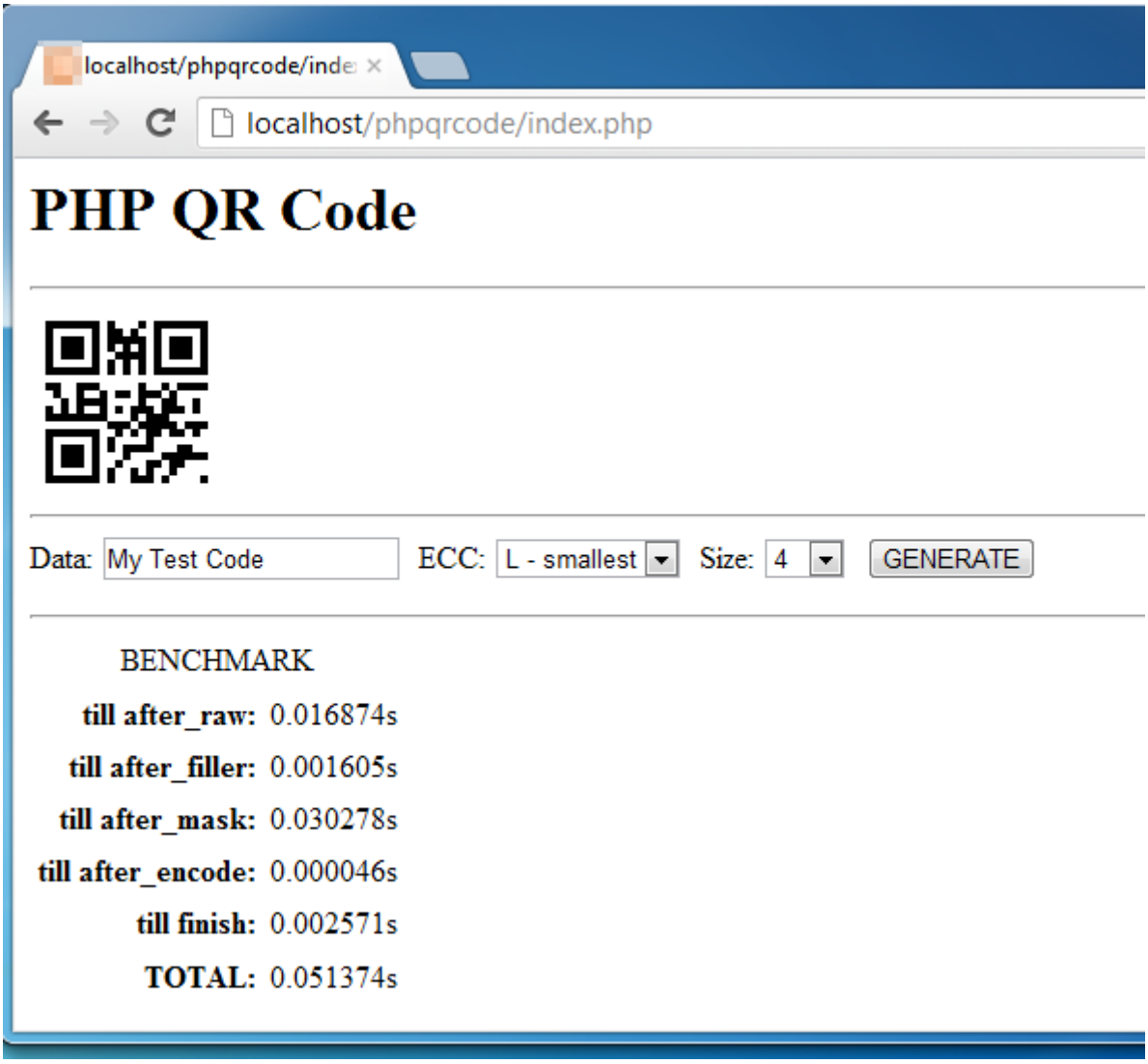
Qrcode con PHP (scarica qrcode da http://phpqrcode.sourceforge.net/)

Utilizzati ampiamente in varie applicazioni recenti, i codici QR possono essere visualizzati su lattine di cola, biglietti da visita, sushi bar e musei.

Il codice QR è una specifica di codice a barre bidimensionale che è stata inventata in Giappone. È brevettato. ma l'inventore, Denso Wave, ha scelto di non esercitarlo e ha lasciato lo standard aperto a beneficio di tutti. Da allora il codice è cresciuto in popolarità grazie alla sua capacità di includere molti dati in una singola immagine e la proliferazione di smartphone con app di scansione.

In questo articolo ti mostrerò come puoi facilmente generare codici QR dall'interno dell'applicazione PHP e condividere alcune idee su come e quando usarli, utilizzeremo PHP QR Code , una libreria scritta in PHP per generare codici QR e che non richiede dipendenze oltre l'estensione grafica standard GD2 per la creazione di immagini.



F. Sacco – qrcode – tratto da https://www.sitepoint.com/generate-qr-codes-in-php/

Generazione del tuo primo codice QR

Inizia scaricando la più recente libreria di codici QR PHP da GitHub. Presumo che tu l'abbia estratto con successo e puoi andare su http: // localhost / phpqrcode nel tuo ambiente di sviluppo per trovare la versione demo funzionante.

È possibile inserire qualsiasi testo nel campo dati che si desidera convertire in un'immagine del codice QR, come mostrato nella schermata seguente.

Se hai qualche problema a farlo funzionare, assicurati di aver installato PHP con l'estensione GD2, ricontrolla se necessario usando una pagina di informazioni PHP (phpinfo())

Crea un nuovo script PHP con il seguente codice:

```
<?php include "phpqrcode/qrlib.php";

// create a QR Code with this text and display it

QRcode::png("My First QR Code");
```

Vedi com'è semplice? Con solo due righe di codice ottieni un codice QR perfettamente valido per la tua applicazione. Le opportunità sono infinite! Ma aspetta, ovviamente questa non è la storia completa. La biblioteca ha più funzioni che vale la pena guardare.

Funzionalità della libreria di codici QR PHP

Per un esempio completo, prova questo codice:

```
<?php QRcode::png("http://www.sitepoint.com", "test.png", "L", 4, 4);
```

Il primo parametro specifica il testo o i dati che verranno codificati nell'immagine e verrà passato come una normale stringa. Il secondo parametro è il nome del file di output per l'immagine PNG generata, se presente. Il valore predefinito è un falso booleano, nel qual caso l'immagine viene scaricata sul browser.

Il terzo parametro è il livello di correzione degli errori per il codice a barre generato, passato come stringa di una sola lettera. Questo specifica la quantità di parole in codice dei dati (8 bit per parola in codice) che può essere ripristinata per un'immagine di codice QR distorta o danneggiata utilizzando l' algoritmo di correzione degli errori Reed-Solomon . Maggiore è il livello di correzione, minore è la capacità di dati del codice a barre per una determinata dimensione. Di seguito è riportata una tabella che associa i livelli alle loro percentuali di ripristino e alle costanti di stringa utilizzate quando si chiama `QRcode::png()` . (Ho compilato la tabella dall'articolo di Wikipedia sui codici QR e la firma del metodo nella libreria di codici QR PHP.)

Level	Restorable Codewords	PHP QR Code Parameter
Low	7%	L
Medium	15%	M
Quartile	25%	Q
High	30%	H

Il quarto parametro specifica la dimensione di ciascuno dei quadrati del codice a barre misurato in pixel. Ogni quadratino di codice (chiamato anche "pixel" o "moduli") è 4 × 4px. Il quinto parametro specifica il limite del

margine bianco attorno al codice a barre, misurato in quadrati di codice (ad es. Un margine di 16 pixel su ciascun lato per un quadrato di codice 4 × 4 pixel).

La libreria supporta l'esportazione di immagini PNG, SVG ed EPS ed è possibile produrre codici QR in uno di questi formati semplicemente cambiando il nome del metodo da `png()` a `svg()` o `eps()` e utilizzare l'estensione corretta per l'immagine generata nome del file.

Inoltre, puoi cambiare i colori di sfondo e di primo piano passandoli come parametri aggiuntivi:

```
 <?php

$backColor = 0xFFFF00;

$foreColor = 0xFF00FF;

// Create a QR Code and export to SVG

QRcode::svg("http://www.sitepoint.com", "test-me.svg", "L", 4, 4, false, $backColor,
$foreColor);
```

Il sesto parametro (falso nell'esempio sopra) sembra essere un parametro inutile. Dovrebbe essere vero per il salvataggio in un file e l'esportazione nel browser, ma semplicemente non ha funzionato per me dopo averlo verificato più volte, quindi mantienilo falso.

La libreria ha più funzionalità che puoi verificare se lo desideri, ad esempio la memorizzazione nella cache e l'analisi comparativa della generazione di immagini.

Ottenere le dimensioni del codice a barre finale

Per ottenere in anticipo la dimensione finale dell'immagine, ecco una semplice formula che può essere utilizzata (poiché l'immagine è un quadrato, dobbiamo solo calcolare una singola dimensione e l'altra sarà la stessa):

```
  Dimensione immagine (px) = (Pixel per modulo) × (Dimensione modulo + 8)
```

Dove indicato in precedenza, i pixel per modulo sono specificati nella chiamata del metodo come quarto parametro e la dimensione del modulo è selezionata da queste tabelle di dimensionamento dei codici a barre come segue:

1. Seleziona la colonna del tipo di stringa (bit di dati, numerico, alfanumerico, binario o Kanji). Questi specificano la lunghezza massima dei dati di tale tipo da impacchettare in un determinato codice a barre. In precedenza ho usato alfanumerico, ma se si utilizzano stringhe codificate UTF-8, è possibile che si stia utilizzando invece il tipo binario. Kanji è per il giapponese, ma non è stato testato dall'autore della biblioteca.
2. Scegli il livello desiderato di correzione degli errori e per la lunghezza della stringa trova il numero di versione minimo in grado di gestire almeno quel numero di caratteri. L'esempio utilizzava 24 o più caratteri di tipo alfanumerico a livello L, quindi il valore sarà la prima riga della versione 1.
3. Ottieni il modulo per la versione che scegli, qui sarà il modulo 21 × 21, dove la dimensione del modulo sarà 21. La libreria di codice QR PHP prende la versione successiva invece per più spazio per sicurezza, quindi salta un'altra .

Se calcoli la dimensione del modulo per la versione utilizzata per l'esempio, scoprirai che la dimensione dell'immagine prodotta dovrebbe essere:

```
Dimensione dell'immagine = 4 × (21 + 8) = 116 × 116px
```

Invece l'immagine generata è 132 × 132px. PHP QR Code ha preso la versione successiva (versione 2 anziché versione 1, o semplicemente modulo 25 × 25), quindi la dimensione generata effettiva sarà:

```
Dimensione dell'immagine = 4 × (25 + 8) = 132 × 132px
```

Usi comuni per i codici QR

L'applicazione più comune per i codici QR è quella di codificare gli URL dei siti Web, come quelli su una fan page di Facebook del tuo ultimo prodotto, della tua azienda, ecc. Le opzioni sono infinite. Io stesso lo uso sul mio biglietto da visita e codifico l'URL sul mio profilo LinkedIn.

I codici QR possono anche memorizzare numeri di telefono, vCard e indirizzi e-mail. Alcuni siti li mettono accanto agli articoli del blog per fungere da segnalibri.

Quando si tratta di utilizzare i codici QR, i tuoi unici limiti sono in realtà la capacità di dati del codice e lo spazio in cui lo visualizzerai.