QRNG API

How to use Quantum Blockchains QRNG API?

Introduction

Quantum Blockchains Inc. offers its QRNG based EaaS (Entropy as a Service) to its clients through a well defined REST based Web API. As for June 2022, the service is offered free of charge. It may become commercial later this year. Interested users shall contact our company for details.

Our service is based on ID Quantique Quantis QRNG installed at our lab. The software layers built on top of the Quantis hardware and on the basic library supplied by its manufacturer have been created by Quantum Blockchains Inc.

In addition to the QRNG installed in our lab, the software layers we have built allow for the use of other sources of random numbers. We have tested the delivery of random numbers from two other providers. We are now building the respective software layers to include their bitstreams into our service. Interested users shall contact us for the details.

Subscribing to the QRNG EaaS services

To subscribe to our services, navigate to https://www.quantumblockchains.io/current-services/api/, fill and submit the form by providing your name, affiliation and e-mail address.

You will receive the mail from qrng@quantumblockchains.io with the links to documentation, Postman collection and the activation link. The zip file that contains the postman collection also includes the SSL certificate required to use HTTPS API calls – **chain.pem**. Depending on the application you will use to call the API you will have to indicate to it the location of that important file.

To start using the service, please follow the activation link. The API key will be displayed:

Your new key:43e339c2-dc62-4a98-ade6-64c060f0d783

A note: If you do not receive the mail after some time since the subscribing, please check the spam folder. Some more stringent spam filters may block the activation mails.

Using the QRNG EaaS services

The QRNG EaaS interface defines three Web API methods:

The API assumes use of HTTP GET call pattern with the following method:

numeric_range - returns various numerical types of random numbers in the desired range (min, max). The types the client can request are: int (or short), long – for integer types and float and double for floating-point types.

byte_types - returns random bytes in various representations: bigint, hex, bin, base64, base56.

bin_file - generates a file with random binary content and allows to download the file.

The API methods and their parameters

Method: numeric_range:

`Type: HTTPS GET`

Method call:

```
https://qrng.qbck.io/<APIKey>/<provider>/<target>/<type>?size=<size>
&min=<min>&max=<max>&file=<file_name>
```

Method parameters:

`APIKey:`
The client API key obtained during the registration process

`provider:`
Random bitstream provider code. Use 'qbck' for the current QRNG service provided by Quantum Blockchains. Codes for other providers will be communicated separately when they are offered.

`target:`
Requests the delivery form:
`block` – deliver data in the method response block (in JSON format)
`download` – deliver data as a download file (which name is specified by 'file' parameter)

`type:`
declares the numeric format of random numbers:

`int`	- 32-bit integer
`short`	- 16-bit integer
`long`	- 64-bit integer
`float`	- 32-bit floating point number
`double`	- 64-bit floating point number

`size:`
declares the size of the sample - the number of random numbers generated

`min:`
defines the lower bound of the range of the generated numbers

`max:`
defines the upper bound of the range of the generated numbers

`file:`

declares the name of the file when the results are to be downloaded. If empty or not specified, the result is returned in the call response in the JSON format.

Default values for parameters

The call parameters: `size, min, max, file` are optional. If they are not present the following default values are assumed:

```
size:
min:
max:
file:
```

Examples:

A) Using curl

Assuming that your key is: 43e374c2-dc62-4a98-ade6-64c060f0d683, and that you have the chain.pem certificate file in the current folder, you can call the **`numeric_range`** method using curl:

```
curl --cacert chain.pem https://qrng.qbck.io/43e374c2-dc62-4a98-
ade6-64c060f0d683/qbck/block/short?size=20&min=0&max=100
```

which returns:

```
{
    "data": {
        "result": [
            49,
            61,
            98,
            43,
            91,
            70,
            17,
            76,
            28,
            38,
            63,
            9,
            87,
            70,
            69,
            29,
            43,
            33,
            55,
            35
        ],
        "QuantisRead": 20,
        "ExecuteTime": "89ms"
    },
    "error": "OK",
    "message": "Random list.",
    "timestamp": "2022-05-22T22:05:15.715+02:00",
    "status": 200
}
```

B) Using Postman

Postman is the primary tool for API testing. We have provided the “collection” definition file: **qrng.postman_collection.json** to help in creation of calls to our QRNG API. After unpacking the ZIP file linked to the introduction message, you must import the collection definition file into the set of Postman collections. Go to the collection tab and press “Import” and then “Upload Files”. Point to the location of qrng.postman_collection.json and read it. Postman will create “qrng” collection with the three calls of our API. Then go to “Settings”. Open “Certificates” tab and register the CA Cerificate from the provided chain.pem file.

You are now ready to use a specific call of our API. First, you need to supply the authorization key you obtained after successful registration at https://www.quantumblockchains.io/current-services/api/ to the **APIKey** variable.
If, for example, you want to get double precision floating numbers from 0 to 1000, you need to go to “Pre-request Script” tab and modify the line with **type** variable:

```
var APIKey   = "43e334c2-dc62-4a98-ade6-64c060f0d683";// ENTER THE AUTHORIZATION KEY HERE
var provider = "qbck";  // Enter QRNG provider code from documentation
var type     = "double";  // select type :"int" or "short" or "long" or "float" or
"double"
var size     = 10;       // the amount of random numbers generated
var min      = 0;        // min value
var max      = 1000;     // max value
var file     = ""        // if empty, the result is json else result to file
//================================================================================
```

After these steps you can hit the “Send” button. If all is OK, the response shall follow immediately:

```
{
    "data": {
        "result": [
            200.46308421945375,
            89.74242337819285,
            419.6639897083956,
            526.4455943470589,
            436.1208538388392,
            738.2790320417547,
            5.677141376047059,
            232.6086208742085,
            491.830285152453,
            571.1162394191692
        ],
        "QuantisRead": 10,
        "ExecuteTime": "87ms"
    },
    "error": "OK",
    "message": "Random list.",
    "timestamp": "2022-06-05T22:31:11.862+02:00",
    "status": 200
}
```

Metod: byte_types:

Type: HTTPS GET

Method call:

```
https://qrng.qbck.io/<APIKey>/<provider>/
<target>/<type>?size=<size>&length=<length>&file=<file_name>
```

Method parameters:

`APIKey:`
The client API key obtained during the registration process

`provider:`
Random bitstream provider code. Use 'qbck' for the current QRNG service provided by Quantum Blockchains. Codes for other providers will be communicated separately when they are offered.

`target:`
Requests the delivery form:
`block` – deliver data in the method response block (in JSON format)
`download` – deliver data as a download file (which name is specified by 'file' parameter)

`type:`
declares the numeric format of random numbers:

`bigint`	- decimal biginteger number
`hex`	- hexadecimal number
`bin`	- binary number
`base64`	- base64 formatted numbers
`base56`	- base56 formatted numbers

`size:`
declares the size of the sample - the number of random numbers generated

`length:`
declares the length of the random number in bytes (for bigint, hex and bin formats) or in characters (for base64 and base56 formats)

`file:`
declares the name of the file when the results are to be downloaded. If empty or not specified, the result is returned in the method response in the JSON format.

Default values for parameters

The call parameters: `size, length, file` are optional. If they are not present the following default values are assumed:

```
size:
length:
file:
```

Examples:

A) Using curl

Assuming that your key is: 75e8607c-ceae-421b-a5d3-1b9a077999a7, and that you have the chain.pem certificate file in the current folder, you can call the byte_types method using curl to obtain 30 big integer numbers each of 128 bytes length:

```
curl --cacert chain.pem https://qrng.qbck.io/75e8607c-ceae-421b-
a5d3-1b9a077999a7/qbck/block/bigint?size=30&length=128
```

which returns:

```
{
    "data": {
        "result": [
            "2230435890722731830491992517811989552 10",
            "32499606697776687111041110944214160386 3",
            "12087591391893032286609603272240896895 7",
            "30987987432331721628174038204927686565 0",
            "5105447783720745011465708818319150289 0",
            "18385469573628303090018740394293602565 3",
            "30925279358371474597004939757616522723 9",
            "10207163588920664854916979458122669860 7",
            "22855797854386656654291236923844157026 0",
            "20888008594861976942250487759850465838 8",
            "29638354077869472339715538579282055629 9",
            "18045524674077239144469623732959812016 6",
            "13699488542304318476715789634975902648 8",
            "33528537271191988329944185157610580921 4",
            "26774713966468443290316471476989554645 1",
            "20453758150211091806683034667264038167 8",
            "31117927488109880058904397336774651742 0",
            "13531189169605237811414210564142908523 0",
            "23273125710673613046414036495973849161 1",
            "2692062958736551241892380586800197186 7",
            "23928120688043207919422481219347770889 4",
            "7890019337730282008812817004004809087 0",
            "16464638116989658134341805241473388426 3",
            "15100325416781410252520097290505074695 8",
            "25078585069116319432986691852867007492 6",
            "1107587373125378717609612976396563298 1",
            "2019653640470450119728804320804050109 0",
            "16156737271688276805961175408744945577 4",
            "18283069261374966246763646352079677158 8",
            "33958135678097307612650934906424557320 6"
        ],
        "QuantisRead": 30,
        "ExecuteTime": "87ms"
    },
    "error": "OK",
    "message": "Random list.",
    "timestamp": "2022-06-19T13:13:50.280+02:00",
    "status": 200
}
```

B) Using Postman

Follow the initial steps we have described in the “using postman” example of the “numeric_range” method. Your Postman settings are now ready to use “byte_types” call.
Remember to go to the “Pre-request Script” section and to supply the authorization key you obtained after successful registration at https://www.quantumblockchains.io/current-services/api/ to the **APIKey** variable.
If, for example, you want to get 10 pure binary 64 bit strings (0s and 1s), in the “Pre-request Script” tab set the **type** variable to “bin”:

```
var APIKey  = "75e8607c-ceae-421b-a5d3-1b9a077999a7/";       // ENTER THE AUTHORI-
ZATION KEY HERE
var provider = "qbck";  // Enter QRNG provider code from documentation
var type = "bin";       // select type: "bigint" or "hex" or "bin" or "base64" or
"base56"
var size = 10;          // the amount of random numbers generated
var length = 8;        // the size of the random number in bytes (for bigint, hex
and bin) or characters (for base64 and base56)
var file = ""           // if empty, the result is json else result to file
//=================================================================================
```

Method: bin_file

Type: HTTPS GET

Method call:

```
https://qrng.qbck.io/<APIKey>/<provider>/download/bytes?length=<length>&file=<file>
```

Method parameters:

`APIKey:`
The client API key obtained during the registration process

`provider:`
Random bitstream provider code. Use ‘qbck’ for the current QRNG service provided by Quantum Blockchains. Codes for other providers will be communicated separately when they are offered.

`length:`
declares the length of the binary random numbers in bytes

`file:`
declares the name of the file when the results are to be downloaded. If empty or not specified, the result is returned in the random.bin file.

Default values for parameters

The call parameters: `length, file` are optional. If they are not present the following default values are assumed:

```
        length:
        file:
```

Examples:

A note: CURL does not properly handle downloadable content, so we present here examples with Postman, Web Browser and wget command.

A) Using Postman

Follow the initial steps we have described in the “using postman” example of the “numeric_range” method. Your Postman settings are now ready to use “bin_file” call.

Remember to go to the “Pre-request Script” section and to supply the authorization key you obtained after successful registration at https://www.quantumblockchains.io/current-services/api/ to the APIKey variable.

If, for example, you want to get 32 bytes pure binary bytes, in the “Pre-request Script” tab set the type variable to “bin”:

```
var APIKey  = "75e8607c-ceae-421b-a5d3-1b9a077999a7"; // ENTER THE AUTHORIZA-
TION KEY HERE
var provider = "qbck";  // Enter QRNG provider code from documentation

var length = 32;         // the size file in bytes
var file = "random32.bin"
//=================================================================================
```

Postman will return binary result, but it is rendered as 8-bit characters. To save the file in pure binary form use Postman’s “Save Response” action.

B) Using any web browser

You can simple use an URL of the following shape:

> https://qrng.qbck.io/db15d911-03a9-4f91-84fe-288582d52365/qbck/download/bytes?length=1000000&file=test.bin

to download 1M byte long stream of random bits.

Using wget

You can get the file with 100K byte long stream of random bits using the following wget command:

You can simple use an URL of the following shape:

```
wget -O random.bin --ca-certificate=chain.pem
"https://qrng.qbck.io/db15d911-03a9-4f91-84fe-
288582d52367/qbck/download/bytes?length=100000"
```