index" si la requête est de type GET**

**\* public function getIndexAction()**

**\* {**

**\* return 'GET treatment';**

**\* }**

**\***

**\* // Appel de l'action "index" si la requête est de type POST**

**\* public function postIndexAction()**

**\* {**

**\* return 'POST treatment';**

**\* }**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Un Module permet d'implémenter une ou plusieurs fonctionnalités dans une application MVC.**

**\***

**\* C'est un répertoire qui peut héberger à peu près tout: classes, contrôleurs,**

**\* ressources web (bien que toutes les ressources publiques seront déplacées dans un autre répertoire),**

**\* fichiers de configuration, etc...**

**\***

**\* Exemple de structure:**

**\***

**\* NomDuModule/**

**\* Bootstrap.php**

**\* Form/**

**\* LoginForm.php**

**\* Model/**

**\* User.php**

**\* Controller/**

**\* IndexController.php**

**\* resources/**

**\* config/**

**\* config.xml**

**\* db.php**

**\* views/**

**\* index/**

**\* login.phtml**

**\* routes/**

**\* routes.php**

**\* DI/**

**\* Service.php**

**\* Listener/**

**\* Listeners.php**

**\***

**\* Configuration:**

**\***

**\* namespace NomDuModule;**

**\***

**\* class Bootstrap extends ModuleAbstract**

**\* {**

**\* // Si ce module surcharge un autre, on précise lequel**

**\* public function getParent()**

**\* {**

**\* return 'NomDuModuleASurcharger';**

**\* }**

**\***

**\* public function boot(DispatcherInterface $pDispatcher, ContainerInterface $pContainer)**

**\* {**

**\* parent::boot($pDispatcher, $pContainer);**

**\***

**\* $app = $this->\_container->get('application'); // Ou directement "$pDispatcher"**

**\***

**\* // Etend le container via un fournisseur**

**\* // On récupère dynamiquement la classe en laissant la possibilité à un autre module**

**\* // de la redéfinir (via "(@NomDuModule)")**

**\* $services = $app->locateClass('(@NomDuModule)\DI\Services');**

**\* $this->\_container->load(new $services());**

**\***

**\* // Ajout d'évènements via un souscripteur d'événements**

**\* $subscribers = $app->locateClass('(@NomDuModule)\Listener\Listeners');**

**\* $this->\_dispatcher->addSubscriber(new $subscribers($this->\_container));**

**\***

**\* // Ajout d'une route**

**\* // Notez l'utilisation de la syntaxe "(@NomDuModule)" afin de permettre aux autres modules**

**\* // la possibilité de surcharger le controlleur**

**\* $router = $this->\_container->get('router');**

**\* $router->getCollection()->add('NomDuModule', new Route('/nom-du-module/{\_controller}/{\_action}',**

**\* array('\_module' => '(@NomDuModule)'),**

**\* array('\_controller' => '[a-z0-9-]+',**

**\* '\_action' => '[a-z0-9-]+',**

**\* '\*' => true)));**

**\* // Autres traitements...**

**\* }**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Création d'une application de type MVC (Model-Vue-Controlleur).**

**\***

**\* L'idée est de séparer l'application en plusieurs modules de manière indépendante.**

**\* Un module est un répertoire qui peut héberger à peu près tout: classes, contrôleurs,**

**\* ressources web (bien que toutes les ressources publiques seront déplacées dans un autre répertoire),**

**\* fichiers de configuration, etc...**

**\* La seule contrainte est de posseder une classe implementant "ModuleInterface".**

**\***

**\* Un module peut étendre un autre module permettant pas exemple de remplacer une vue**

**\* par une autre, surcharger une action d'un controlleur etc...**

**\***

**\* L'application possède deux méthodes qui permettent la surcharge de fichiers ("locateClass" et "locateFile").**

**\* En utilisation la syntaxe "(@NomDuModule)" dans les chemins, l'application va substituer cette partie par**

**\* le nom/namespace/chemin (suivant les cas) du module enfant (ou enfant de l'enfant etc...).**

**\***

**\* Par exemple un module peut s'occuper de la partie admin, un autre du site en lui-même, mais un module peut tout simplement**

**\* configurer une partie de votre application (comme des aides de vues etc...).**

**\***

**\* Un des points central d'une application est le container d'injection de dépendance, il contient**

**\* tout ce qui est essentiel à l'application, il passe de modules en modules qui peuvent donc lui attribuer**

**\* divers services.**

**\***

**\* Le second point central est l'objet de requête, vu qu'il contient les informations concernant le client.**

**\***

**\* Le rôle du résolveur de controlleurs est de derterminer en en fonction de l'objet de requête**

**\* le controlleur à executer ainsi que les arguments à lui transmettre.**

**\***

**\* Utilisation:**

**\***

**\* // On crée l'application en y injectant un container d'injection de dépendance**

**\* $application = new Application(new Container());**

**\***

**\* // Permet la redirection vers un controlleur en fonction de la requête**

**\* $application->setControllerResolver(new ControllerResolver());**

**\***

**\* // Ajout du module de base (contient la configuration par défaut)**

**\* $application->addModule(new Elixir\Module\Application\Bootstrap());**

**\***

**\* // Ajout d'autres modules**

**\* $application->addModule(new Hello\Bootstrap());**

**\***

**\* // Initialize tous les modules**

**\* $application->boot();**

**\***

**\* // Crée la réponse**

**\* $response = $application->handle(RequestFactory::create());**

**\* $response->send();**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**PAGINATION :**

**/\*** **/\*\***

**\* Permet la pagination de collections de données et la présentation de ces données.**

**\***

**\* Ce composant ne s'occupe ni de la découpe des données (le nombre d'items par page et celui au total doit être calculé par l'utilisateur) ni de son rendu.**

**\* S'utilise en corrélation avec l'aide "\Elixir\Helper\Pagination" qui s'occupe du rendu de la pagination.**

**\***

**\* Utilisation:**

**\***

**\* $max = 100;**

**\* $total = 1000;**

**\* $maxPage = ceil($total / $max);**

**\* $page = 6;**

**\***

**\* if($page > $maxPage)**

**\* {**

**\* $page = $maxPage;**

**\* }**

**\* else if($page < 1)**

**\* {**

**\* $page = 1;**

**\* }**

**\***

**\* $paginator = new Pagination(new Elastic());**

**\* $paginator->setItemsPerPage(10);**

**\* $paginator->setTotalItems($maxPage);**

**\* $paginator->setCurrentItem($page);**

**\***

**\* // Rendu de la pagination**

**\* $helper = new \Elixir\Helper\Pagination();**

**\* $helper->setTemplating($view);**

**\***

**\* echo $helper->paginate($paginator, 'pagination.phtml');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**PARAMETERS :**

**/\*** **/\*\***

**\* Se comporte comme un tableau avec la possiblité d'échappement des données.**

**\***

**\* Utilisation:**

**\***

**\* $data = array('key' => 15);**

**\* $parameter = new Parameters($data);**

**\***

**\* // Lecture**

**\* $value = $parameter->get('key', array('filter' => FILTER\_SANITIZE\_NUMBER\_INT)); // $parameter->key; // $parameter['key'];**

**\***

**\* // Ecriture**

**\* $parameter->set('key', 'new value'); // $parameter->key = 'new value'; // $parameter['key'] = 'new value';**

**\***

**\* // Utilisation de la notation pointée**

**\* $parameter->set('multi:key', 'value');**

**\***

**\* // Utilisation de la notation sous forme de tableau**

**\* $value = $parameter->get(array('multi', key'));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**ROUTING :**

**/\*** **/\*\***

**\* Génère une URL où la route se trouve dans un paramètre de type "GET" (?r=nom/de/ma/route).**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Génère une URL via une route.**

**\***

**\* Si la propriétée "strict" vaut "true", alors les options seront validées selon leurs contraintes.**

**\***

**\* Utilisation:**

**\***

**\* $generator = new URLGenerator(RequestFactory::create());**

**\* $url = $generator->generate(new Route('profil', array(Route::MVC => 'default::profil::index'), array('\*' => true)),**

**\* array('name' => 'username', 'id' => '6'),**

**\* GeneratorInterface::URL\_RELATIVE); // profil/username/id/6**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Fait la correspondance entre paramètre de type "GET" et une route.**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Fait la correspondance entre une URL et une route.**

**\***

**\* Utilisation:**

**\***

**\* $collection = new Collection();**

**\* $collection->add('profil',**

**\* new Route('profil/{name}',**

**\* array(Route::MVC => 'default::profil::index'),**

**\* array('name' => '[a-z]+'))));**

**\***

**\* $matcher = new URLMatcher(RequestFactory::create());**

**\* $route = $matcher->match($collection, 'profil/username'); // Match route "profil"**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Le routeur est l'objet qui va permettre de savoir quelle action doit être exécutée.**

**\***

**\* En règle générale, le routage examinera une URL, et tentera de la faire correspondre avec une route.**

**\* C'est à dire un segment d'URL associé à des paramètres (module, controlleur, action, etc...).**

**\***

**\* Utilisation:**

**\***

**\* $collection = new Collection();**

**\* $collection->add('profil',**

**\* new Route('profil/{name}',**

**\* array(Route::MVC => 'default::profil::index'),**

**\* array('name' => '[a-z]+')**

**\* Route::SECURE => false,**

**\* Route::METHOD => array('GET', 'POST'),**

**\* Route::ASSERT => function($pMatch)**

**\* {**

**\* echo 'Match called';**

**\* return true;**

**\* },**

**\* Route::ATTRIBUTES => true)));**

**\***

**\* $request = RequestFactory::create();**

**\* $request->setBaseURL('http://www.example.com');**

**\***

**\* $routing = new Router($collection);**

**\* $routing->setRequest($request);**

**\* $routing->setURLMatcher(new URLMatcher());**

**\* $routing->setURLGenerator(new URLGenerator());**

**\***

**\* $route = $routing->match('profil/username'); // Match route "profil"**

**\* $url = $routing->generate('profil', array('name' => 'username', 'id' => '6'); // http://www.example.com/profil/username/id/6**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**SECURITY :**

**/\*** **/\*\***

**\* Stocke les données d'un utilisateur authentifié (le plus souvent en session).**

**\***

**\* Utilisation:**

**\***

**\* $manager = new Manager(new SessionStorage(Session::instance()));**

**\***

**\* // Via authentification**

**\* $manager->authenticate(new CustomAuthentification('user@example.com', 'password'), Manager::DEFAULT\_IDENTITY);**

**\***

**\* // Directement**

**\* $manager->set(**

**\* Manager::DEFAULT\_IDENTITY,**

**\* new Identity(array('USER', 'ADMIN'),**

**\* array(**

**\* 'id' => 6,**

**\* 'pseudo' => 'user-pseudo',**

**\* 'email' => 'user@example.com'**

**\* )**

**\* );**

**\***

**\* $identity = $manager->get(Manager::DEFAULT\_IDENTITY);**

**\* $identity->pseudo = 'new-user-pseudo';**

**\***

**\* $manager->update($identity);**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Pare-feu autorisant l'accès ou non à une ressource selon les droits nécessaires pour le faire.**

**\***

**\* Une ressource peut être défini par une simple chaîne de caractères ou par une expression régulière.**

**\* Fonctionne en corrélation avec le composant "Elixir\Security\Authentification\Manager".**

**\***

**\* Utilisation:**

**\***

**\* $firewall = new Firewall($manager);**

**\***

**\* // Via un fichier**

**\* $firewall->load(\_\_DIR\_\_ . '/../resources/security/security.php');**

**\***

**\* // Via un contrôle d'accès**

**\* $ac = new AccessControl('/protect-me-url/', array('domains' => 'default'));**

**\* $firewall->addAccessControl($ac, 100);**

**\***

**\* $firewall->addListener(FirewallEvent::IDENTITY\_NOT\_FOUND, function(FirewallEvent $e)**

**\* {**

**\* $options = $e->getAccessControl()->options();**

**\***

**\* if(isset($options['identity\_not\_found\_uri']))**

**\* {**

**\* $behavior = new IdentityNotFound($options['identity\_not\_found\_uri']);**

**\* }**

**\* else**

**\* {**

**\* $behavior = new AccessForbidden();**

**\* }**

**\***

**\* $behavior($e->getTarget());**

**\* });**

**\***

**\* $firewall->addListener(FirewallEvent::ACCESS\_FORBIDDEN, function(FirewallEvent $e)**

**\* {**

**\* $behavior = new AccessForbidden();**

**\* $behavior($e->getTarget());**

**\* });**

**\***

**\* $firewall->analyse($resource);**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Pare-feu autorisant l'accès ou non à une ressource selon les droits nécessaires pour le faire.**

**\***

**\* Une ressource peut être défini par une simple chaîne de caractères ou par une expression régulière.**

**\* Fonctionne en corrélation avec le composant "Elixir\Security\Authentification\Manager" possédant des identités**

**\* ayant comme context de sécurité le composant "Elixir\Security\RBAC\RBAC".**

**\***

**\* Utilisation:**

**\***

**\* $firewall = new Firewall($manager);**

**\***

**\* // Via un fichier**

**\* $firewall->load(\_\_DIR\_\_ . '/../resources/security/security.php');**

**\***

**\* // Via un contrôle d'accès**

**\* $ac = new AccessControl('/protect-me-url/', array(**

**\* 'roles' => 'user',**

**\* 'permissions' => 'read-article',**

**\* 'assert' => null,**

**\* 'domains' => 'default'**

**\* ));**

**\***

**\* $firewall->addAccessControl($ac, 100);**

**\***

**\* $firewall->addListener(FirewallEvent::IDENTITY\_NOT\_FOUND, function(FirewallEvent $e)**

**\* {**

**\* $options = $e->getAccessControl()->options();**

**\***

**\* if(isset($options['identity\_not\_found\_uri']))**

**\* {**

**\* $behavior = new IdentityNotFound($options['identity\_not\_found\_uri']);**

**\* }**

**\* else**

**\* {**

**\* $behavior = new AccessForbidden();**

**\* }**

**\***

**\* $behavior($e->getTarget());**

**\* });**

**\***

**\* $firewall->addListener(FirewallEvent::ACCESS\_FORBIDDEN, function(FirewallEvent $e)**

**\* {**

**\* $behavior = new AccessForbidden();**

**\* $behavior($e->getTarget());**

**\* });**

**\***

**\* $firewall->analyse($resource);**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Modèle de contrôle d'accès dans lequel chaque décision d'accès est basée sur le rôle auquel l'utilisateur est attaché.**

**\***

**\* Utilisation:**

**\***

**\* $RBAC = new RBAC();**

**\***

**\* // Ajout d'un rôle utilisateur**

**\* $RBAC->addRole(new Role('user', array('read')));**

**\***

**\* // Ajout d'un rôle rédacteur héritant du rôle utilisateur**

**\* $RBAC->addRole(new Role('redactor', array('write')), array('user'));**

**\***

**\* // Ajout d'un rôle administrateur**

**\* $RBAC->addRole(new Role('admin', array(Role::ALL\_PERMISSIONS\_GRANTED)));**

**\***

**\* $result = $RBAC->isGranted('user', 'read', null); // true**

**\* $result = $RBAC->isGranted('user', 'write', null); // false**

**\* $result = $RBAC->isGranted('redactor', 'read', null); // true**

**\***

**\* $result = $RBAC->isGranted('admin', null, function($pHasRole, $pHasPermission, $pRBAC) use($identity)**

**\* {**

**\* return $identity->id == 6;**

**\* }); // true ou false selon l'id de l'identité**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Sécurité anti "CSRF".**

**\***

**\* Utilisation:**

**\***

**\* $CSRF = new CSRF($request);**

**\***

**\* // Crée le jeton à injecter dans le formulaire de la page "http://www.previous.com"**

**\* $token = $CSRF->create('token');**

**\***

**\* $result = $CSRF->isValid('token', 900, 'http://www.previous.com');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**SESSION :**

**/\*** **/\*\***

**\* Stocke les données de session en base de données.**

**\***

**\* Model de table :**

**\***

**\* CREATE TABLE `sessions` (**

**\* `id` varchar(255) binary NOT NULL default '',**

**\* `expires` int(10) unsigned NOT NULL default '0',**

**\* `data` text,**

**\* PRIMARY KEY (`id`));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Simule une session PHP (moyen de préserver des données entre plusieurs accès).**

**\***

**\* Cette classe est un singleton mais ne s'instancie pas automatiquement (permet une instantiation**

**\* avec paramètres).**

**\***

**\* Utilisation:**

**\***

**\* if(null === Session::instance())**

**\* {**

**\* $session = new Session();**

**\* }**

**\***

**\* if(!Session::instance()->exist())**

**\* {**

**\* Session::instance()->setName('sessionName');**

**\* Session::instance()->start();**

**\* }**

**\***

**\* // Ecriture**

**\* Session::instance()->set('key', 'value'); // Session::instance()->key = 'value'; // $session['key'] = 'value';**

**\***

**\* // Lecture**

**\* $value = Session::instance()->get('key'); // Session::instance()->key; // $session['key'];**

**\***

**\* // Utilisation de la notation pointée**

**\* Session::instance()->set('multi:key', 'value');**

**\***

**\* // Utilisation de la notation sous forme de tableau**

**\* $value = Session::instance()->get(array('multi', key'));**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**TEMPLATING :**

**/\*** **/\*\***

**\* Moteur de template utilisant des fichiers de templates contenant du PHP.**

**\***

**\* Utilisation:**

**\***

**\* $template = new PHP();**

**\* $template->setHelperContainer(new Container());**

**\* $template->set('key', 'value'); // $template->key = 'value';**

**\* $result = $template->render('file.phtml');**

**\***

**\* Fichier layout.phtml:**

**\***

**\* <html>**

**\* <head>**

**\* <title>LAYOUT</title>**

**\* <?php $this->start('styles'); ?>**

**\* <link href="layout.css" rel="stylesheet">**

**\* <?php $this->end(); ?>**

**\* <?php echo $this->block('styles'); ?>**

**\* </head>**

**\* <body>**

**\* <?php echo $this->\_content; ?>**

**\* </body>**

**\* </html>**

**\***

**\* Fichier file.phtml:**

**\***

**\* <?php $this->extend('layout.phtml'); ?>**

**\* <?php $this->start('styles'); ?>**

**\* <?php echo $this->parent(); ?>**

**\* <link href="file.css" rel="stylesheet">**

**\* <?php $this->end(); ?>**

**\* <p>My key equal <?php echo $this->helper('helper.escaper')->escapeHTML($this->key); ?></p>**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Groupe plusieurs moteurs de template.**

**\* Se base sur l'extension du fichier pour choisir le moteur le plus approprié.**

**\***

**\* Utilisation:**

**\***

**\* $manager = new Manager();**

**$manager->registerExtension('^(phtml|php)$', new PHP());**

**\* $manager->setHelperContainer(new Container());**

**\* $manager->set('key', 'value'); // $manager->key = 'value';**

**\* $result = $manager->render('file.phtml');**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**TREE :**

**/\*** **/** **/\*\***

**\* Fabrication d'un arbre d'éléments.**

**\***

**\* Base pour la création de menus, fils d'Ariane etc...**

**\* Permet la recherche d'éléments selon leurs paramètres.**

**\***

**\* Utilisation:**

**\***

**\* $id = 1;**

**\***

**\* $tree = new Tree();**

**\* $tree->setParameter('id', $id);**

**\***

**\* for($i = 0; $i < 2; ++$i)**

**\* {**

**\* $id++;**

**\***

**\* $tree1 = new Tree();**

**\* $tree1->setParameter('id', $id);**

**\***

**\* for($j = 0; $j < 2; ++$j)**

**\* {**

**\* $id++;**

**\***

**\* $tree2 = new Tree();**

**\* $tree2->setParameter('id', $id);**

**\***

**\* $tree1->addChild($tree2);**

**\* }**

**\***

**\* $tree->addChild($tree1);**

**\* }**

**\***

**\* $result = $tree->find(array('id' => 2), Tree::ALL\_LEVEL, true);**

**\***

**\* @author Nicola Pertosa <n.pertosa@peoleo.fr>**

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**VALIDATOR :**

**/\*** **/\*\***

**\* Validation d'un type "boolean".**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Boolean();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Le contenu n'est pas de type "boolean"');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid(true, array('error' => 'Le contenu n'est pas de type "boolean"')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation anti "CSRF".**

**\***

**\* Utilisation:**

**\***

**\* $validator = new CSRF();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Possibilité d\'attaque de type "CSRF"');**

**\* $token = $validator->createToken('token');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('token', array('error' => 'Possibilité d\'attaque de type "CSRF"')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation via une fonction de rappel.**

**\***

**\* Utilisation:**

**\***

**\* $callback = function($pValue)**

**\* {**

**\* return trim($pValue) !== '';**

**\* }**

**\***

**\* $validator = new Callback();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('La chaîne ne peut être vide');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('hello world', array('options' => $callback,**

**\* 'error' => 'La chaîne ne peut être vide')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Permet le chainage de validateurs en un seul.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Chain();**

**\* $validator->addValidator(new Email());**

**\* $validator->addValidator(new NotEmpty());**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Données invalides');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('hello@.fr', array('error' => 'Données invalides')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation d'une date.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Date();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Date invalide');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('2013-10-24', array('error' => 'Date invalide',**

**\* 'format' => 'Y-m-d')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation d'une adresse email.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Email();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Email invalide');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('hello@example.com', array('error' => 'Email invalide')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation d'égalité entre deux valeurs.**

**\***

**\* La comparaison peut directement être effectuée sur un élément de formulaire.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Equal();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('La valeur n\'est pas égale à la valeur de comparaison');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('my value', array('error' => 'La valeur n\'est pas égale à la valeur de comparaison',**

**\* 'compare' => $form->get('input-comparison'))))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation de l'extension du nom d'un fichier.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Extension();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('L\'extension du fichier est invalide.');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('file.gif', array('error' => 'L\'extension du fichier est invalide.',**

**\* 'extension' => array('jpeg', 'png'))))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation du poids d'un fichier.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new FileSize();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Le poids du fichier est invalide.');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('file.gif', array('error' => 'Le poids du fichier est invalide.',**

**\* 'max' => 1000)))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation sur les nombres à virgules flottantes.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Float();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Ce n\'est pas un nombre valide');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('102.5f', array('error' => 'Ce n\'est pas un nombre valide')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation des dimensions d'un fichier.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Format();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Les dimensions du fichier sont invalides.');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('file.gif', array('error' => 'Les dimensions du fichier sont invalides.',**

**\* 'maxWidth' => 150,**

**\* 'maxheight' => 150)))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation sur les adresses IP.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new IP();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Adresse IP invalide');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('84.55.180.213', array('error' => 'Adresse IP invalide')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation sur les nombres entiers.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Float();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Ce n\'est pas un nombre valide');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('99', array('error' => 'Ce n\'est pas un nombre valide')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation de la longueur d'une chaîne de caractères.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Length();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('La longueur de la chaîne de caractères est invalide (minimum {MIN} caratères).');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('hello world', array('min' => 6,**

**\* 'error' => 'La longueur de la chaîne de caractères est invalide (minimum {MIN} caratères).')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation du type mime d'un fichier.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new MimeType();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Le type mime du fichier est invalide.');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('file.gif', array('error' => 'Le type mime du fichier est invalide.',**

**\* 'mimeType' => 'image/gif')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Valide qu'une chaîne de caractères est non vide.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new NotEmpty();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Chaîne invalide');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid(' ', array('error' => 'Chaîne invalide')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation d'un nombre dans une plage délimitée par min et/ou max.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Range();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('La valeur est en dehors des limites.');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('75', array('min' => 0,**

**\* 'max' => 100,**

**\* 'error' => 'La valeur est en dehors des limites.')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation via expression régulière.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new Regex();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('Aucun résultat trouvé.');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('Hello world', array('regex' => '/^hello/',**

**\* 'match' => true,**

**\* 'error' => 'Aucun résultat trouvé.')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation d'URL.**

**\***

**\* Utilisation:**

**\***

**\* $validator = new URL();**

**\***

**\* // Ajout d'un message d'erreur personnalisé**

**\* $validator->setErrorMessage('URL invalide');**

**\***

**\* // Ou directement dans la méthode**

**\* if($validator->isValid('http://www.elixir.com', array('error' => 'URL invalide')))**

**\* {**

**\* // Contenu valide**

**\* }**

**\* else**

**\* {**

**\* $errors = $validator->errors();**

**\* }**

**\***

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**/\*\***

**\* Validation statique.**

**\***

**\* Utilisation:**

**\***

**\* // Si on possede un container d'injection de dépendance, la recherche ce basera sur celui-ci en priorité**

**\* Validator::setContainer($container);**

**\***

**\* $isValid = Validator::valid('\Elixir\Validator\Email', 'hello@example.com');**

**\* $isValid = Validator::valid('\Elixir\Validator\Date', '2013-10-24', array('format' => 'Y-m-d'));**

**\***

**\* @see http://fr.wikipedia.org/wiki/Fa%C3%A7ade\_(patron\_de\_conception)**

**\* @author Cédric Tanghe <c.tanghe@peoleo.fr>**

**\*/**

**ARR :**

**/\*\*** **/\*\***

**\* Vérifie l'existence une ou plusieurs clé(s) d'un tableau.**

**\***

**\* @param mixed $pKey**

**\* @param array $pData**

**\* @return boolean**

**\*/**

**/\*\***

**\* Retourne une donnée d'un tableau.**

**\***

**\* @param mixed $pKey**

**\* @param array $pData**

**\* @param mixed $pDefault**

**\* @return mixed**

**\*/**

**/\*\***

**\* Spécifie une donnée d'un tableau.**

**\***

**\* @param mixed $pKey**

**\* @param mixed $pValue**

**\* @param array $pData**

**\*/**

**/\*\***

**\* Supprime une donnée d'un tableau.**

**\***

**\* @param mixed $pKey**

**\* @param array $pData**

**\*/**

**/\*\***

**\* Fusionne deux tableaux de manière récursive.**

**\***

**\* @param array $pA**

**\* @param array $pB**

**\* @return array**

**\*/**

**CSV :**

**/\*** **/\*\***

**\* Converti du contenu au format CSV en tableau PHP.**

**\***

**\* @param string $pCsv**

**\* @param string $pDelimiter**

**\* @param string $pEnclosure**

**\* @param boolean $pAssociate**

**\* @return array**

**\*/**

**/\*\***

**\* Converti un tableau PHP en contenu au format CSV.**

**\***

**\* @param array $pArray**

**\* @param string $pDelimiter**

**\* @param string $pEnclosure**

**\* @param boolean $pAssociate**

**\* @return string**

**\*/**

**FILE :**

**/\*** **/** **/\*\***

**\* Retourne l'extension du fichier**

**\***

**\* @param string $pFilePath**

**\* @return string**

**\*/**

**/\*\***

**\* Retourne le chemin du fichier**

**\***

**\* @param string $pFilePath**

**\* @return string**

**\*/**

**/\*\***

**\* Retourne le nom du fichier sans extension**

**\***

**\* @param string $pFilePath**

**\* @return string**

**\*/**

**/\*\***

**\* Retourne le type mime du fichier**

**\***

**\* @param string $pFilePath**

**\* @return string**

**\*/**

**/\*\***

**\* Copie un fichier/dossier**

**\***

**\* @param string $pSrcPath**

**\* @param string $pDstPath**

**\* @param boolean $pOverride**

**\* @return boolean**

**\*/**

**/\*\***

**\* Supprime un fichier/dossier**

**\***

**\* @param string $pFilePath**

**\* @return boolean**

**\*/**

**IMAGE :**

**/\*** **/** **/\*\***

**\* Retourne un ensemble d'informations concernant une image si un redimensionnement était effectué.**

**\***

**\* @param string $pImgPath**

**\* @param integer $pW**

**\* @param integer $pH**

**\* @param boolean $pRatio**

**\* @param string $pMode**

**\* @return \stdClass|null**

**\*/**

**/\*\***

**\* Redimentionne une image en s'assurant qu'elle rentre entièrement dans l'espace qui lui est attribué.**

**\***

**\* @param string $pSrcPath**

**\* @param string $pDstPath**

**\* @param integer $pWMax**

**\* @param integer $pHMax**

**\* @param boolean $pRatio**

**\* @return boolean**

**\*/**

**/\*\***

**\* Redimentionne une image en s'assurant qu'elle prenne tout l'espace qui lui est attribué.**

**\***

**\* @param string $pSrcPath**

**\* @param string $pDstPath**

**\* @param integer $pWMin**

**\* @param integer $pHMin**

**\* @param boolean $pRatio**

**\* @return boolean**

**\*/**

**/\*\***

**\* Recadre une image**

**\***

**\* @param string $pSrcPath**

**\* @param string $pDstPath**

**\* @param integer $pWEnd**

**\* @param integer $pHEnd**

**\* @return boolean**

**\*/**

**STR :**

**/** **/\*\***

**\* Supprime les accents d'une chaîne de caractères.**

**\***

**\* @param string $pStr**

**\* @param string $pCharset**

**\* @return string**

**\*/**

**/\*\***

**\* Applique une notation dite "CamelCase".**

**\***

**\* @param string $pStr**

**\* @return string**

**\*/**

**/\*\***

**\* "Slugifie" une chaîne de caractères.**

**\***

**\* @param string $pStr**

**\* @param string $pSeparator**

**\* @return string**

**\*/**

**/\*\***

**\* Retourne une chaîne découpée en fonction du nombre maximum de caractères.**

**\***

**\* @param string $pStr**

**\* @param integer $pMaxCharacters**

**\* @param string $pCut**

**\* @param boolean $pUseWord**

**\* @return string**

**\*/**

**/\*\***

**\* Génère un ensemble mots-clés à partir d'une chaîne de caractères.**

**\***

**\* @param string $pStr**

**\* @param integer $pSensitivity**

**\* @param integer $pMin**

**\* @param integer $pMax**

**\* @param integer $pLimit**

**\* @return array**

**\*/**