

## Magento 2 : Ajouter Une Validation Du Numéro De Téléphone

15 décembre 2021 / Par herve / Magento

*Cet article est assez ancien, malgré toute l'attention que j'apporte à mes contenus il est possible que celui-ci ne soit plus d'actualité. N'hésitez pas à me le signaler si nécessaire via le [formulaire de contact](#).*

Nous allons voir comment ajouter facilement une règle de validation spécifique sur le numéro de téléphone de l'adresse client sur les formulaires suivants :

- Ajout / Édition d'adresse dans le compte client
- Ajout d'adresse dans le tunnel de commande.

Pour faire cela nous allons créer un module **CustomerPhoneValidation** ( les sources complètes sont sur github et le lien disponible en fin d'article  
Cet article est basé sur la version 2.4.1

### Fonctionnement général

Les données des formulaires magento sont validées via l'objet javascript « validator » de magento, qui utilise ensuite la librairie jquery validate.

L'objet « Validator » dispose d'un certains nombre de règles de base qui sont visibles dans le fichier : *lib/web/mage/validation.js*

Il est possible assez facilement via les mixins d'ajouter nos propres règles de validation.

L'environnement de validation de données utilisées dans le compte clients et dans le tunnel de commandes étant différents il faudra créer 2 mixins.

Voici donc comment ajouter la nouvelle règle de validation

Cette règle vérifiera que le format du téléphone est au format français ( 10 chiffres commençant par 0 )

Pour commencer créer le fichier *view/frontend/requirejs-config.js* avec le contenu suivant qui va les déclarer.

```
var config = {
  config: {
    mixins: {
      //Ajout de la mixin pour la validation générale
      'mage/validation': {
        'Hhennes_CustomerPhoneValidation/js/validation-mixin': true
      },
    },
  },
}
```

```

//Ajout de la mixin pour le tunnel de commande
'Magento_Ui/js/lib/validation/validator': {
    'Hhennes_CustomerPhoneValidation/js/validator-mixin': true
}
}
}
}
}
}

```

Nous pouvons ensuite créer les 2 fichiers de mixins qui déclarent la nouvelle fonction de validation.

Validation générale dans le fichier `view/frontend/web/js/validation-mixin.js`

```

define(['jquery'], function($) {
    'use strict';
    return function() {
        $.validator.addMethod(
            'validate-custom-phone',
            function(value, element) {
                return $.mage.isEmptyNoTrim(value) || /^0[1-9][0-9]{8}$/.test(value);
            },
            $.mage.__('Please enter a valid french phone number.')
        );
    };
});

```

Puis validation pour les Ui component dans le fichier `view/frontend/web/js/validator-mixin.js`

```

define([
    'jquery',
], function ($) {
    'use strict';
    return function (validator) {
        validator.addRule(
            'validate-custom-phone',
            function (value, params, additionalParams) {
                return $.mage.isEmptyNoTrim(value) || /^0[1-9][0-9]{8}$/.test(value);
            },
            $.mage.__('Please enter a valid french phone number.')
        );
        return validator;
    };
});

```

A présent notre nouvelle validation est disponible partout en ajoutant la classe css `validate-custom-phone`

[Ajout sur le formulaire d'ajout / édition d'adresse.](#)

Le formulaire de gestion des adresses est dans le fichier `vendor/magento/module-customer/view/frontend/templates/address/edit.phtml`  
Mais le téléphone a son propre template dans le fichier `/vendor/magento/module-customer/view/frontend/templates/widget/telephone.phtml`

```
</label>
<div class="control">
    <input type="tel"
        name="telephone"
        id="telephone"
        value="<?=$block->escapeHtmlAttr($block->getTelephone()) ?>"
        title="<?=$block->escapeHtmlAttr(__('Phone Number')) ?>"
        class="input-text <?=$block->escapeHtmlAttr(
            $block->getAttributeValidationClass( attributeCode: 'telephone' )
        ) ?>"
    >
</div>
```

Pour l'ajouter sur le formulaire d'ajout d'édition d'adresse, il y'a ensuite plusieurs possibilités :

- Ajouter manuellement la classe `validate-custom-phone` dans l'attribut `class` de l'input
- Ajouter la classe via la programmation dans la fonction `getAttributeValidationClass`

On va suivre cette 2 ème option si on ne veut pas toucher au template.

La fonction `getAttributeValidationClass` appelle ensuite la fonction `\Magento\Customer\Helper\Address::getAttributeValidationClass` avec le code attribut `telephone`

On va donc faire un **plugin after** sur cette fonction ( cf. <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/plugins.html> )

Pour commencer on déclare le plugin dans le fichier `di.xml` du module

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd">
    <type name="Magento\Customer\Helper\Address">
        <plugin name="hhennes_customerphonevalidation"
            type="Hhennes\CustomerPhoneValidation\Plugin\Magento\Customer\Helper\AddressPlugin" />
    </type>
</config>
```

Puis voici la classe de gestion du plugin dans le fichier : `Plugin/Magento/Customer/Helper/AddressPlugin.php`

Avec le contenu suivant :

```
namespace Hhennes\CustomerPhoneValidation\Plugin\Magento\Customer\Helper;
```

```

use Magento\Customer\Helper\Address;

class AddressPlugin
{

    /** @var string Magento customer address "telephone" attribute code */
    const ATTRIBUTE_CODE_PHONE = 'telephone';

    /** @var string javascript custom validation class */
    const PHONE_VALIDATION_CLASS = 'validate-custom-phone';

    /**
     * Add a custom validation rule for "telephone" attribute code
     *
     * @param Address $subject
     * @param string $result
     * @param string $attributeCode
     * @return string
     */
    public function afterGetAttributeValidationClass(Address $subject, string $result, $attributeCode): string
    {
        if ($attributeCode == self::ATTRIBUTE_CODE_PHONE) {
            //Si nécessaire on peut rajouter d'autres conditions pour ajouter la classe de validation.
            $result .= " " . self::PHONE_VALIDATION_CLASS;
        }
        return $result;
    }
}

```

Avec ce code on en a fini pour la partie édition du compte.

On peut voir que notre validation spécifique est bien prise en compte

# Edit Address

## Contact Information

---

First Name \*

Last Name \*

Company

Phone Number \*

Please enter a valid french phone number.

Validation spécifique du numéro de téléphone dans le compte client

## Ajout dans le tunnel de commande

Dans le tunnel de commande, les configuration des attributs des champs sont mergés dans la fontction `\Magento\Checkout\Block\Checkout\AttributeMerger::merge`

On va donc faire un plugin sur cette fonction pour rajouter la règle de validation

Dans le fichier `etc/di.xml`

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd">
  <type name="Magento\Checkout\Block\Checkout\AttributeMerger">
    <plugin name="hhennes_customerphonevalidation"
      type="Hhennes\CustomerPhoneValidation\Plugin\Magento\Checkout\Block\AttributeMergerPlugin"/>
  </type>
</config>
```

```
</type>  
</config>
```

Puis voici la classe de gestion du plugin dans le fichier : *Plugin|Magento|Checkout|Block|AttributeMergerPlugin*

Avec le contenu suivant :

```
namespace Hhennes\CustomerPhoneValidation\Plugin\Magento\Checkout\Block;  
  
use Magento\Checkout\Block\Checkout\AttributeMerger;  
  
class AttributeMergerPlugin  
{  
  
    /** @var string Magento customer address "telephone" attribute code */  
    const ATTRIBUTE_CODE_PHONE = 'telephone';  
  
    /** @var string javascript custom validation class */  
    const PHONE_VALIDATION_CLASS = 'validate-custom-phone';  
  
    /**  
     * @param AttributeMerger $subject  
     * @param array $result  
     * @param array $elements  
     * @param string $providerName  
     * @param string $dataScopePrefix  
     * @param array $fields  
     * @return array  
     */  
    public function afterMerge(AttributeMerger $subject, array $result, $elements, $providerName, $dataScopePrefix, array $fields = []): array  
    {  
        //Ajout d'une règle de validation custom sur le téléphone pour les clients FR  
        if (array_key_exists(self::ATTRIBUTE_CODE_PHONE, $result)) {  
            $result['telephone']['validation'] = [  
                'required-entry' => true,  
                self::PHONE_VALIDATION_CLASS => true,  
            ];  
        }  
  
        return $result;  
    }  
}
```

Avec ce code on en a fini pour la partie tunnel de commande

On peut voir que notre validation spécifique est bien prise en compte

# Shipping Address




Street Address \*

Country \*

France 

State/Province

Please select a region, state or province. 

City \*

Zip/Postal Code \*

Phone Number \*



Please enter a valid french phone number.

Save In address book

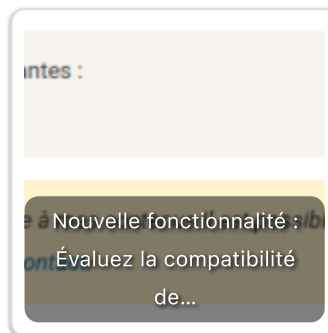
Cancel

Ship Here

Nous en avons à présent fini avec ce point 😊

Le module complet est disponible sur github : [https://github.com/nenes25/magento2\\_samplemodules/tree/master/Hhennes/CustomerPhoneValidation](https://github.com/nenes25/magento2_samplemodules/tree/master/Hhennes/CustomerPhoneValidation)

## Articles Relatif à Magento 2 : Ajouter une validation du numéro de téléphone



[← Article précédent](#)

[Article suivant →](#)

### 1 réflexion sur "Magento 2 : Ajouter une validation du numéro de téléphone"



**GAËTAN J**

18 AOÛT 2022 À 13 H 58 MIN

Merci Hervé pour cet article. Pas évident de trouver les bonnes infos sur ce sujet, même en anglais.

Merci d'avoir partagé.

[Répondre](#)

### Laisser un commentaire

Votre adresse e-mail ne sera pas publiée. Les champs obligatoires sont indiqués avec \*