



# Guide to Developed FATOORA Compliant QR Code

E-invoicing (FATOORA) implementation in the  
Kingdom of Saudi Arabia

Nov 18th, 2021



# Content

## Item

- ● ● 1. ZATCA QR Code Specifications
- ● ● 2. What is a Tag - Length - Value (TLV)
- ● ● 3. How to create a TLV QR Code & Common Mistakes
- ● ● 4. Manual Decoding a TLV QR Code
- ● ● 5. Code Snippets
- ● ● 6. Representation of the QR Code Data Examples



# ZATCA QR Code Specifications

## Structure of the QR code For Electronic Tax Invoices

It is mandatory to generate and print QR code encoded in Base64 format with up to 500 characters that must contain the fields specified in the below table as per Annex (2) of the Controls, Requirements, Technical Specifications and Procedural Rules for Implementing the Provisions of the E-Invoicing Regulation.

The QR code fields shall be encoded in Tag-Length-Value (TLV) format with the tag values specified in the "Tag" column of the adjacent table.

The TLV encoding shall be as follows:

- Tag: the tag value as mentioned above stored in one byte
- Length: the length of the byte array resulted from the UTF8 encoding of the field value. The length shall be stored in one byte.
- Value: the byte array resulting from the UTF8 encoding of the field value.

**Source:** [https://zatca.gov.sa/ar/E-Invoicing/SystemsDevelopers/Documents/20210528\\_ZATCA\\_Electronic\\_Invoice\\_Security\\_Features\\_Implementation\\_Standards\\_vShared.pdf](https://zatca.gov.sa/ar/E-Invoicing/SystemsDevelopers/Documents/20210528_ZATCA_Electronic_Invoice_Security_Features_Implementation_Standards_vShared.pdf)



# ZATCA QR Code Specifications

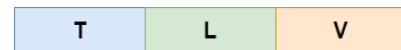
## Field Definition for the QR Code

Description	Tag	Due Date
Seller's name	1	4th Dec 2021
VAT registration number of the seller	2	4th Dec 2021
Time stamp of the invoice (date and time)	3	4th Dec 2021
Invoice total (with VAT)	4	4th Dec 2021
VAT total	5	4th Dec 2021
Hash of XML invoice	6	1st Jan 2023
ECDSA signature	7	1st Jan 2023
ECDSA public key	8	1st Jan 2023
For Simplified Tax Invoices and their associated notes, the ECDSA signature of the cryptographic stamp's public key by ZATCA's technical CA	9	1st Jan 2023

**Source:** [https://zatca.gov.sa/ar/E-Invoicing/SystemsDevelopers/Documents/20210528\\_ZATCA\\_Electronic\\_Invoice\\_Security\\_Features\\_Implementation\\_Standards\\_vShared.pdf](https://zatca.gov.sa/ar/E-Invoicing/SystemsDevelopers/Documents/20210528_ZATCA_Electronic_Invoice_Security_Features_Implementation_Standards_vShared.pdf)



## What is a Tag - Length - Value (TLV)

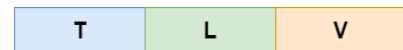


### What is a TLV (Tag - Length - Value) file format and how is it constructed?

- QR code is the base64 encoded TLV (Tag, Length, Value)
- Type/Tag-Length-Value (TLV) is an encoding scheme used in many communication protocols to encode data. A TLV-encoded message has a defined structure which consists of 3 sections/parts, see Figure (1). Those are:
  - Code of the message type (T) - 1 Byte
  - Message value length (L) - 1 Byte
  - Message value itself. (V) - Variable
- The Tag/Type and Length are of fixed sizes of 1 bytes while the value has a variable size.
- As the general idea behind encoding is to transform abstract data into a stream of bits, using TLV, there are different sets of encoding rules that can be used according to the Abstract Syntax Notation Version 1 (ASN.1). We are using a simple version of Basic Encoding Rules (BER).



## How to create a TLV QR Code? (1)

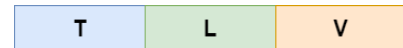


What is a TLV (Tag - Length - Value) file format and how is it constructed?

Description	Tag	Length	Value	Hex
Seller's name	1	12	Bobs Records	010c426f6273205265636f726473
VAT registration number	2	15	310122393500003	020F333130313232333933353030303033
Time stamp	3	20	2022-04-25T15:30:00Z	0314323032322d30342d32355431353a33303a30305a
Invoice total (with VAT)	4	7	1000.00	0407313030302e3030
VAT total	5	6	150.00	05063135302e3030



## How to create a TLV QR Code? (2)



What is a TLV (Tag - Length - Value) file format and how is it constructed?

Hex Representation:

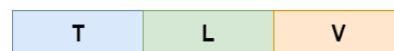
```
010c426f6273205265636f726473020f3331303132323339333530303030330314323032322d-30342d32355431353a33303a30305a0407313030302e303005063135302e3030
```

Base 64:

```
AQxCb2JzIFJIY29yZHMCDzMxMDEyMjM5MzUwMDAwMwMUMjAyMiOwNCOyNVQxNTozMDowM-FoEBzEwMDAuMDAFBjE1MC4wMA==
```



## Common Mistakes



### What are some of the common mistakes we have seen with TLV Creation?

- The **Tag and Length** are **binary values** of exactly one byte, therefore represented as Hex, 21 (as a length) is 15 and 46 is 2E
- The **Value must also be converted to binary** before adding to the byte array therefore when adding a value like Bobs Basement Records to the array it would be represented as 426F627320426173656D656E74205265636F726473 in Hex and الجواهربي العربي would be represented as 62764462c64862764763164a2062764463963162864a
- There should be **no padding or separators** between the TLV sets in binary array, the binary bytes should follow each other concurrently, 0115 42 6f 62 73 20 42 61 73 65 6d 65 6e 74 20 52 65 63 6f 72 64 73 02 0f 31 30 30 30 32 35 39 30 36 37 30 30 30 30 33 03 14 32 30 32 32 2d 30 34 2d 32 35 54 31 35 3a 33 30 3a 30 30 5a 04 0a 32 31 30 30 31 30 30 2e 39 39 05 09 33 31 35 30 31 35 2e 31 35, the decoding is done by reading the length of each Tag from the second byte in each TLV pair
- In order to encode Arabic Text into binary it is important to use **UTF8 Encoding**





## Manual Decoding a TLV QR Code (1)

T

L

V

### Extract the QR Code and Convert to Hex using publicly available tools

1. Example Base64 Encode QR Code, extracted using QR Code reader (i.e. Mobile Phone):

```
ARVCb2JzIEJhc2VtZW50IFJIY29yZHMCDzEwMDAyNTkwNjcwMDAwMwMUMjAyMi0wNCOyNVQxN-TozMDowMFMFoECjlxMDAxMDAuOTkFCTMxNTAxNS4xNQ==
```

2. Decode this to a hex representation, this can be done at the following site: [cryptii](https://cryptii.com/convert/base64-to-hex)

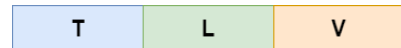
```
0115 42 6f 62 73 20 42 6173 65 6d 65 6e 74 20 52 65 63 6f 72 64 73 02 0f 31 30 30 30 32 35 39 30 36  
37 30 30 30 30 33 03 14 32 30 32 32 2d 30 34 2d 32 35 54 31 35 3a 33 30 3a 30 30 5a 04 0a 32 31 30  
30 31 30 30 2e 39 39 05 09 33 31 35 30 31 35 2e 31 35
```

3. Hex Representation can be read by a TLV reader, i.e. : [emvlab](https://emvlab.com/)

4. UTF8 Encoded values can be read using an online tool, i.e. : [onlineutf8tools](https://onlineutf8tools.com/)



## Manual Decoding a TLV QR Code (2)



Using a TLV Decoder to split the record shows the Hex Values, these can then be decoded using a hex to string decoder

Tag	Hex Value	Hex to String
01 Sellers Name	426F627320426173656D656E74205265636F726473	Bobs Basement Records
02 VAT Number	313030303235393036373030303033	100025906700003
03 Time Stamp	323032322D30342D32355431353A33303A30305A	2022-04-25T15:30:00Z
04 Invoice Amount	323130303130302E3939	2100100.99
05 VAT Amount	3331353031352E3135	315015.15



# Javascript - nodeJS

## How to create a QR Code in Java

This function takes in 2 args:

- tagNum: Tag Number
- tagValue: Value of Message

you convert the tagNum into byte array



get the length of the value and you convert it into byte array



convert the value into byte array



Once we have 3 byte arrays, we concat them into 1 byte array representing our TLV Message

```
function getTLVForValue(tagNum, tagValue) {  
    var tagBuf = Buffer.from([tagNum], 'utf8');  
  
    tagValueLenBuf = Buffer.from([tagValue.length], 'utf8')  
  
    var tagValueBuf = Buffer.from(tagValue, 'utf8');  
  
    var bufsArray = [tagBuf, tagValueLenBuf, tagValueBuf]  
  
    return Buffer.concat(bufsArray);  
}
```



## Javascript - nodeJS Cont.

You do the previous steps for each of the Tags you want to add to the QR code. For example, here we have sellerName, VatReg, etc.

you concat those buffs into a single array representing the QR code (see 1, 2)



Aftwards, you encode into Base64 (see 3)

```
// 1. Seller Name
var sellerNameBuf = getTLVForValue("1", "salah hospital");

// 2. VAT Registration
var vatRegistrationNameBuf = getTLVForValue("2", "31234567890123");

// ....
```

```
var tagsBufsArray = [sellerNameBuf,
  vatRegistrationNameBuf,
  (1) timestampBuf, taxTotalNameBuf, vatTotalBuf,
  hashedXmlBuf, keyBuf, signatureBuf];

var qrCodeBuf = Buffer.concat(tagsBufsArray); (2)

var qrCodeB64 = qrCodeBuf.toString('base64'); (3)
```



## Dart

Use the `ByteBuffer` class to add each segment of each TLV message i.e. 3 per message.

We repeat for each message we want to add to the QR Code

```
ByteBuffer bytesBuilder = ByteBuffer();

// 1. Seller Name
bytesBuilder.addByte(1);
List<int> sellerNameBytes = utf8.encode(sellerName);
bytesBuilder.addByte(sellerNameBytes.length);
bytesBuilder.add(sellerNameBytes);

// 2. VAT Registration
bytesBuilder.addByte(2);
List<int> vat_registrationBytes = utf8.encode(vat_registration);
bytesBuilder.addByte(vat_registrationBytes.length);
bytesBuilder.add(vat_registrationBytes);

// .....|
```



## Dart Cont.

```
Uint8List qrCodeAsBytes = bytesBuilder.toBytes();  
final Base64Encoder b64Encoder = Base64Encoder();  
return b64Encoder.convert(qrCodeAsBytes);
```

Once all messages added to the builder, you convert it into bytes (see 1) which gives you Uint8List (Darts way of byte []), then you encode the list into Base64 using an instance of the Base64Encoder class (see 2)



## Representation of the QR code Data Examples

Hyperlink to a Website  
(incorrect)

<https://zatca.gov.sa/en/pages/default.aspx>



Data in Text Format  
(incorrect)

Seller's name Bobs Records  
VAT registration number 310122393500003  
Time stamp 2022-04-25T15:30:00Z  
VAT total 1000.00  
VAT total 150.00



Hyperlink to the invoice online  
(incorrect)

[https://mcusercontent.com/a90cefeb037ed376188308d34/files/2ca406b2-8627-66d9-45a4-94d186a-4f3a5/User\\_Manual\\_Software\\_Development\\_Kit\\_SDK\\_.01.pdf](https://mcusercontent.com/a90cefeb037ed376188308d34/files/2ca406b2-8627-66d9-45a4-94d186a-4f3a5/User_Manual_Software_Development_Kit_SDK_.01.pdf)



Empty or Random Values  
(Incorrect)

2000555663314



TLV Base64 string (correct)

AQxCb2JzIFJlY29yZHMCDzMxMDEyMjM-  
5MzUwMDAwMwMUMjAyMiOwNCOyN-  
VQxNTozMDowMFoEBzEwMDAuMDAFB-  
jE1MC4wMA==





## SDK validation (QR related)

<https://zatca.gov.sa/en/E-Invoicing/SystemsDevelopers/ComplianceEnablementToolbox/Pages/DownloadSDK.aspx>

Command line	Results
fatoorah -v	To display (Version)
fatoorah -h	Help window
fatoorah validateqr -qr	Validate QR code structure
fatoorah generate -f (Invoicename.xml) -q	Generate compliant QR code

فاتورة Fatoora

العربية

search ...

Login Signup

2030

About Us E-Services Rules and Regulations E-Invoicing Integrated Tariffs Media Center Knowledge Center Contact Us

Home -> E-Invoicing -> Systems Developers -> Compliance and Enablement Toolbox

[Download SDK](#) Supporting Document

The Compliance and Enablement Toolbox SDK User Manual provides guidance with regards to the functional and technical aspects of the Compliance and Enablement Toolbox SDK such as what is the SDK, how to use the SDK and how to install it.

[Download User Manual](#)

Zakat, Tax and Customs Authority "ZATCA" has developed "the SDK toolkit" to help Persons subject to E-Invoicing Regulation and developers of technical solutions verify the compliance of generated E-Invoices, credit and debit notes to the requirements of the E-Invoicing Regulation.

When using the SDK Toolkit, persons subject to E-Invoicing Regulation and developers of technical solutions must consider the following:

- A. Invoice files are considered compliant with the E-Invoicing Regulation only once they have passed the verification process which is carried out through the SDK Toolkit.
- B. All requirements for the Integration Phase as defined in the E-Invoicing Regulation, as well as E-Invoice Specifications documents and Security Features must be fulfilled.
- C. Meeting the requirements under the verification process made through the SDK Toolkit does not imply that the E-invoices, credit or debit notes have been approved by ZATCA. And it does not exempt Persons subject to E-Invoicing Regulation and developers of technical solutions from the responsibility of ensuring that the E-Invoice meets the E-Invoicing requirements; or any penalties or fines arising from failure to comply with applicable laws.

I accept the above terms and conditions

[Download SDK](#)





External Document

This guide has been prepared for educational and awareness purposes only, its content may be modified at any time. It is not considered in any way binding to ZATCA and is not considered in any way a legal consultation. It cannot be relied upon as a legal reference in and of itself, It is always necessary to refer to the applicable regulations in this regard. Every person subject to zakat, tax and customs laws must check his duties and obligations, he is solely responsible for compliance with these regulations. ZATCA shall not be responsible in any way for any damage or loss The taxpayer is exposed to that results from non-compliance with the applicable regulations.