وزارة الداخلية
Ministry of Interior
State of Qatar ◆ دولة قطر

Information Systems Department
Systems Development and Maintenance Section

Qatar Digital Identity for External Entities (QDEX)

Integration Document

Status : Draft
Version : 0.01
Prepared by : MUHAMMED SAJEER KOROTH
Modified by :

Approvals:

______________________________ _______________
Iyad Ahmad Date

______________________________ _______________
Application Integration Committee Date

Circulation:
MOI- IS Department

Confidential

Document Change History

Doc Version	**Service Version**	**Date**	**Owner**	**Description**
0.01	1.0.0	16/02/2023	MUHAMMED SAJEER KOROTH	First draft

Publishers
Owner: Iyad Ahmed, Project Manager, ISD, MOI
Contributors:

Contents

1. General description

This document outlines the technical specifications that external service providers must adhere to for seamless integration with the QDI App (Qatar Digital Identity). The QDI App is a core component of the Qatar Public Key Infrastructure (QPKI) platform, which is managed and operated by the Ministry of Interior (MOI).

QDI services, provided by the MOI, enable integration of the national login system via NAS (National Authentication System) for the Ministry of Communications and Information Technology (MCIT) and other external entities. To access these services, end users are required to install the QDI mobile application. This app provides essential functionalities such as authorized document signing and secure login to e-services.

2. QDI and QPKI Overview

The Qatar Digital Identity (QDI) App, a core element of the Qatar Public Key Infrastructure (QPKI) initiative, represents a government-led effort to establish a secure and trusted framework for digital communications and transactions. Managed by the Ministry of Interior (MOI), the project focuses on deploying a robust infrastructure to enable the issuance, distribution, and management of digital certificates and keys. These capabilities ensure secure digital transactions and foster trust in electronic services.

Key Objectives and Features:

1. **Enhanced Security and Trust:**
   The QDI App underpins secure communication and data exchange between government agencies, private organizations, and citizens, reducing fraud and bolstering trust in digital services.
2. **E-Government Enablement:**
   The project supports e-government services by enabling seamless login to the National Authentication System (NAS) portal. This provides Single Sign-On (SSO) access to all government agencies and ensures a secure and reliable channel for individual document signing.
3. **Efficiency and Scalability:**
   By leveraging PKI-based security, the project enhances efficiency in digital transactions, creating a scalable framework for future digital service expansions.
4. **Comprehensive Stakeholder Engagement:**
   The initiative involves various stakeholders, including government entities, private organizations, and citizens. It prioritizes stakeholder education to maximize understanding and adoption of the PKI system's benefits.

Core Benefits:

- Strengthened security for digital communications.

- Streamlined access to services via SSO through NAS.
- Increased efficiency and reduced fraud in digital transactions.
- Improved trust and usability of government and private digital services.

In essence, the QDI App and the broader QPKI initiative aim to create a secure, efficient, and trusted digital ecosystem, aligning with Qatar's vision of fostering advanced and reliable e-governance systems.

3. Purpose

This document outlines the process for integrating external entities with the MOI's QDI and PKI infrastructure. The QDE is a SOAP-based web service that utilizes the Simple Object Access Protocol (SOAP) for seamless communication with external applications.

4. WebService Security

The service uses UserName token profile as security standards.
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf

5. QDI Web Service Endpoint URL

Test Environment

https://testqdign.moi.gov.qa/QDE_EJB_HTTPRouter/QDEManagerWSService/QDEManagerWSService.wsdl

Production

https://qdign.moi.gov.qa/QDE_EJB_HTTPRouter/QDEManagerWSService/QDEManagerWSService.wsdl

6. WebService Methods:

4.1 getVersion:

4.1.1 Security / Availability

Security Role (RACF): QDE00001
Available Since Service Version: 1.0.0

4.1.2 **Function description**

The service method is used to get the current version of QDE Service.

4.1.3 Input / output parameters

Function Name	Input	Output	Exceptions
getVersion		String	WsException

4.1.4 Input parameters

No Input parameter required.

4.1.5 Business

1. This method will return the current version of the QDE web service.
2. "QDE600" will be thrown in case if version information does not exists.

4.1.6 Output parameters

String: current version of the service.

4.1.7 Error Handling

One type of Exception will be handled in this function:

- A general exception (**WsException**)

WsException:

Method Name	Type	Description
getErrorCode()	String	Error Code representing the error
getErrorMessageARB()	String	Arabic Error Message in case error happened.
getErrorMessageENG()	String	English Error Message in case error happened.

4.2 validateQid:

4.2.1 Security / Availability

Security Role (RACF): QDE00001
Available Since Service Version: 1.0.0

4.2.2 **Function description**

This service is used to validate a Qid to do any digital transaction related to QPKI. It includes all the validation based on the Qid which also includes the given Qid holder has any restriction or blacklist references.

4.2.3 Input / output parameters

Function Name	Input	Output Dto	Exceptions
validateQid	String qid	QDEUserDto	WsException

4.2.4 Input parameters

Field Name	Type	Mandatory	Description
qid	String[11]	True	Qid to validate

4.2.5 Business

The service will return *responseCode*=1 if there is no Blacklist record or Restriction record for the given Qid after validating the Qid validity information. It also checks whether the Qid holder has an active digital certificate in the format *QID_VSC_QDI* .Otherwise, it will return *responseCode*=0 with proper error messages. If the responseCode=0, it says that there is an issue to move further with this Qid and the reason will be provided in the properties which indicate error messages.

4.2.6 Output parameters

QDEUserDto Object details

Field Name	Type	Description
responseCode	String	1 – (SUCCESS. This flag indicates it is fine with proceeding the request for this particular Qid.) 0 – (ERROR. This flag indicates there is an issue with this Qid holder to proceed with the request.)
successMessageEN	String	Success message in English
errorMessageEN	String	Error message in English
erroMessageAR	String	Error message in Arabic
successMessageAR	String	Success message in Arabic
certficateAlias	String	Specifies the certificate alias for this Qid holder. It will be in the format *Qid_VSC_QDI* (Eg:-27163400125_VSC_QDI)

certificate	String	Public Certificate created for this user.

4.2.7 Error Handling

One type of Exception will be handled in this function:

- A general exception (**WsException**)

WsException:

Method Name	Type	Description
getErrorCode()	String	Error Code representing the error
getErrorMessageARB()	String	Arabic Error Message in case error happened.
getErrorMessageENG()	String	English Error Message in case error happened.

4.3validateAndAuthorizeQid:

4.3.1 Security / Availability

Security Role (RACF): QDE00001
Available Since Service Version: 1.0.0

4.3.2 **Function description**

This service enables the implementation of remote authorization login via the QDI Mobile App. External entities, including NAS, can create an authorization request in the QDI Mobile App using this service, which will then be approved or rejected by the QID holder. Based on the response of this call, the initiator of the service call can decide whether to allow the user to log in to the system or not.

4.3.3 Input / output parameters

Function Name	Input	Output Dto	Exceptions
validateAndAuthorizeQid	QDEInputDto	ResponseDto	WsException

4.3.4 Input parameters QDEInputDto

Field Name	Type	Mandatory	Description

qid	String[11]	True	Qid holder who tries to login the system
systemName	String	True	System Name defined for the external entity. Eg:- NAS
sessionId	String	True	Web session Id to be captured from the browser.
browserType	String	True	The browser used by the user. IE, Mozilla/5,chrome etc.
sourceIP	String	True	Source IP Address .
data2Display	String	No	This field is used to hold the display message to be shown in the QDI Mobile APP while Authorizing the Approval Request.

4.3.5 Business

When the qid holder's digital certificate is successfully verified, the service will respond with *responseCode=1*, allowing the external entity to redirect to their home page. However, if there are any issues with the validation of the digital certificate, the service will respond with *responseCode=0* along with an appropriate error message indicating the reason. In the case of *responseCode=0*, it means that there is an issue and the external entity cannot proceed with this qid. The error messages will be provided in the properties of the ResponseDto.

4.3.6 Output parameters

ResponseDto Object details

Field Name	Type	Description
responseCode	String	1 – (SUCCESS. This flag indicates it is fine with proceeding the request for this particular Qid.) 0 – (ERROR. This flag indicates there is an issue with this Qid holder to proceed with the request.)
successMessageEN	String	Success message in English
errorMessageEN	String	Error message in English
erroMessageAR	String	Error message in Arabic
successMessageAR	String	Success message in Arabic

4.3.7 Error Handling

One type of Exception will be handled in this function:

- A general exception (**WsException**)

WsException:

Method Name	Type	Description
getErrorCode()	String	Error Code representing the error
getErrorMessageARB()	String	Arabic Error Message in case error happened.
getErrorMessageENG()	String	English Error Message in case error happened.

4.4 retrieveSignedDocument:

4.4.1 Security / Availability

Security Role (RACF): QDE00001
Available Since Service Version: 1.0.0

4.4.2 **Function description**

This service is used to retrieve a document that contains a digital signature by reading the document from the FileNet. This service will fetch the document from the FileNet by using documentReferenceNumber.The service verifies the digital signature and its validity and it will return the document if the document is valid.

Input / output parameters

Function Name	Input	Output Dto	Exceptions
retrieveSignedDocument	String documentReferanceNumber	SignedDocumentDto	WsException

4.4.3 Input parameters

Field Name	Type	Mandatory	Description
documentReferenceNumber	String	True	Document Reference Number to get the document from the file net.
systemName	String	True	System Name defined for the external entity.

ipaddress	String	True	IP Address.

4.4.4 Business

The service will retrieve the document from the fileNet based on the documentReferanceNumber. This digital signatures embedded in the document will be verified and the service will return the verification output and the document if the digital signature is valid.

4.4.5 Output parameters

SignedDocumentDTo Object details

Field Name	Type	Description
responseCode	String	1 – (SUCCESS. This flag indicates it is fine with proceeding the request for this particular documentReferanceNumber.) 0 – (ERROR. This flag indicates there is an issue with this documentReferance holder to proceed with the request.)
successMessageEN	String	Success message in English
errorMessageEN	String	Error message in English
erroMessageAR	String	Error message in Arabic
successMessageAR	String	Success message in Arabic
documentContent	Byte[]	The content of the document in byte array format.
issueDate	String	Document issue date.
expiryDate	String	Document expiry date.
signatureStatus	String	The status indicates whether the given signature is valid or not. (Valid/Invalid)

DocumentStatusDto Object details

Field Name	Type	Description
documentStatus	String	The status indicates (Active/Inactive)
descriptionEN	String	Description in English

descriptionAR	String	Description in Arabic

4.4.6 Error Handling

One type of Exception will be handled in this function:

- A general exception (**WsException**)

WsException:

Method Name	Type	Description
getErrorCode()	String	Error Code representing the error
getErrorMessageARB()	String	Arabic Error Message in case error happened.
getErrorMessageENG()	String	English Error Message in case error happened.

5. Web Service Error Codes:

Error Code	English Description	Arabic Description
QDE999	Internal Server Error	
QDE888	External Service Connection Timeout Settings Error	
QDE777	Not able to access external service	
QDE666	Error while calling external service	
QDE600	NO VERSION INFORMATION FOUND	
QDE500	Database Error	
QDE501	Unable to locate data source	
QDE502	Unable to get Database Connection	
QDI117	Unable to proceed, Refer to the Qatari passport department	
QDI118	Unable to proceed, Refer to the immigration department	
QDI119	Qid does not exists or its status is invalid	
QDI120	Qid has been expired, Please renew it	
QDE121	Please provide a valid 11 digit QID	
QDI126	Error while calling BLS(Black List) Service	
QDI127	Qid Status is invalid	
QDI128	Personal information not found for this QID	
QDI129	RP Expiry date not found	
QDI135	Please provide a valid video input	
QDI190	User status is suspended in QDI mobile Application	
QDI191	No valid certificate found	

QDI192	QDI user status is invalid	
QDI231	User is not registered/Active for digital signing	
QDI235	Age should be at least 18 Years to generate digital signature	
QDI236	User did not respond to the pending request using Mobile App within 30 seconds	
QDI243	The document has been expired.	
QDI256	You are not authorized to view this certification.	
QDI260	User did not respond the mobile authorisation using QDI Mobile App within the stipulated time period.	
QDI264	The signature is not trusted	
QDI269	External service token is expired or invalid	