PHP Barcode Generator(picqer)

Link: https://github.com/picqer/php-barcode-generator?tab=readme-ov-file

Es una librería que ofrece crear códigos de barra en imágenes de tipo SVG, PNG, JPG y HTML a partir de los estándares 1D más utilizados.

No funciona con ningún código de barra 2D, como los QR.

Solamente genera las 'barras' del código, cualquier otro agregado como letras o números debajo de las barras debe agregarse después. No obstante, ofrece una amplia gama de opciones para personalizar dichas barras, desde cambiar colores hasta el tipo del código creado.

Tener en cuenta:

Para usar esta librería es necesario composer y si se desea crear imágenes en PNG o JPG son necesarias las librerías GD library o Imagick. SVG y HTML no tienen dependencias.

Uso

Los códigos de barra se muestran según el tipo que sea y en que formato de imagen se crearan:

Primero se codifican los datos que se desean dentro de un objeto con el tipo (ej: code 128) del código que se desea obtener. Luego, se utiliza alguno de las representaciones disponibles (PNG, etc) para convertirlo en la imagen de los datos dentro del objeto.

Ej:

```
<?php

require 'vendor/autoload.php';

// se crea el objeto con el tipo code128

$barcode = (new Picqer\Barcode\Types\TypeCode128())->getBarcode('081231723897');


// Output the barcode as HTML in the browser with a HTML Renderer

// Muestra el codigo de barras como HTML en el buscador hecho con el render HTML .

$renderer = new Picqer\Barcode\Renderers\HtmlRenderer();
```

echo $renderer->render($barcode);

Resultado:

Como se dijo antes, esta librería cuenta con 4 formatos para entregar la imagen del código de barras y a su vez cada uno de ellos tiene métodos asociados para modificar los colores, así como el ancho y largo.

El método render () necesita el objeto de código de barras, el ancho y la altura. Para las imágenes JPG/PNG, solo obtiene un código de barras válido si da un ancho que es un factor del ancho del objeto de código de barras. Puede dar un número arbitrario como ancho y la imagen se escalará lo mejor posible, pero sin anti-aliasing, no será perfectamente válido. Los renderizados HTML y SVG pueden manejar cualquier ancho y altura, incluso decimales.

Estas son todas las opciones para cada método:

SVG

$ renderer = new Picqer \ BarCode \ Renderers \ Svgrenderer ();

- $ renderer-> setforeGroundColor ([255, 0, 0]); // Da un color rojo para las barras, el valor predeterminado es negro.
- $ renderer-> setbackgroundcolor ([0, 0, 255]); // Da un color azul para el fondo, el valor predeterminado es transparente.
- $ renderer-> setSvgType ($ renderer :: type_svg_inline); // Cambia la salida que se usa en línea dentro de los documentos HTML, en lugar de una imagen SVG independiente (predeterminada)
- $ renderer-> setSvgType ($ renderer :: type_svg_standalone); // Si desea forzar el valor predeterminado, cree una imagen SVG independiente
- $ renderer-> render ($ Barcode, 450.20, 75); // números con coma de soporte de ancho y altura

PNG y JPG

$ renderer = new Picqer \ BarCode \ Renderers \ Pngrenderer ();

- $ renderer-> setforeGroundColor ([255, 0, 0]); // Da un color para las barras, el valor predeterminado es negro.
- $ renderer-> SucedbackgroundColor ([0, 255, 255]); // Da un color para el fondo, el valor predeterminado es transparente (en PNG) o blanco (en JPG).
- $ renderer-> useGd (); // Si tienes Imagick y GD instalados, pero quieres usar GD
- $ renderer-> useImagick (); // Si tienes Imagick y GD instalados, pero quieres usar Imagick
- $ renderer-> render ($ código de barras, 5, 40); // Factor de ancho (cuántos píxeles de ancho cada barra es) y la altura en píxeles

HTML.

$ renderer = new Picqer \ BarCode \ Renderers \ Htmlrenderer ();

- $ renderer-> setforeGroundColor ([255, 0, 0]); // Da un color rojo para las barras, el valor predeterminado es negro.
- $ renderer-> setbackgroundcolor ([0, 0, 255]); // Da un color azul para el fondo, el valor predeterminado es transparente.

- $ renderer-> render ($ Barcode, 450.20, 75); // Soporte de ancho y altura.

HTML Dinámico:

$ renderer = new Picqer \ BarCode \ Renderers \ Dynamichtmlrenderer ();

- $ renderer-> setforeGroundColor ([255, 0, 0]); // Dar un color rojo para las barras, el valor predeterminado es negro.
- $ renderer-> setbackgroundcolor ([0, 0, 255]); // dar un color azul para el fondo, el valor predeterminado es transparente.
- $ renderer-> render ($ Barcode);

La librería cuenta con una amplia gama de tipos de códigos de barras, siendo los más comunes TYPE_CODE_128 y TYPE_CODE_39, pero existen muchos más:

- TYPE_CODE_32 (italian pharmaceutical code 'MINSAN')
- TYPE_CODE_39
- TYPE_CODE_39_CHECKSUM
- TYPE_CODE_39E
- TYPE_CODE_39E_CHECKSUM
- TYPE_CODE_93
- TYPE_STANDARD_2_5
- TYPE_STANDARD_2_5_CHECKSUM
- TYPE_INTERLEAVED_2_5
- TYPE_INTERLEAVED_2_5_CHECKSUM
- TYPE_CODE_128
- TYPE_CODE_128_A
- TYPE_CODE_128_B
- TYPE_CODE_128_C
- TYPE_EAN_2
- TYPE_EAN_5
- TYPE_EAN_8

- TYPE_EAN_13
- TYPE_ITF14 (Also known as GTIN-14)
- TYPE_UPC_A
- TYPE_UPC_E
- TYPE_MSI
- TYPE_MSI_CHECKSUM
- TYPE_POSTNET
- TYPE_PLANET
- TYPE_RMS4CC
- TYPE_KIX
- TYPE_IMB
- TYPE_CODABAR
- TYPE_CODE_11
- TYPE_PHARMA_CODE
- TYPE_PHARMA_CODE_TWO_TRACKS
- Entre otros…