

# How to customize UserHash generation



## Version compatibility

This recipe is compatible with **eZ Publish 5.2 / 2013.07**

- [Description](#)
- [Example](#)
- [Securing hash generation request](#)



Be sure to have read [Context aware HTTP cache](#) documentation before reading this recipe.

## Description

When user hash generation is requested, eZ Publish will create a **hashable User Identity** object.

One can add information to the Identity object making the resulted hash vary. This can be done by registering **Identity definers**.

For this, all you need to do is to declare a service with **ezpublish.identity\_definer** tag. Class for this service **must** implement **eZ\Publish\SPI\User\IdentityAware** interface.

## Example

### services.yml (inside a bundle)

```
parameters:
    my_identity_definer.class: Acme\TestBundle\Identity\MyDefiner

services:
    my_identity_definer:
        class: %my_identity_definer.class%
        tags:
            - { name: ezpublish.identity_definer }
```

```
<?php

namespace Acme\TestBundle\Identity;

use eZ\Publish\SPI\User\IdentityAware;
use eZ\Publish\SPI\User\Identity;

class MyDefiner implements IdentityAware
{
    public function setIdentity( Identity $identity )
    {
        // Here I can add information to $identity.
        // value MUST be scalar.
        $identity->setInformation( 'my_key', 'my_value' );
    }
}
```

## Securing hash generation request

By default, hash generation requests are granted for localhost (127.0.0.1, ::1, fe80::1).

If you want to enlarge the scope (e.g. if your Varnish server is not running on the same machine), you can override `canGenerateUserHash()` protected method in your main kernel class (mostly `EzPublishKernel`).