
TUTORIAL (TRANSLATE.CLASS)

6 DE FEBRERO DE 2012 (v. 0.9)

NETFLIE .

www.netflie.es

meteolive.netflie.es

.....

E info@netflie.es

.....

Índice de contenido

1.Introducción.....	4
2.Creando el contenido de las traducciones.....	4
3.Traduciendo nuestro sitio web.....	4
4.Nuestras traducciones en ficheros externos.....	5
5.Traducción automática.....	5

Índice de ilustraciones

1. Introducción

La clase Translate forma parte del paquete MeteoLive(<http://meteolive.netflie.es>) aunque se puede usar de manera independiente a éste.

Es una clase que nos permite conseguir una web multidioma sin la necesidad de crear un script por cada idioma, lo que nos ahorra tiempo, espacio y permite tener un sitio web con menos código y más organizado. A continuación se explica con un ejemplo su funcionamiento.

2. Creando el contenido de las traducciones

El contenido de la traducción son arrays planos, la clave será el texto original y el valor el texto a traducir:

array (CLAVE => VALOR)

```
// Array de traducción al español
$es = array(
    'ahora' => 'Ahora',
    'ultima-actualizacion' => 'Última actualización',
    'dios-existe-o-no' => 'Dios existe, o no.'
);

// Array de traducción al inglés
$en = array(
    'ahora' => 'Now',
    'ultima-actualizacion' => 'Last update',
    'dios-existe-o-no' => 'God exists, or not'
);
```

Recuerda que las **CLAVES** deben ser las mismas para todos los arrays de idiomas que creemos.

3. Traduciendo nuestro sitio web

Ahora queremos, por ejemplo, traducir nuestro script index.php

```
// Archivo: index.php

$es = array(
    'ahora' => 'Ahora',
    'ultima-actualizacion' => 'Última actualización',
    'dios-existe-o-no' => 'Dios existe, o no.'
);

// Array de traducción al inglés
$en = array(
    'ahora' => 'Now',
    'ultima-actualizacion' => 'Last update',
    'dios-existe-o-no' => 'God exists, or not'
);

// Creamos un objeto de tipo Translate e inicializamos la traducción al español
$translate = new Translate(array(
    'locale' => 'es',
    'content' => $es
));
```

```
// El texto que queremos traducir:
$translate->_('ultima-actualizacion'); // Devuelve "Última actualización"

// También podemos añadir nuevos idiomas
$translate->addTranslation(array(
    'locale' => 'en',
    'content' => $en
));

// Indicamos que ahora las traducciones se deben hacer al inglés
$translate->setLocale('en');

$translate->_('ultima-actualizacion'); // Devuelve "Last update"
```

4. Nuestras traducciones en ficheros externos

Con el método anterior teníamos el contenido de las traducciones y la lógica de nuestro sitio en el mismo script y esto es enfarragoso si tuviéramos, por ejemplo, mil líneas de claves distintas a traducir. Por eso es conveniente guardar las traducciones en un archivo externo.

```
// Archivo: languages/es.php

return array(
    'ahora' => 'Ahora',
    'ultima-actualizacion' => 'Última actualización',
    'dios-existe-o-no' => 'Dios existe, o no.'
);

// Archivo: index.php

$translate = new Translate(array(
    'locale' => 'es',
    'content' => 'languages/es.php'
));
```

La carpeta por defecto para los idiomas es /languages. Si tenemos nuestros archivos de idiomas en esa carpeta no hará falta indicar dónde se encuentra el contenido:

```
// Archivo: index.php

// esto
$translate = new Translate(array(
    'locale' => 'es',
    'content' => 'languages/es.php'
));

// y esto son equivalentes en el caso que nuestros idiomas estén en la carpeta /languages
$translate = new Translate(array(
    'locale' => 'es',
));

// además podemos hacer esto sin tener que llamar al método addTranslation()
$translate->setLocale('en');
```

5. Traducción automática

Por defecto la traducción automática está activada. Esto quiere decir que la página será traducida según los parámetros locales del navegador del visitante. Esto es útil para tener una web multiidioma flexible. Supongamos que tenemos en nuestra carpeta /languages los idiomas español e inglés. Con esta configuración cuando un visitante de Reino Unido visita nuestra página el contenido le será mostrado en inglés, en cambio cuando lo haga una persona de España la página estará en español.

```
// Archivo: index.php

$translate = new Translate();

$translate->_('ahora') // Devuelve la traducción si se ha encontrado
```

¿Qué sucede en este caso? Se leerá la lista de idiomas que tenga el visitante de nuestro sitio web en su navegador. Se buscará en la carpeta /languages si alguno de los idiomas está disponible y si es así, el primero que se encuentre será el usado por defecto. En el caso que no encontrara ningún idioma en esa carpeta entonces la traducción no se aplicaría y sería mostrado el valor literal de la clave. Para evitar esto podemos definir un lenguaje por defecto en el caso que no encontráramos ninguno de manera automática:

```
// Archivo: index.php

$translate = new Translate();
$translate->setLocale('es');

$translate->_('ahora') // Si se ha encontrado una traducción automática devuelve el valor de ésta,
si no lo intentará con la traducción indicada por defecto(en este caso el idioma 'es').
```

Podemos desactivar la traducción automática. Esto puede ser útil si queremos que algunas partes de nuestro sitio web sólo puedan estar disponibles en el idioma que nosotros indiquemos.

```
// Archivo: index.php

$translate->setAutomatic(false);
```

En este caso la traducción automática no estará disponible y sólo se aplicará la traducción que hayamos indicado con el constructor de la clase o el método *setLocale()*