

Shortcodes are coming to XOOPS 2.6.0 :D - XOOPS

NEWS_PDF_AUTHOR: Mamba

NEWS_PDF_DATE: 2015/11/1 10:50:00

Richard continues to doing amazing changes to XOOPS 2.6.0. The latest addition are "Shortcodes", which are probably best known from WordPress. I was actually hoping for it for a long, long time, and I was very happy to see that Richard has added it to XOOPS! YES!!! 😊

It was possible thanks to refactoring of our old TextSanitizer. You can read more details from [Richard's info on GitHub](#)

Quote:

MyTextSanitizer has become Xoops\Core\Text\Sanitizer.

Highlights:

- Single unified configuration file

[xoops_data/configs/system_sanitizer_prefs.yml](#)

- Extensions (i.e. YouTube) are now a SanitizerComponent, a base class that also includes Filters (i.e. Xss)

- Each SanitizerComponent auto-configures, and configuration is automatically updated with new additions

- Custom components can be added via response to `core.sanitizer.configuration.defaults` event

- ShortCodes (WordPress style) are now the basis of XoopsCode and extensions

- Custom ShortCodes can be added via response to

[core.sanitizer.shortcuts.add](#) event, or by directly accessing the ShortCodes engine with

`Sanitizer::getShortCodes()`

- All built-in components and shortcodes can be overridden with same named custom replacements
- Some extensions have updated syntax. For example:

```
[youtube=640,385]https://www.youtube.com/watch?v=JxS5E-kZc2s[/youtube]
```

can now be:

```
[youtube url="JxS5E-kZc2s" width="640" height="385" /]
```

(It also can use virtually any known YouTube URL as the url attribute.)

- Another example is SoundCloud, which now can directly support the "Wordpress code" option offered in the Share/Embed dialog on soundcloud.com.

The old styles are still accepted for compatibility with existing content, and not all extensions have conversions to new styles, yet. There is more work to be done, but the backing code is now in place.

What does it mean for XOOPS developers? We'll be able now to simplify a lot of activities around designing the best looking Websites for our users and customers! 😊

Also our users will be able to add tons of cool features without any programming! 😊

For those of you who never worked with ShortCodes, let's provide a short overview and a little tutorial. The basic classes that we'll be using, are the TextSanitizer classes refactored by Richard:

Currently we have following classes derived from **FilterAbstract** :

- Censor
- Clickable
- Quote
- SyntaxHighlight
- TextFilter
- Xss

and following classes derived from **ExtensionAbstract**:

- Flash
- Iframe
- Image
- Mms
- Mp3
- Rtsp
- SoundCloud
- UnorderedList
- Wiki
- Wmp
- XoopsCode
- YouTube

For our purposes we'll be using mainly the **ExtensionAbstract** class. Let's see now how can we use it to create ShortCodes for **Google Charts** and **Google Maps** ([Read more-->](#))

But before we go there, lets provide a short overview of **ShortCodes** **ShortCodes** are nothing different than macros or scripts that you create to repeating the some activity over and over again. They can be very simple, and they can be very complex. Some of them are true code snippets, with no direct interaction, and some can be quite sophisticated, with visual GUI that helps you to select its options. There are also different ways to interact with Shortodes: a) simple ShortCodes without any attributes, using basically a single command, e.g. our BBcodes are simple Shortcodes that we use by placing [**b**] and [/**b**] around text to make it bold b) more complex ShortCodes with Attributes, like the YouTube example presented by Richard above:

```
[youtube url="JxS5E-kZc2s" width="640" height="385" /]
```

Creating a Shortcode for Google Maps After I saw yesterday Richard's announcement about the Shortcodes in XOOPS, I've got very excited and decided to explore them. In order to create a Shortcode, we'll be using the **ExtensionAbstract** class. We create a new file in the /Extensions folder called "Googlemap.php". We'll start with:

```
class Googlemap extends ExtensionAbstract
{
}
```

A Google map has just very basic features, like width, height, etc. So as first, we can create a default configuration for it. This will also make it easier for us, as we won't have to then fill out all the options every time we use it:

```
protected static $defaultConfiguration = [
'enabled' => true,
'template' => "",
'clickable' => true, // Click to open a map in a new window in full size
'resize' => true, // Resize the map down to max_width set below
'max_width' => 640, // Maximum width of a map displayed on page
'max_height' => 480, // Maximum width of a map displayed on page
'allowmap' => true, // true to allow maps, false to force links only
```

```
];
```

Of course, you can decide for yourself which options do you want to include and which one not. As next step, we need to register our extension with the Sanitizer. This is being done by overriding the abstract method **registerExtensionProcessing** from **ExtensionAbstract** :

```
public function registerExtensionProcessing()
{
}
```

Now we need to register our Shortcode by using the "**addShortcode()**" method of the **ShortCodes** class. This method requires two parameters: a) the name of the Shortcode b) callback function that will be called every time the Shortcode is being used In our case, we'll call it our Shortcode a "Googlemap" and the method will have this signature:

```
$this->shortcodes->addShortcode( 'googlemap', function ($attributes, $content,
$tagName) use ($config) {
```

As mentioned earlier, because we're using a "callback" function, we could make the Shortcode very complex and sophisticated. But this maybe next time 😊 Our "addShortcode" method will look like this:

```
public function registerExtensionProcessing()
{
$config = $this->ts->getConfig('googlemap');
// direct load to allow Sanitizer to change 'allowchart'

$this->shortcodes->addShortcode('googlemap', function ($attributes, $content,
$tagName) use ($config) {
    $xoops = Xoops::getInstance();
    $defaults = ['width' => 640,
                'height' => 480,
                'src' => "];
    $cleanAttributes = $this->shortcodes->shortcodeAttributes( $defaults, $attributes);
    $width = $cleanAttributes['width'];
    if (preg_match('/[0-9]{1}$/', $width)) {$width .= 'px';}
    $height = $cleanAttributes['height'];
    if (preg_match('/[0-9]{1}$/', $height)) {
        $height .= 'px';
    }
    $src = $cleanAttributes['src'];
    $newContent = "];

return $newContent;
});
}
```

Once tested that it works, we'll be able to post directly a Google Map in any article of News or Publisher module, or any other XOOPS module, by just using the Shortcode "**[googlemap]**" Of course, we need to provide more info to it, so in the end it could look like this:

```
[googlemap width="600" height="300" src=
```

"http://maps.google.com/maps?q=Heraklion,+Greece&hl=en&ll=35.327451,25.140495&spn=0.233326,0.445976&sll=37.0625,-95.677068&sspn=57.161276,114.169922&oq=Heraklion&hnear=Heraklion,+Greece&t=h&z=12"]

And that's how the result of this Shortcode looks in Publisher module in XOOPS 2.6.0: **WHAT NEXT?** I've got so excited about the Shortcodes in XOOPS, that I have experimented yesterday with few other Shortcodes: - **Google Charts - Country Flags - QR Code - DateTime widget** For example, to add a flag to any content, we can now place just these Shortcodes:

[countryflag]

which will show us a default flag with default size (I set it to US and 64px) but it's up to you what you choose. The standard flag sizes are: 16, 32, 64. Or you can specify manually what country and what size of the flag you want to have:

[qrcode] [countryflag size=64 country="CA"]

This will create a Canadian flag 64px long. Countries are being selected using standard two letter country codes, e.g. US for USA, FR for France, DE for Germany, etc. Please note that you can also do the same programmatically, and to do so, the best way is to go install in XOOPS 2.6.0 the "Codex" module and check out several of the new additions to XOOPS 2.6.0. By using this Shortcode for Google Chart:

[chart charttype="pie" title="Example Pie Chart" data=

"41.12,32.35,21.52,5.01" labels=

"First+Label|Second+Label|Third+Label|

Fourth+Label" background_color="FFFFFF" colors=

"D73030,329E4A,415FB4,DFD32F" size="450x180"]

we can have this chart: And we can also create Shortcodes for XOOPS internal features, e.g. the QR Code for the current page:

[qrcode]

, and the previously mentioned country flag: Thanks to Richard, the Shortcodes are opening some very cool opportunities for XOOPS. There are plenty of Shortcodes on WordPress, some of them very cool, so feel free to convert them to XOOPS, and share with us. **To learn more how to create Shortcodes**, you can do following: a) study the refactored by Richard TextSanitizer classes which are basically Shortcodes themselves b) study some of the Shortcodes from other CMS, like WordPress and try to convert them to XOOPS **If you're interested, I have placed the five Shortcodes that I was experimenting with yesterday, in my Github repository.** Feel free to fork them and contribute back 🙌 Clearly, [with all the exciting work being done by Richard on Core](#), and [by Eduardo merging RM Common Utilities into Core](#), the next version of XOOPS will be very, very cool! 😊 **Viva XOOPS!** 😊

Richard continues to doing amazing changes to XOOPS 2.6.0. The latest addition are "Shortcodes", which are probably best known from WordPress. I was actually hoping for it for a long, long time, and I was very happy to see that Richard has added it to XOOPS! YES!!! 😊

It was possible thanks to refactoring of our old TextSanitizer. You can read more details from [Richard's info on GitHub](#)

Quote:

MyTextSanitizer has become Xoops\Core\Text\Sanitizer.

Highlights:

- Single unified configuration file

[xoops_data/configs/system_sanitizer_prefs.yml](#)

- Extensions (i.e. YouTube) are now a SanitizerComponent, a base class that also includes Filters (i.e. Xss)

- Each SanitizerComponent auto-configures, and configuration is automatically updated with new additions

- Custom components can be added via response to `core_sanitizer_configuration_defaults` event

- ShortCodes (WordPress style) are now the basis of XoopsCode and extensions

- Custom ShortCodes can be added via response to

[core_sanitizer_shortcodes_add](#) event, or by directly accessing the ShortCodes engine with

`Sanitizer::getShortCodes()`

- All built-in components and shortcodes can be overridden with same named custom replacements

- Some extensions have updated syntax. For example:

[youtube=640,385]https://www.youtube.com/watch?v=JxS5E-kZc2s[/youtube]

can now be:

[youtube url="JxS5E-kZc2s" width="640" height="385" /]

(It also can use virtually any known YouTube URL as the url attribute.)

- Another example is SoundCloud, which now can directly support the "Wordpress code" option offered in the Share/Embed dialog on soundcloud.com.

The old styles are still accepted for compatibility with existing content, and not all extensions have conversions to new styles, yet. There is more work to be done, but the backing code is now in place.

What does it mean for XOOPS developers? We'll be able now to simplify a lot of activities around designing the best looking Websites for our users and customers! 😊

Also our users will be able to add tons of cool features without any programming! 😊

For those of you who never worked with ShortCodes, let's provide a short overview and a little tutorial. The basic classes that we'll be using, are the TextSanitizer classes refactored by Richard:

Currently we have following classes derived from **FilterAbstract** :

- Censor
- Clickable
- Quote
- SyntaxHighlight
- TextFilter
- Xss

and following classes derived from **ExtensionAbstract**:

- Flash
- Iframe
- Image
- Mms

- Mp3
- Rtsp
- SoundCloud
- UnorderedList
- Wiki
- Wmp
- XoopsCode
- YouTube

For our purposes we'll be using mainly the **ExtensionAbstract** class. Let's see now how can we use it to create ShortCodes for **Google Charts** and **Google Maps** ([Read more-->](#))

But before we go there, lets provide a short overview of **ShortCodes** **ShortCodes** are nothing different than macros or scripts that you create to repeating the some activity over and over again. They can be very simple, and they can be very complex. Some of them are true code snippets, with no direct interaction, and some can be quite sophisticated, with visual GUI that helps you to select its options. There are also different ways to interact with Shortodes: a) simple ShortCodes without any attributes, using basically a single command, e.g. our BBcodes are simple Shortcodes that we use by placing [**b**] and [/**b**] around text to make it bold b) more complex ShortCodes with Attributes, like the YouTube example presented by Richard above:

```
[youtube url="JxS5E-kZc2s" width="640" height="385" /]
```

Creating a Shortcode for Google Maps After I saw yesterday Richard's announcement about the Shortcodes in XOOPS, I've got very excited and decided to explore them. In order to create a Shortcode, we'll be using the **ExtensionAbstract** class. We create a new file in the /Extensions folder called "Googlemap.php". We'll start with:

```
class Googlemap extends ExtensionAbstract
{
}
```

A Google map has just very basic features, like width, height, etc. So as first, we can create a default configuration for it. This will also make it easier for us, as we won't have to then fill out all the options every time we use it:

```
protected static $defaultConfiguration = [
'enabled' => true,
'template' => "",
'clickable' => true, // Click to open a map in a new window in full size
'resize' => true, // Resize the map down to max_width set below
'max_width' => 640, // Maximum width of a map displayed on page
'max_height' => 480, // Maximum width of a map displayed on page
'allowmap' => true, // true to allow maps, false to force links only
];
```

Of course, you can decide for yourself which options do you want to include and which one not. As next step, we need to register our extension with the Sanitizer. This is being done by overriding the abstract method **registerExtensionProcessing** from **ExtensionAbstract** :

```
public function registerExtensionProcessing()
{
}
```


Now we need to register our Shortcode by using the **addShortcode()** method of the **ShortCodes** class. This method requires two parameters: a) the name of the Shortcode b) callback function that will be called every time the Shortcode is being used In our case, we'll call it our Shortcode a "Googlemap" and the method will have this signature:

```
$this->shortcodes->addShortcode( 'googlemap', function ($attributes, $content, $tagName) use ($config) {
```

As mentioned earlier, because we're using a "callback" function, we could make the Shortcode very complex and sophisticated. But this maybe next time 😊 Our "addShortcode" method will look like this:

```
public function registerExtensionProcessing()
{
$config = $this->ts->getConfig('googlemap');
// direct load to allow Sanitizer to change 'allowchart'

$this->shortcodes->addShortcode('googlemap', function ($attributes, $content, $tagName) use ($config) {
    $xoops = Xoops::getInstance();
    $defaults = ['width' => 640,
                'height' => 480,
                'src' => ""];
    $cleanAttributes = $this->shortcodes->shortcodeAttributes( $defaults, $attributes);
    $width = $cleanAttributes['width'];
    if (preg_match('/[0-9]{1}$/', $width)) {$width .= 'px';}
    $height = $cleanAttributes['height'];
    if (preg_match('/[0-9]{1}$/', $height)) {
        $height .= 'px';
    }
    $src = $cleanAttributes['src'];
    $newContent = "";

    return $newContent;
});
}
```

Once tested that it works, we'll be able to post directly a Google Map in any article of News or Publisher module, or any other XOOPS module, by just using the Shortcode "**[googlemap]**" Of course, we need to provide more info to it, so in the end it could look like this:

```
[googlemap width="600" height="300" src=
"http://maps.google.com/maps?q=Heraklion,+Greece&hl=en&ll=35.327451,25.140495&spn=0.233326,0.445976& sll=37.0625,-95.677068&sspn=57.161276,114.169922& oq=Heraklion&hnear
=Heraklion,+Greece&t=h&z=12"]
```

And that's how the result of this Shortcode looks in Publisher module in XOOPS 2.6.0: **WHAT NEXT?** I've got so excited about the Shortcodes in XOOPS, that I have experimented yesterday with few other Shortcodes: - **Google Charts** - **Country Flags** - **QR Code** - **DateTime widget** For example, to add a flag to any content, we can now place just these Shortcodes:

[countryflag]

which will show us a default flag with default size (I set it to US and 64px) but it's up to you what you choose. The standard flag sizes are: 16, 32, 64. Or you can specify manually what country and what size of the flag you want to have:

[qrcode] [countryflag size=64 country="CA"]

This will create a Canadian flag 64px long. Countries are being selected using standard two letter country codes, e.g. US for USA, FR for France, DE for Germany, etc. Please note that you can also do the same programmatically, and to do so, the best way is to go install in XOOPS 2.6.0 the "Codex" module and check out several of the new additions to XOOPS 2.6.0. By using this Shortcode for Google Chart:

```
[chart charttype="pie" title="Example Pie Chart" data=
"41.12,32.35,21.52,5.01" labels=
"First+Label|Second+Label|Third+Label|
Fourth+Label" background_color="FFFFFF" colors=
"D73030,329E4A,415FB4,DFD32F" size="450x180"]
```

we can have this chart: And we can also create Shortcodes for XOOPS internal features, e.g. the QR Code for the current page:

[qrcode]

, and the previously mentioned country flag: Thanks to Richard, the Shortcodes are opening some very cool opportunities for XOOPS. There are plenty of Shortcodes on WordPress, some of them very cool, so feel free to convert them to XOOPS, and share with us. **To learn more how to create Shortcodes**, you can do following: a) study the refactored by Richard TextSanitizer classes which are basically Shortcodes themselves b) study some of the Shortcodes from other CMS, like WordPress and try to convert them to XOOPS **If you're interested, I have placed the five Shortcodes that I was experimenting with yesterday, in [my Github repository](#)**. Feel free to fork them and contribute back 🙌 Clearly, [with all the exciting work being done by Richard on Core](#), and [by Eduardo merging RM Common Utilities into Core](#), the next version of XOOPS will be very, very cool! 😊 **Viva XOOPS!** 😊