Preface

This file contains important annotations for the 3rd party APIs that are used by the SmartqBundle, especially concerning **modifications** that have been made on them.

Read the respective section for the particular API carefully before updating, modifying or even using it and complete these annotations when updating or modifying the corresponding API.

Last changes: **03-05-2017 10:00:00**

SmartqBundle installation under Symfony

Place the Bundle Folder under the **src** folder (as you would do it with any custom bundle).

Add `new SmartqBundle\SmartqBundle()` in file **app/AppKernel.php** to the `$bundles`-array inside of the `registerBundles()` method.

Add
`- { resource: "@SmartqBundle/Resources/config/services.yml" }` to the imports-array at the top of file **app/config/config.yml**.

It is currently not necessary to add any routes to add any routes to the routing table and thus adding the bundle's routing.yml to the system's app/config/routing.yml. Anyway - For future reference - this would look like:

```
smartq:
    resource: "@SmartqBundle/Resources/config/routing.yml"
    prefix:   /
```

Bitgrave Barcode Generator

Folder

BGBarcodeGenModified

Origin

https://github.com/paterik/BGBarcodeGenerator
or
https://www.versioneye.com/php/bitgrave:barcode-generator/1.0.3
or
https://packagist.org/packages/bitgrave/barcode-generator

Modifications

Changes made in modules/Datamatrix.php (compare Datamatrix.MODIFIED vs. Datamatrix.ORIGINAL):

- reduced **Method lookAheadTest()** to simply giving always back "self::ENC_ASCII"
  (Changed on 02 May 2017 by Gregor Eggert)
  see source code

Usage

Standalone ...

For basic usage examples of the Bitgrave Barcode Generator API see source code in file
BGBarcodeGenModified/samples/index.php

Basic PHP example for a generation of a DataMatrix using only the API itself:

```
<?php

// adapt path as needed ... :
require_once( 'SmartqBundle/3rdParty/BGBarcodeGenModified/Base2DBarcode.php' );

$path = '/some/directory/path';

$fileName = 'DataMatrix_' . date( 'YmdHis' ); // or the like

$myDataMatrix = new \BG\Barcode\Base2DBarcode();
$myDataMatrix->savePath = $path;

// Note: all these characters work without Problems:
$text = <<<________EOT
This is some Text that shall be encoded into a DataMatrix.
it contains some special characters like Ä, Ö and Ü as well
as !"#$%&'()*+-./:;<=>?@[\]^_{|}~¡¢£¤¥¦§¨©ª«®º±²³µ·¹º»×,
ÀÁÂÃÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕØÙÚÛÝÞ, àáâãäåæçèéêëìíîïðñòóôõö÷ and
øùúûüýþÿ.
________EOT;

// ggf.: $text = str_replace( ... );

$text = utf8_decode( $text );

$fPath = $myDataMatrix->getBarcodeFilenameFromGenPath(
            $myDataMatrix->getBarcodePNGPath(
                $text, 'datamatrix' , $fileName , 4 , 4 , [0,0,0]
            )
        );
```

`$fPath` is the local system path to the image file which is a PNG file by default.

The next example does essentially do the same as the above but uses the SmartqStandalone\BarcodeService (where UTF-8-decoding and replacements are done transparently):

```
<?php

require_once( 'SmartqBundle/BarcodeService.php' ); // or similar

$barcodeSvc = new \SmartqStandalone\BarcodeService();

$text = <<<________EOT
This is some Text ...
________EOT;

$fPath = $barcodeSvc->textToBarcode( $text , 'datamatrix' , ['w'=>4,'h'=>4] );
```

For further options of method `textToBarcode()` please refer to the comments inside of the source code.

inside Symfony ...

Again the same example using Symfony's "service container":

```
<?php

$barcodeSvc =  $ServiceContainer->get('smartq.barcode');
// or inside a controller: $this->get('smartq.barcode');

$text = <<<________EOT
This is some Text ...
________EOT;

$fPath = $barcodeSvc->textToBarcode( $text , 'datamatrix' , ['w'=>4,'h'=>4] );
```

Generating a medication plan ("Mediplan") DataMatrix

The approch to this is in general identical for the Standalone and the Symfony version.

Standalone would start with:

```
require_once( 'SmartqBundle/BarcodeService.php' ); // or similar
$barcodeSvc = new \SmartqStandalone\BarcodeService();
```

Symfony would start with:

```
$barcodeSvc =  $ServiceContainer->get('smartq.barcode');
```

The rest could then look like:

```
$mpData = [
    'U'=>"B544B6976AB84E3498AA96D8E6FA29C1" ,
    'l'=>"de-DE" ,
    'v'=>"021" ,
    'P'=>[
        'b'=>"1936-12-13" ,
        'egk'=>"P123456789" ,
        'f'=>"Mustermann" ,
        'g'=>"Michaela"
    ] ,
    'A'=>[
        'c'=>"Am Ort" ,
        'e'=>"m.ueberall@mein-netz.de" ,
        'n'=>"Dr. Manfred Überall" ,
        'p'=>"04562-12345" ,
        's'=>"Hauptstraße 55" ,
        't'=>"2014-12-15" ,
        'z'=>"01234"
    ] ,
    'O'=>[] ,
    'S'=>[
        [
          'M'=>[
              [
                  'du'=>"1" ,
                  'i'=>"während der Mahlzeiten" ,
                  'm'=>"1" ,
                  'p'=>"4213974" ,
                  'r'=>"Bluthochdruck"
              ] ,
              [
                  'du'=>"1" ,
                  'i'=>"während der Mahlzeiten" ,
                  'm'=>"1" ,
                  'p'=>"6453174" ,
                  'r'=>"Bluthochdruck"
              ] ,
              [
                  'du'=>"1" ,
                  'i'=>"während der Mahlzeiten" ,
                  'p'=>"4129423" ,
                  'r'=>"art. Verschluss" ,
                  'v'=>"1"
              ] ,
              [
                  'du'=>"1" ,
                  'i'=>"nach der Mahlzeit" ,
                  'p'=>"1048871" ,
                  'r'=>"erhöhte Blutfette" ,
                  'v'=>"1"
```

```
            ]
          ]
        ] ,
        [
            'c'=>"416" ,
            'M'=>[
                [
                    'du'=>"p" ,
                    'i'=>"sub cutan" ,
                    'm'=>"20" ,
                    'p'=>"544757" ,
                    'r'=>"Diabetes" ,
                    'v'=>"10"
                ]
            ]
        ] ,
        [
            'c'=>"411" ,
            'M'=>[
                [
                    'du'=>"5" ,
                    'i'=>"akut" ,
                    'p'=>"4877970" ,
                    'r'=>"Herzschmerzen" ,
                    't'=>"max. 3"
                ] ,
                [
                    'du'=>"1" ,
                    'h'=>"1" ,
                    'i'=>"bei Bedarf" ,
                    'p'=>"2083906" ,
                    'r'=>"Schlaflosigkeit"
                ]
            ]
        ] ,
        [
            'c'=>"422" ,
            'X'=>[
                    't'=>"Bitte messen Sie Ihren Blutdruck täglich!"
                ]
        ]
    ]
];

$xml = $barcodeSvc->generateMediplanDMapXML( $mpData );

$out = $barcodeSvc->textToBarcode( $xml , 'datamatrix' , ['w'=>4,'h'=>4,'jpg'=>true] );
// please note: option "jpg" is only used to demonstrate the possibility to
// force the generation of (conversion to) a JPEG file instead of a PNG
```

The array `$mpData` corresponds to the structure of the **resulting XML** - in the upper example this would look like:

```
<MP U="B544B6976AB84E3498AA96D8E6FA29C1" l="de-DE" v="021">
    <P b="1936-12-13" egk="P123456789" f="Mustermann" g="Michaela"/>
    <A c="Am Ort" e="m.ueberall@mein-netz.de" n="Dr. Manfred Überall" p="04562-12345"
     s="Hauptstraße 55" t="2014-12-15" z="01234"/>
    <O/>
    <S>
        <M du="1" i="während der Mahlzeiten" m="1" p="4213974" r="Bluthochdruck"/>
        <M du="1" i="während der Mahlzeiten" m="1" p="6453174" r="Bluthochdruck"/>
        <M du="1" i="während der Mahlzeiten" p="4129423" r="art. Verschluss" v="1"/>
        <M du="1" i="nach der Mahlzeit" p="1048871" r="erhöhte Blutfette" v="1"/>
    </S>
    <S c="416">
        <M du="p" i="sub cutan" m="20" p="544757" r="Diabetes" v="10"/>
    </S>
```

```
    <S c="411">
        <M du="5" i="akut" p="4877970" r="Herzschmerzen" t="max. 3"/>
        <M du="1" h="1" i="bei Bedarf" p="2083906" r="Schlaflosigkeit"/>
    </S>
    <S c="422">
        <X t="Bitte messen Sie Ihren Blutdruck täglich!"/>
    </S>
</MP>
```

(Actually the result will not contain line breaks and indentations which are added here to increase the comprehensibility.)

See also attached document **BMP_Anlage3.pdf** (a ready-made medication plan that matches the upper example can be found there on page 27).

As an alternative to the array you may build an e quivalent DOMDocument which you can pass to method `generateMediplanDMapXML()`. Or - of course - you can build the complete XML yourself and simply pass it directly to `textToBarcode()`, if that should be more convenient.