



**Ministry of Digital Economy
& Entrepreneurship**

QR Code Generator - Guideline Document

DOCUMENT CONTROL

| Version | Date | Author | Change Reference |
|---------|-----------------------------|---------------------|------------------|
| 1.0 | DEC 12 th , 2022 | Agiletz team member | Initial Document |
| | | | |
| | | | |

REVISION HISTORY

| Version | Date | Author | Comments |
|---------|------------|--------|----------|
| V1.0 | 20/12/2022 | | |
| | | | |
| | | | |

Definitions and Glossary

| Term | Definition |
|---------------------|---------------------------------------------------------------------|
| MoDEE | Ministry of Digital Economy and Entrepreneurship |
| Entity | Any government entity such as Ministry, Department ... etc. |
| Super Administrator | MoDEE administrator account that can manage all government entities |
| User | Government Entity account that can use and consume the QR API's. |

TABLE OF CONTENTS

| | |
|----------------------------------------------------------|----|
| INTRODUCTION | 4 |
| BUSINESS REQUIREMENT | 4 |
| PREREQUISITES TO USE QR FROM ANY GOVERNMENT ENTITY | 4 |
| PORTAL SPECIFICATION AND FEATURES..... | 4 |
| PORTAL SCREEN - USER FEATURES | 6 |
| QR CODE SPECIFICATION | 10 |
| RECOMMENDATION AND BEST PRACTICE: | 10 |
| QR CODE IMPLEMENTED SECURITY | 10 |
| THE STANDARD REQUIREMENT TO BE CONSIDERED:..... | 10 |
| HOW TO IMPLEMENT NEW QR | 11 |
| HOW TO VALIDATE QR USING ONLINE VALIDATOR:..... | 14 |

Introduction

This document will provide multiple views of QR system components and guidelines for the user; What and How to use the QR system, QR code definition, components, portal features, user roles, QR code solutions operation, recommendations, and best practices.

Business Requirement

Ministry of Digital Economy and Entrepreneurship (MoDEE) has many services and government entities that requested to use QR code generator, to secure government documents from forgery or any suspicious manipulation by using the QR-Code, therefore, we developed a flexible QR management solution to give MODEE the ability to manage each entity separately and to choose which information need to be encoded in a flexible way.

With a **dynamic QR code**, you can change the embedded information anytime without creating a new QR code, this saves you time and money.

The secure QR code solution can easily plug into existing document production systems for identification and verification purposes. It generates a highly secure personal identification format with embedded data that is placed on the document as a QR Code.

Prerequisites to use QR from any government entity

There is some configuration that is required to be set in both source and destination environments as per the below details:

- Make sure to add the new government domain entity to the MoDEE whitelist by adding the IP domain to the host file (local DNS file) in order to access it privately.
- Also, need to add the private IP address of MoDEE to the host file of the government entity as well.
- Network team must create a rule to allow communication between MoDEE and government entities.

Portal Specification and features

All users can be able to access the QR generator through the link <https://portal.qr.gov.jo/login>, based on the user role, users can be able to use and access the features.

User Authentication:

- 1- **Government Entity User**, as a user will be able to manage the QR services detail, APIs, and selected QR keys, below are the features available for this role which are:
 - a. Define and manage the QR document services for each entity and define what are the API's source to be populated (Source QR document).
 - b. Select the required Keys for each QR service document.
- 2- **QR code generator in runtime**: based on the predefined information (entity, services, and selected keys, the system will generate a QR code for a specific service.

Portal screen - User Features

The below features are accessible by government entities which are given access by the Admin portal.

- **Services Management:** User can be able to view the list of services that belongs to the entities, or he can be able to either **Search**, **Add New Services** (refer to next point), **Delete**, or **View** specific entities as the below screen shown in **figure 3.0**.

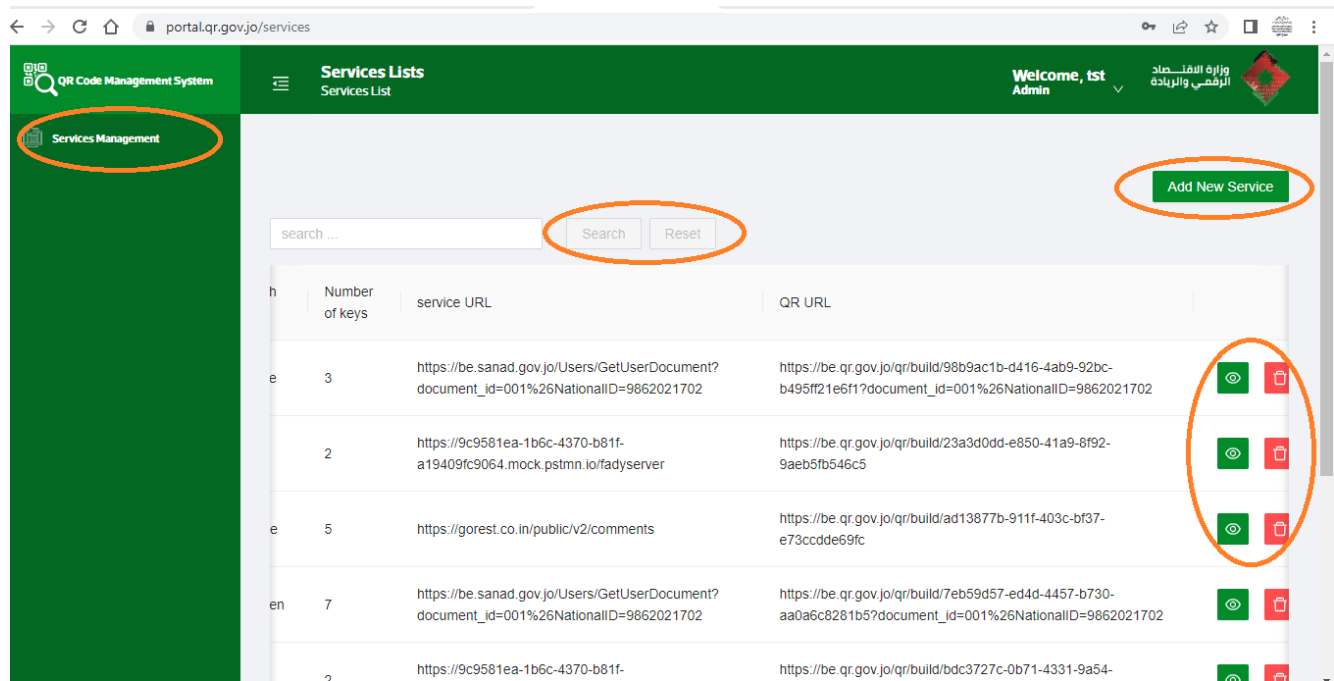


Figure 3.0

- **Add new Service:** User can be able to add a new government entity Service, however, the system can accept **4 types of APIs Authentication** Additionally, the system can accept two types of “**Request Type**”, which could be either: Get or Post, this combination gives enrichment to the system to accept different types of API, below screen shown from **figure 3.1** till **3.5**.

The four types of APIs Authentication are:

1. Token

The screenshot shows the 'Add Service' form in the QR Code Management System. The form is titled 'Add Service' and includes a breadcrumb 'Services List > Add New Service'. The user is logged in as 'tst Admin'. The form contains the following fields:

- Service English Name**: A text input field.
- Service Arabic Name**: A text input field.
- Authentication Type**: A dropdown menu with 'Token' selected.
- Request Type**: A dropdown menu with 'Please select authentication type' selected.
- Token**: A text input field.
- Service Url**: A text input field with 'URL' entered and a search icon.
- Select Key Elements**: A dropdown menu with 'Please select keys' selected.

At the bottom of the form are two buttons: 'Cancel' and 'Submit'.

Figure 3.1

2. Basic Authentication

The screenshot shows the 'Add Service' form in the QR Code Management System for Basic Authentication. The form is titled 'Add Service' and includes a breadcrumb 'Services List > Add New Service'. The user is logged in as 'tst Admin'. The form contains the following fields:

- Service English Name**: A text input field.
- Service Arabic Name**: A text input field.
- Authentication Type**: A dropdown menu with 'Basic Auth' selected.
- Request Type**: A dropdown menu with 'Please select authentication type' selected.
- Username**: A text input field.
- Password**: A text input field.
- Service Url**: A text input field with 'URL' entered and a search icon.
- Select Key Elements**: A dropdown menu with 'Please select keys' selected.

At the bottom of the form are two buttons: 'Cancel' and 'Submit'.

Figure 3.2

3. Open ID

The screenshot shows the 'Add Service' form in the QR Code Management System. The form is titled 'Add Service' and includes a breadcrumb 'Services List → Add New Service'. The user is logged in as 'Welcome, tst Admin'. The form contains the following fields:

- * Service English Name**: Text input field.
- * Service Arabic Name**: Text input field.
- * Authentication Type**: Dropdown menu with 'Open ID' selected.
- * Request Type**: Dropdown menu with 'Please select authentication type' selected.
- * Username**: Text input field.
- * Password**: Text input field.
- * Client ID**: Text input field.
- * Client Secret**: Text input field.
- * Service Uri**: Text input field with 'URL' entered and a search icon.
- * Select Key Elements**: Dropdown menu with 'Please select keys' selected.

At the bottom of the form are two buttons: 'Cancel' and 'Submit'.

Figure 3.3

4. No Authentication

The screenshot shows the 'Add Service' form in the QR Code Management System. The form is titled 'Add Service' and includes a breadcrumb 'Services List → Add New Service'. The user is logged in as 'Welcome, tst Admin'. The form contains the following fields:

- * Service English Name**: Text input field.
- * Service Arabic Name**: Text input field.
- * Authentication Type**: Dropdown menu with 'No Auth' selected.
- * Request Type**: Dropdown menu with 'Please select authentication type' selected.
- * Service Uri**: Text input field with 'URL' entered and a search icon.
- * Select Key Elements**: Dropdown menu with 'Please select keys' selected.

At the bottom of the form are two buttons: 'Cancel' and 'Submit'.

Figure 3.4

- Two types of “Request Type”, which could be either:
 - Post
 - Get

The screenshot shows the 'Add Service' form in the QR Code Management System. The form includes fields for 'Service English Name', 'Service Arabic Name', and 'Authentication Type'. The 'Request Type' dropdown menu is open, showing 'GET' and 'POST' options, which are circled in orange. The 'Select Key Elements' field is also visible, with a placeholder 'Please select keys'. The form has 'Cancel' and 'Submit' buttons at the bottom.

Figure 3.5

QR Code specification

- QR code is the base64 encoded TLV (Tag, Length, Value).
- The expected response structure always will be JSON Data format which is a data-interchange format, JSON is a text format that is completely language-independent.
- Each QR will have a unique code.
- QR can be Validated by other authenticated users.
- Requested body, can be Dynamic.
- Arabic characters will take double size rather than other languages.
- The key sequence order retrieval is dependent on the key selected order.

Recommendation and Best Practice:

- Each QR will have a max of 10 values including backend fields with a max of 2 to 3 values.
- Each Field data will have max 25 characters.
- Don't change the QR code once the document and QR are created.
- Don't use many keys with no value information.

QR Code Implemented Security

The below security parameters have been considered while developing the solution:

- Modee organization security: This includes security protocols and firewalls.
- QR Code digitally signed with the encryption key.
- Signature in the generated document.
- Prevent changing values in the QR code and/or change QR data.
- Protecting the application from unwanted vulnerabilities and threats.

The standard requirement to be considered:

- ClientID header and Client Secret header need to be put in the Client Id and Client Secret (Data entry) text boxes.
- ClientID Value and Client Secret Value needs to be put in the user name and password (Data entry) text boxes.
- SANAD Services need to use a Token Auth type instead of OpenID.
- SANAD Service needs to include the Token, generated by SANAD Auth URL in the header when you generate and validate the QR for any services.

How to Implement new QR

```
curl --location --request POST 'https://be.qr.gov.jo/qr/build/[SERVICE_ID]?[SERVICE_PARAM]' \
\
--header 'service_token: [DOCUMENT_SERVICE_TOKEN]' \
--header 'Authorization: Bearer [TOKEN_OF_USER_HO_CREATED_SERVICE]' \
--header 'Content-Type: application/json' \
--header 'qr_include_keys: [INCLUDE_KEYS_OPTION]' \
--data-raw '[POST_PODY]'
```

Examples

//GET SERVICE WITH TOKEN AUTH

```
curl --location --request POST 'https://be.qr.gov.jo/qr/build/f5f63b5d-ba19-4ecf-9244-9788ba306122?document_id=001&NationalID=9862021702' \
--header 'service_token: C1dluuuGYaSnqygYNfBD9xuiuJOIB72mcgr14FdA' \
--header 'Authorization: Bearer eyJhbGciOiJS...' \
--header 'Content-Type: application/json' \
--data-raw '{}'
```

```
curl --location --request POST 'https://be.qr.gov.jo/qr/build/f5f63b5d-ba19-4ecf-9244-9788ba306122' \
--header 'service_token: C1dluuuGYaSnqygYNfBD9xuiuJOIB72mcgr14FdA' \
--header 'Authorization: Bearer eyJhbGciOiJS...' \
--header 'Content-Type: application/json' \
--data-raw '{}'
```

//GET SERVICE with TOKEN AUTH and Include Keys in QR

```
curl --location --request POST 'https://be.qr.gov.jo/qr/build/f5f63b5d-ba19-4ecf-9244-9788ba306122?document_id=001&NationalID=9862021702' \
--header 'service_token: C1dluuuGYaSnqygYNfBD9xuiuJOIB72mcgr14FdA' \
--header 'qr_include_keys: true' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5c.. \
--header 'Content-Type: application/json' \
--data-raw '{}'
```

//POST SERVICE with NO AUTH and Include Keys in QR

```
curl --location --request POST 'https://be.qr.gov.jo/qr/build/f5f63b5d-ba19-4ecf-9244-9788ba306122?document_id=001&NationalID=9862021702' \
--header 'qr_include_keys: true' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5c.. \
--header 'Content-Type: application/json' \
--data-raw '{
    "RegNo": "",
    "NatId": "",

```

```

    "RegAddressName": "",
    "RegNatId": "100469840",
    "CommName": "",
    "RegStatusCode": "",
    "RegGovCode": "",
    "OwnerName": "",
    "PSize": 10,
    "PNumber": 1
  }

```

| API APRAM | Description | Mandatory /Optional | Example |
|-----------------------------------|-----------------------------------------------------------|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SERVICE_ID | Created service ID, that the QR needs to be built against | Mandatory | f5f63b5d-ba19-4ecf-9244-9788ba306122 |
| [SERVICE_PARAM] | mostly used in case of GET service | Optional | ?document_id=001&NationalID=9862021702 |
| DOCUMENT_SERVICE_TOKEN | | mandatory in case of TOKEN Auth | Header: 'service_token: C1d1uuuGYaSnqygYNfBD9xuiiuJOIB72mcgr14FdA' |
| TOKEN_OF_USER_WHO_CREATED_SERVICE | | login user must be token for the user who created the service to be able to generate | eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJSX2RqcW9CQ2dXZ1ppa0dLaWVfdklycVpLZUdHWmFBdWVnX21RRVB2bTRJIn0.eyJleHAiOjE2NzExNTY1MjAsImhhdCI6MTY3MTEyMDUyMCwianRpIjoimTdkNzFmZDQtYzk2My00YTZhLTlhYTQtYThlNGQ3ZmMwODFjIiwiaXNzIjoiaHR0cDovL3NhbmFkLWtleWNSb2FrLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwvcvVhbGlzL0R5bmFtaWNRckNrIiwiaXVkJjpbInJlYWxtLWlhbWFnZW11bnQiLCJhY2NvdW50I10sInN1YiI6Ijg3NTY3M2M4LTE0MjItNGIwNC1iNGYzLWVjNzIyNzdjMmJjMyIsInR5cCI6IkJlYXJlciIsImF6cCI6IkkxvZ2luIiwic2Vzc2lvdj19ZGF0ZSI6ImRjOTMyODRkLTVjMTAtNGExMC04MjZlLTNlZGM0ZWJhZjNhZiIsImFsbG93ZWQtb3JpZ2luc |

| | | | |
|---------------------|---------------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | | <p>yI6WyIqIl0sInJlYWxtX2FjY2VzcyI6ey Jyb2xlcyI6WwYkZmZhdWx0LXJvbGVzLWR 5bmFtaWNxcnNrIiwib2ZmbGluZV9hY2Nl c3MiLCJlbWFffYXV0aG9yaXphdGlubiIsI kFkbWluI1l9LClJyZXNvdXJjZV9hY2Nlc3 MiOnsicmVhbG0tbWFuYWdlbWVudCI6eyJ yb2xlcyI6WwYjYtYW5hZ2UtcmlvbmG0iLCJt YW5hZ2UtdXNlcnMiLCJtYW5hZ2UtY2xpZ W50cyJdfSwiYWNjb3VudCI6eyJyb2xlcy I6WwYjYtYW5hZ2UtYWNjb3VudCIsImhbmF nZS1hY2NvdW50LWxpbmtzIiwidmlldylw cm9maWxlI1l9fSwic2NvcGUioiJlbWFPb CBwcm9maWxlIiwic2lkIjoizGM5MzI4NG QtNWMxMC00YTEwLTgyNmUtM2VkYzRlYmF mM2FmIiwizZWlhaWxfdmVyaWZpZWQiOnRy dWUsIm5hbWUiOiJmYWR5IGZhZHkiLCJwc mVmZXJyZWRFdXNlcm5hbWUiOiJmYWR5QH Rlc3QuY29tIiwizZW5fbmFtZSI6ImZ hZHkiLCJmYWlpbHlfbmFtZSI6ImZhZHki LCJlbWFPbCI6ImZhZHlAdGVzdC5jb20if Q.SQ2Mt7yl8XdPmkfk_aSyqcJbc1wjFjo BkEDHimO5_5sRWxYGrq9z8IauX0QQq3oJ jGZF9GtHMVYH74A2GlgxttDpsaHz26Zc7 eg7- 4z4qZ4VV3YLamfO3JiIEPRADOUly4l0ka kTkpr3wt3FOLuL7D2IfOi3iBS-9oYE1B- BBUXSltcWYMgB3a0u4wSAg5poG4- StpyARF4WCSKPghBNVmcNVY7wVFfa_jtx4 t96Qx8EzagjFlpadnyUTfluwJqde3HIkL pJtHObkTXNpn7BpmCaQe6ekKABJGtg13R HRjAgqtd6gTLLdEH9BJR0a6cCYIIotZsw iLgtLvFpalvzdwQ</p> |
| INCLUDE_KEYS_OPTION | Decide while generation, if keys are to be included in QR TLVs or not | OPTIONAL default=false | <p>True: generated QR will include keys in TLV</p> <p>False: generated QR will include only values in TLV</p> |
| POST_BODY | { } in case of get and service post body in case document service is POST | Mandatory | <pre>{ "RegNo": "", "NatId": "", "RegAddressName": "", "RegNatId": "100469840",</pre> |

| | | | |
|--|--|--|------------------------------------------------------------------------------------------------------------------|
| | | | <pre> "CommName": "", "RegStatusCode": "", "RegGovCode": "", "OwnerName": "", "PSize": 10, "PNumber": 1 } </pre> |
|--|--|--|------------------------------------------------------------------------------------------------------------------|

How to Validate QR using Online Validator:

```

curl --location --request POST 'https://be.qr.gov.jo/qr/validate/' \
--header 'service_token: [DOCUMENT_SERVICE_TOKEN]' \
--header 'Authorization: Bearer [QR_SYSTEM_TOKEN]' \
--header 'Content-Type: application/json' \
--data-raw '{
    "qrCode": "[QR_TO_BE_VALIDATED]"
}'

```

Example:

```

curl --location --request POST 'https://be.qr.gov.jo/qr/validate/' \
--header 'service_token: R5s2R7JjokrwJuzl9ng026bMn3LlRMuIexIJhQM2' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1... ' \
--header 'Content-Type: application/json' \
--data-raw '{
    "qrCode":
"AUo5MmIyZTE2Ni1mMzE2LTRmZDctOTQ3MC0wOWNhZjU0NzE5YWI/ZG9jdWllbnRfaWQ9MDAxJk5hdGlvbmFsSUQ9OT
c3MTAzODUzNQICE30CC2dlbmRlcj1NYWxlAxBVYXRpb25hbElEPTk3NzEwMzg1MzUEJmVuZ2xpc2h0YW1lPU1BSE1PV
UQgWU9VU0VGIE1PVVNBIEIERU1TBTJhcmFiaWNOYW1lPdmF2K3Zhdml2K8g2YrZiNiz2Yeg2YXZiNiz2Ykg2KfYr9i5
2YrYswZgTUVZQ0lRRGkzOGtWcG4xU0JxRW1lU1NiN01Dbm9kWHZBRUYrNnVtVW5pZ2xheVhIVXdJaEFNOW5XVmEyeFB
SRTlCdVR4TzVnWmUrZjQxVlh2ZW9aa0U0YmRCM29WYkVs"
}'

```

| API APRAM | Description | Mandatory /Optional | Example |
|--------------|-------------|------------------------|---------|
|--------------|-------------|------------------------|---------|

[illegible]

| | | | |
|--------------------|--------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | | AkTkpr3wt3FOLuL7D2IfOi3iBS-9oYEIB-BBUXSltcWYMgB3a0u4wSAg5poG4-StpyARF4WCSKPghBNVmcNvY7wVFfa_jtx4t96Qx8EzagjF1padnyUTfluwJqde3HlkLpJtHObkTXNpn7BpmCaQe6ekKABJGtg13RHRjAgqtd6gTLLdEH9BJR0a6cCYllotZswiLgtLvFpa1vzdwQ |
| QR_TO_BE_VALIDATED | QR to be validated | Mandatory | <p>True: generated QR will include keys in TLV</p> <p>False: generated QR will include only values in TLV</p> |