View Helpers

La classe Phalcon\Tag fourni un ensemble de méthodes facilitant l'écriture des vues, avec phtml ou avec volt.

Elle est définie en tant que service et est accessible depuis les vues et les controlleurs.

-- Doctype

Définition du doctype dans le contrôleur :

```
...
public function testAction(){
    $this->tag->setDoctype(\Phalcon\Tag::HTML401_STRICT);
    echo $this->view->render("test");
}
```

Utilisation du docType dans la vue :

```
<?php echo $this->tag->getDocType();
```

```
{{ get_doctype() }}
```

Résultat

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head></head>
...
```

-- Génération de liens

```
<!-- for the default route -->
<?= $this->tag->linkTo("products/search", "Search") ?>

<!-- with CSS attributes -->
<?= $this->tag->linkTo(array('products/edit/10', 'Edit', 'class' => 'edit-btn')) ?>

<!-- for a named route -->
<?= $this->tag->linkTo(array(array('for' => 'show-product', 'title' => 123, 'name'
=> 'carrots'), 'Show')) ?>
```

-- Création de formulaires

phtml

```
<?php
    $this->tag->form(array("products/search", "method" => "get"));
    $this->tag->endForm();
```

volt

```
{{ form("products/search", "method": "get") }}
{{end_form()}}
```

-- Éléments de formulaires

phtml

```
<?php
echo $this->tag->textField("username");

echo $this->tag->textArea(array(
    "comment",
    "This is the content of the text-area",
    "cols" => "6",
    "rows" => 20
));

echo $this->tag->passwordField(array(
    "password",
    "size" => 30
));

echo $this->tag->hiddenField(array(
    "parent_id",
    "value"=> "5"
));
```

volt

```
{{ text_field("username") }}
```

```
{{ text_area("comment", "This is the content", "cols": "6", "rows": 20) }}

{{ password_field("password", "size": 30) }}

{{ hidden_field("parent_id", "value": "5") }}
```

-- Modification des valeurs des éléments

Les éléments de formulaire possédant un attribut **value** peuvent être affectés à la création :

phtml

```
<?php
echo $this->tag->textField(array("username",value:"SMITH"));
echo $this->tag->textArea(array(
    "comment",
    "This is the content of the text-area",
    "cols" => "6",
    "rows" => 20
)) ?>
```

volt

```
{{ text_field("username","value":"SMITH") }}

{{ text_area("comment", "This is the content", "cols": "6", "rows": 20) }}
```

Il est également possible, préférable (pour l'architecture MVC) et inévitable pour certains tags (select par exemple) de faire cette affectation dans le contrôleur :

```
class ProductsController extends Controller
{
    public function indexAction()
    {
        $this->tag->setDefault("username", "SMITH");
    }
}
```

-- Contenus statiques

Feuilles de style

phtml

```
<?php
    echo
Phalcon\Tag::stylesheetLink("http://fonts.googleapis.com/css?family=Rosario",
false);
    echo Phalcon\Tag::stylesheetLink("css/style.css");
```

volt

```
    {{ stylesheet_link("http://fonts.googleapis.com/css?family=Rosario", false) }}
    {{ stylesheet_link("css/style.css") }}
```

javascript file

phtml

```
<?php

        echo
Phalcon\Tag::javascriptInclude("http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/j
query.min.js", false);
        echo Phalcon\Tag::javascriptInclude("javascript/jquery.js");
```

volt

```
 {{
javascript_include("http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js
", false) }}
 {{ javascript_include("javascript/jquery.js") }}
```

Images

phtml

```
<?php

        echo Phalcon\Tag::image("img/bg.png");
        echo Phalcon\Tag::image(array("img/photo.jpg", "alt" => "Some Photo"));
```

volt

```
{{ image("img/bg.png") }}
{{ image("img/photo.jpg", "alt": "Some Photo") }}
{{ image("http://static.mywebsite.com/img/bg.png", false) }}
```

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/slam4/php/phalcon/viewhelpers

Last update: **2019/08/31 14:21**