API Reference

S2S RNG API

API Version: 0.9.1

Szrek2Solutions Random Number Generator API

CONTACT

EMAIL: info@szrek.com
URL: http://www.szrek.com

INDEX

Security and Authentication

SECURITY SCHEMES

KEY	TYPE	DESCRIPTION

API

1. RNG

1.1 POST /rng/quick-picks

Quick Pick

Conduct RNG for quick pick products. Returns reduced-size results for performance.

REQUEST

REQUEST BODY - application/json

```
{
  clientId                    string      Name or identifier of the client system.
  clientTime                  string      Time on the client system.
  clientBusinessDay           integer     Client's current business day. Format is YYMMDD.
  clientTransactionId         integer     Client's unique identifier for this transaction.
  rngRequests* [{
  Array of object:
    clientRecordId    string    Client's unique identifier for this RNG request.
    productId*        integer   Identifier of the product to be generated. The product defines all RNG parameters,
                                and MAY allow them to be overridden in the RNG matrix structure.
    rngMatrix {
    The complete RNG parameter matrix. The product definition may or may not allow overriding of these values
      numberOfSets*         integer       >=1
                                          How many sets of numbers to generate.
      setSize               integer       >=1
                                          How many numbers in each set.
      minValue              integer       >=0
                                          Lowest number that can be generated.
      maxValue              integer       >=0
                                          Highest number that can be generated.
      withReplacement       boolean       True if generated numbers can be repeated within a set.
      sortOrder             enum          ALLOWED:UNSORTED, ASCENDING
                                          Sort order for the generated numbers within a set.
      exclusions            [integer]
      distribution {
      Initial distribution values and depletion type. Used for TABLE products only.
        values          [integer]
        depletionType   enum        ALLOWED:NO_DEPLETION, DECREASE_ENTRY,
                                    REMOVE_ENTRY
                                    Specifies method of depletion for the distribution values.
      }
    }
    clientData          [string]
  }]
}
```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  serverId*             string    Transaction server identifier.
  serverVersion*        string    Transaction server version number.
```

transactionStatus* {
Status code and description

code* integer Numeric error or status code.
description string Short description of the error or status code.

}
clientTransactionId integer Client's unique identifier for this transaction.
timestamp string Date and time the request was processed.
businessDay integer Business day under which the transaction was recorded. Format is YYMMDD.
results [{
Array of object:

rngStatus* {
Status code and description

code* integer Numeric error or status code.
description string Short description of the error or status code.

}
clientRecordId string Client's unique identifier for this RNG request.
recordId string Unique record identifier.
recordHandle integer Handle that can optionally be used for faster verification processing.
productId integer Identifier of the product definition used for the random generation.
productChecksum number CRC32 checksum of the product definition used for the random generation.
sets [{
Array of object:

}]
}]
}

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

{
Describes the error and any specific problems with fields in the request.

code* integer Numeric error or status code.
description string Short description of the error or status code.
fieldErrors [{
Array of object: Details about a specific field error.

code string Error code name.
message string Additional detail about the error code.
fieldName string Name of the field that caused the error, if applicable.
path string Path to the field that caused the error, if applicable.

}]
}

1.2 POST /rng/file-draws

File Draw

Conduct draws for file-based draw products.

REQUEST

REQUEST BODY - application/json

{
clientId string Name or identifier of the client system.
clientTime string Time on the client system.
clientBusinessDay integer Client's current business day. Format is YYMMDD.
clientTransactionId integer Client's unique identifier for this transaction.
testMode boolean Marks the draw as "test mode" and not valid for use.
drawUsers [{

Array of object: Draw participant's name and login time, to be included in the draw reports.

- `name` string — Full name of the user as it should appear in the draw reports.
- `loginTime` string — If included, the login time for the user will appear in the draw reports.

`}]`

`returnRngMatrix` boolean — Include RNG parameter matrix in RNG results.

`rngRequests* [{`

Array of object:

- `clientRecordId` string — Client's unique identifier for this RNG request.
- `productId*` integer — Identifier of the product to be generated. The product defines all RNG parameters, and MAY allow them to be overridden in the RNG matrix structure.

`rngMatrix {`

The complete RNG parameter matrix. The product definition may or may not allow overriding of these values

- `numberOfSets*` integer >=1 — How many sets of numbers to generate.
- `setSize` integer >=1 — How many numbers in each set.
- `minValue` integer >=0 — Lowest number that can be generated.
- `maxValue` integer >=0 — Highest number that can be generated.
- `withReplacement` boolean — True if generated numbers can be repeated within a set.
- `sortOrder` enum — ALLOWED:UNSORTED, ASCENDING — Sort order for the generated numbers within a set.
- `exclusions` [integer]
- `distribution {`

Initial distribution values and depletion type. Used for TABLE products only.

- `values` [integer]
- `depletionType` enum — ALLOWED:NO_DEPLETION, DECREASE_ENTRY, REMOVE_ENTRY — Specifies method of depletion for the distribution values.

`}`

`}`

- `clientData` [string]
- `fileName` string — Name of the file used for the draw.
- `fileHash` string — SHA-1 hash of the file used for the draw.

`}]`

`securityRequests [{`

Array of object: Optional requests for security blocks to be included in the response.

- `blockType*` string — Security block type identifier.
- `version` string — Format version of the security block.

`}]`

`}`

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

`{`

- `serverId*` string — Transaction server identifier.
- `serverVersion*` string — Transaction server version number.
- `transactionStatus* {`

Status code and description

- `code*` integer — Numeric error or status code.
- `description` string — Short description of the error or status code.

`}`

- `clientTransactionId` integer — Client's unique identifier for this transaction.
- `timestamp` string — Date and time the request was processed.
- `businessDay` integer — Business day under which the transaction was recorded. Format is

YYMMDD.

testMode boolean Marks the draw as "test mode" and not valid for use.

results [{

Array of object:

rngStatus* {

Status code and description

code* integer Numeric error or status code.

description string Short description of the error or status code.

}

clientRecordId string Client's unique identifier for this RNG request.

recordId string Unique record identifier.

recordHandle integer Handle that can optionally be used for faster verification processing.

productId integer Identifier of the product definition used for the random generation.

productChecksum number CRC32 checksum of the product definition used for the random generation.

sets [{

Array of object:

}]

drawNumber integer

rngMatrix {

The complete RNG parameter matrix. The product definition may or may not allow overriding of these values

numberOfSets* integer >=1

How many sets of numbers to generate.

setSize integer >=1

How many numbers in each set.

minValue integer >=0

Lowest number that can be generated.

maxValue integer >=0

Highest number that can be generated.

withReplacement boolean True if generated numbers can be repeated within a set.

sortOrder enum ALLOWED:UNSORTED, ASCENDING

Sort order for the generated numbers within a set.

exclusions [integer]

distribution {

Initial distribution values and depletion type. Used for TABLE products only.

values [integer]

depletionType enum ALLOWED:NO_DEPLETION, DECREASE_ENTRY, REMOVE_ENTRY

Specifies method of depletion for the distribution values.

}

}

}]

securityBlocks [{

Array of object: Security blocks are used to authenticate various elements of the transaction. It can be an encrypted ICS block, product checksums, etc.

blockType* string Security block type identifier.

version string Format version of the security block.

keyId string Identifier of the public key needed to verify the security block, if applicable.

data string Base 64 encoded content of the security block.

}]

}

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

{

Describes the error and any specific problems with fields in the request.

code* integer Numeric error or status code.

description string Short description of the error or status code.

fieldErrors [{

Array of object: Details about a specific field error.

```
      code       string  Error code name.
      message    string  Additional detail about the error code.
      fieldName  string  Name of the field that caused the error, if applicable.
      path       string  Path to the field that caused the error, if applicable.
    }]
  }
```

1.3 POST /rng/regular-draws

Regular Draw

Conduct draws for regular draw products.

REQUEST

REQUEST BODY - application/json

```
{
  clientId                    string   Name or identifier of the client system.
  clientTime                  string   Time on the client system.
  clientBusinessDay           integer  Client's current business day. Format is YYMMDD.
  clientTransactionId         integer  Client's unique identifier for this transaction.
  testMode                    boolean  Marks the draw as "test mode" and not valid for use.
  drawUsers [{
  Array of object:  Draw participant's name and login time, to be included in the draw reports.
    name       string  Full name of the user as it should appear in the draw reports.
    loginTime  string  If included, the login time for the user will appear in the draw reports.
  }]
  returnRngMatrix             boolean  Include RNG parameter matrix in RNG results.
  rngRequests* [{
  Array of object:
    clientRecordId    string   Client's unique identifier for this RNG request.
    productId*        integer  Identifier of the product to be generated. The product defines all RNG parameters, and MAY
                               allow them to be overridden in the RNG matrix structure.
    rngMatrix {
    The complete RNG parameter matrix. The product definition may or may not allow overriding of these values
      numberOfSets*          integer  >=1
                                      How many sets of numbers to generate.
      setSize                integer  >=1
                                      How many numbers in each set.
      minValue               integer  >=0
                                      Lowest number that can be generated.
      maxValue               integer  >=0
                                      Highest number that can be generated.
      withReplacement        boolean  True if generated numbers can be repeated within a set.
      sortOrder              enum     ALLOWED:UNSORTED, ASCENDING
                                      Sort order for the generated numbers within a set.
      exclusions             [integer]
      distribution {
      Initial distribution values and depletion type. Used for TABLE products only.
        values         [integer]
        depletionType  enum     ALLOWED:NO_DEPLETION, DECREASE_ENTRY, REMOVE_ENTRY
                                Specifies method of depletion for the distribution values.
      }
    }
    clientData        [string]
    drawHash          string   A hash string that will be included in the signed data, such as the SHA-1 hash of the bets.
    drawText          string   An extra text string that will appear in the draw reports and is also appended to the draw
                               report file names.
  }]
```

securityRequests [{
Array of object: Optional requests for security blocks to be included in the response.

blockType* string Security block type identifier.
version string Format version of the security block.
}]
}

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

{
serverId* string Transaction server identifier.
serverVersion* string Transaction server version number.
transactionStatus* {
Status code and description

code* integer Numeric error or status code.
description string Short description of the error or status code.
}
clientTransactionId integer Client's unique identifier for this transaction.
timestamp string Date and time the request was processed.
businessDay integer Business day under which the transaction was recorded. Format is YYMMDD.
testMode boolean Marks the draw as "test mode" and not valid for use.
results [{
Array of object:

rngStatus* {
Status code and description

code* integer Numeric error or status code.
description string Short description of the error or status code.
}
clientRecordId string Client's unique identifier for this RNG request.
recordId string Unique record identifier.
recordHandle integer Handle that can optionally be used for faster verification processing.
productId integer Identifier of the product definition used for the random generation.
productChecksum number CRC32 checksum of the product definition used for the random generation.
sets [{
Array of object:

}]
drawNumber integer
rngMatrix {
The complete RNG parameter matrix. The product definition may or may not allow overriding of these values

numberOfSets* integer >=1
How many sets of numbers to generate.
setSize integer >=1
How many numbers in each set.
minValue integer >=0
Lowest number that can be generated.
maxValue integer >=0
Highest number that can be generated.
withReplacement boolean True if generated numbers can be repeated within a set.
sortOrder enum ALLOWED:UNSORTED, ASCENDING
Sort order for the generated numbers within a set.
exclusions [integer]
distribution {
Initial distribution values and depletion type. Used for TABLE products only.

```
        values          [integer]
        depletionType   enum        ALLOWED:NO_DEPLETION, DECREASE_ENTRY, REMOVE_ENTRY
                                    Specifies method of depletion for the distribution values.
      }
    }
    securityBlocks [{
    Array of object:  Security blocks are used to authenticate various elements of the transaction. It can be an encrypted ICS block, product checksums, etc.
      blockType*  string  Security block type identifier.
      version     string  Format version of the security block.
      keyId       string  Identifier of the public key needed to verify the security block, if applicable.
      data        string  Base 64 encoded content of the security block.
    }]
  }]
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
Describes the error and any specific problems with fields in the request.
  code*         integer   Numeric error or status code.
  description   string    Short description of the error or status code.
  fieldErrors [{
  Array of object:  Details about a specific field error.
    code        string  Error code name.
    message     string  Additional detail about the error code.
    fieldName   string  Name of the field that caused the error, if applicable.
    path        string  Path to the field that caused the error, if applicable.
  }]
}
```

1.4 GET /rng/verification

Verify an RNG record

Retrieves and verifies a previously generated RNG record.

REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*record-id	string	Search using the Unique Record Identifier (URI) returned in the RNG response.
record-handle	integer	The recordHandle returned in RNG response. May improve response time.

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  serverId*                string   Transaction server identifier.
  serverVersion*           string   Transaction server version number.
  verificationStatus {
  Status code and description
    code*       integer  Numeric error or status code.
```

description string Short description of the error or status code.

}

rngRequest {

The original RNG request.

clientId string Name or identifier of the client system.

clientTime string Time on the client system.

clientBusinessDay integer Client's current business day. Format is YYMMDD.

clientTransactionId integer Client's unique identifier for this transaction.

testMode boolean Marks the draw as "test mode" and not valid for use.

drawUsers [{

Array of object: Draw participant's name and login time, to be included in the draw reports.

name string Full name of the user as it should appear in the draw reports.

loginTime string If included, the login time for the user will appear in the draw reports.

}]

clientRecordId string Client's unique identifier for this RNG request.

productId* integer Identifier of the product to be generated. The product defines all RNG parameters, and MAY allow them to be overridden in the RNG matrix structure.

rngMatrix {

The complete RNG parameter matrix. The product definition may or may not allow overriding of these values

numberOfSets* integer >=1

How many sets of numbers to generate.

setSize integer >=1

How many numbers in each set.

minValue integer >=0

Lowest number that can be generated.

maxValue integer >=0

Highest number that can be generated.

withReplacement boolean True if generated numbers can be repeated within a set.

sortOrder enum ALLOWED:UNSORTED, ASCENDING

Sort order for the generated numbers within a set.

exclusions [integer]

distribution {

Initial distribution values and depletion type. Used for TABLE products only.

values [integer]

depletionType enum ALLOWED:NO_DEPLETION, DECREASE_ENTRY, REMOVE_ENTRY

Specifies method of depletion for the distribution values.

}

}

clientData [string]

drawHash string A hash string that will be included in the signed data, such as the SHA-1 hash of the bets.

drawText string An extra text string that will appear in the draw reports and is also appended to the draw report file names.

fileName string Name of the file used for the draw.

fileHash string SHA-1 hash of the file used for the draw.

}

rngResult {

The regenerated RNG results.

timestamp string Date and time the request was processed.

businessDay integer Business day under which the transaction was recorded. Format is YYMMDD.

rngStatus* {

Status code and description

code* integer Numeric error or status code.

description string Short description of the error or status code.

}

clientRecordId string Client's unique identifier for this RNG request.

recordId string Unique record identifier.

recordHandle integer Handle that can optionally be used for faster verification processing.

productId integer Identifier of the product definition used for the random generation.

productChecksum number CRC32 checksum of the product definition used for the random generation.

sets [{
Array of object:

}]

rngMatrix {
The complete RNG parameter matrix. The product definition may or may not allow overriding of these values

numberOfSets* integer >=1
How many sets of numbers to generate.

setSize integer >=1
How many numbers in each set.

minValue integer >=0
Lowest number that can be generated.

maxValue integer >=0
Highest number that can be generated.

withReplacement boolean True if generated numbers can be repeated within a set.

sortOrder enum ALLOWED:UNSORTED, ASCENDING
Sort order for the generated numbers within a set.

exclusions [integer]

distribution {
Initial distribution values and depletion type. Used for TABLE products only.

values [integer]

depletionType enum ALLOWED:NO_DEPLETION, DECREASE_ENTRY, REMOVE_ENTRY
Specifies method of depletion for the distribution values.

}

}

}

hsm {
Details about the cryptographic Hardware Security Module (HSM).

serial string Serial number of the HSM.

sequencer integer Sequence number on the HSM.

time string Real-time clock value on the HSM.

keyId string Identifier of the RSA key pair on the HSM.

}

extractFields [{
Array of object:

name* string Column name for the CSV extract field.

value* string Value of the extract field, formatted for CSV.

}]

}

STATUS CODE - 400: Bad Request

STATUS CODE - 404: Not Found

2. STATUS

2.1 GET /rng/status

RNG Status

Get current status of the RNG server and the configured products.

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  serverId*          string     Transaction server identifier.
  serverVersion*     string     Transaction server version number.
  serverTime*        string     Current system time of the RNG server.
  businessDay*       integer    Current business day for this RNG server. Format is YYMMDD.
  products [{
  Array of object:
    id*                  integer    Product identifier.
    name*                string     Product name.
    type*                enum       ALLOWED:INTEGER, TABLE, LOTTO
                                    The configured RNG algorithm type for the product.
    requestCount*        integer    Current total number of RNG requests for this product.
    lastRequestTime      string     The time of the last RNG request for this product.
    distribution         [integer]  <=64
    distributionOverride [integer]  <=64
  }]
  hsms [{
  Array of object:  Details about the cryptographic Hardware Security Module (HSM).
    slot*          integer  Slot number of the HSM.
    active*        boolean  Indicates the currently active HSM.
    status*        string   Current status description for this HSM.
    serial         string   Serial number of the HSM.
    sequencer      integer  Sequence number on the HSM.
    time           string   Real-time clock value on the HSM.
    keyId          string   Identifier of the RSA key pair on the HSM.
    lastErrorCode  integer  The error code of the most recent HSM error, if any.
    lastErrorTime  string   The time of the last error code, if any.
  }]
}
```