

## Quick start

### Contents

1. Introduction .....	2
2. Security .....	3
3. Resources .....	4
4. Tori Properties .....	5
5. Retrieve Tori Data .....	6
6. Write Tori Data .....	7
7. Tori Rules.....	9
8. Create Tori.....	10

## 1. Introduction

This quick start has been written to make use of the API as user friendly as possible, hard core coders can skip straight to the code examples!

The makeHistori API is based on the REST format. Currently supported are POST and GET calls. In general GET calls can be tested from a browser if (and only if) the data is public (published). Private data and POST calls require the use of an API key which can be obtained from the site. The API key must be added to the HTTP header (X-MakeHistoriApiKey) of the call – see examples listed later on how to do this.

### Two quick examples

Paste the following URL into your browser – note, if you use Google Chrome you will have to enable *view source* from the Developer options in the menus

#### **Obtain a list of the first 10 toris**

<http://www.makehistori.net/Resources?which=published&count=10>

#### **Obtain a list of the last 10 entries in the BBC news Tori**

<http://www.makehistori.net/tori/93?count=-10>

*Note: the 93 is the toriID for BBC UK News Edition found by executing the above resources end point API call. It can also be obtained by using the info button on a tori list part in the Silverlight portal,*

To try these API calls you can use

- a. CURL (try this link – <http://curl.haxx.se/>) for a command line based interface (Mac/Windows/Linux) – example here <http://forum.makehistori.com/index.php?topic=12.0>
- b. VBA – for example within Microsoft Excel (Windows) – example here <http://forum.makehistori.com/index.php?topic=11.0>
- c. Python (Mac/Windows/Linux) – example here <http://forum.makehistori.com/index.php?topic=9.0>
- d. .NET – examples here are in C# but could be done in VB.NET (Windows) – example here <http://forum.makehistori.com/index.php?topic=10.0>
- e. Ruby (Mac/Windows/Linux) – example here <http://forum.makehistori.com/index.php?topic=13.0>

## 2. Security

“Or how to keep data under lock and key”

There are 3 aspects to controlling the use of data within makeHistori

a. The API key

The API key is added to the HTTP header X-MakeHistoriApiKey when a call is made to the web service where the user needs to be identified. This is similar to logging into the portal using your username and password. The API key can be obtained either from the portal page by selecting the profile button or from the “Get API Key” menu on the home page (<http://www.makehistori.net/mhiUserDetails.aspx>). Passing in this key uniquely identifies you to the system and must be kept secret, just like your password. The subsequent sections in this document indicate when the API key is required, but in general it is required whenever data is not publically available or when writing data to a tori.

b. Published

A tori can be ‘published’ which means that it can be found whenever a search is performed. This doesn’t change any access rights to the data in a tori – it is similar to not having your phone number listed in a phone directory.

c. Permissions

Each tori can have permissions for access to it’s data. Currently for the API access this is set when the tori is created and is set for all users apart from the owner of the tori. The portal also allows giving named users different access rights – this will be added to the API in the future. The access rights are as follows (see the Create tori section for how to apply these values)

0: no access – i.e. other users can neither read nor write data to this tori

1: readonly – i.e. other users can read the data from this tori

5: read/write – i.e. other users can both read and write data to this tori

### 3. Resources

“Or how to Find what’s in makeHistori”

This is the RESOURCES end point in the web service, which provides meta data (descriptive information) on the toris and associated data within makeHistori.

The general form of the call is

<http://www.makehistori.net/resources?which=mine>

“*which*” provides information just for the toris related to the users API key (note this will not work from a browser as the API key is required)

or

<http://www.makehistori.net/resources?which=published>

“*which*” provides information for all published toris within makeHistori – this does not require an API key so can be executed from the browser

Additional options which can be added to the call are:-

- a. Count – number of items to return, specifying a negative number returns items in reverse order – i.e. most recent item first
- b. Format – specifies the return format – any of text, json or xml
- c. Skip – allows paging of data, so skip=100&count=10 would return the 101<sup>st</sup> tori thru to the 110<sup>th</sup> tori.
- d. Search – allows simple text search within the tori name, this has to be url encoded (i.e. spaces converted to +) See [http://www.w3schools.com/TAGS/ref\\_urlencode.asp](http://www.w3schools.com/TAGS/ref_urlencode.asp) which also includes a utility for performing encoding.

Example

<http://www.makehistori.net/resources?which=published&search=bbc> – retrieves all toris containing BBC in the name

## 4. Tori Properties

“Or what sort of things does this tori store”

(coming soon)

## 5. Retrieve Tori Data

“Or how to read the data in the tori”

The general form to GET data from a tori is as follows:-

<http://www.makehistori.net/tori/toriID>

where *toriID* is the number of the tori (you can obtain this number by using the info button on a tori list part in the Silverlight portal, or by executing an API call to the resources end point as mentioned previously.)

Additional optional switches which can be added to the call are:-

- a. From – from date/time (see later for valid date time formats)
- b. To – to date/time (see later for valid date time formats)
- c. Count – number of items to return, specifying a negative number returns items in reverse order – i.e. most recent item first
- d. Sampled – returns <count> samples from a long time, for instance if `sampled=true&count=2` the first and last readings are return; if `sample=true&count=3` the first, middle and last reading are returned. This is useful when producing overview charts of numerical data
- e. Quiet – reduces information messages
- f. Format – specifies the return format – any of text, json or xml (default is xml)
- g. Kwhoover – performs an integration function, used for power readings only

For example

<http://www.makehistori.net/tori/93?count=-10&format=text> – retrieves the last 10 items from the BBC news in text format

<http://www.makehistori.net/tori/93?sampled=true&count=2> – retrieves the first and last value from the google stock ticker

## 6. Write Tori Data

“Or how to write data to a tori”

This requires the use of an API key so cannot be executed from the browser.

To prevent over use of our server resources don't upload more often than once every 5 minutes. We hope to increase these rate soon.

There are two basic forms for writing.

- i) Execute a POST against the general form as follows to write a single value:-

<http://www.makehistori.net/tori/toriID>

where *toriID* is the number of the tori (you can obtain this number by using the info button on a tori list part in the Silverlight portal, or by executing an API call to the resources end point as mentioned previously.)

Using the following switches:-

- a. value - value to be stored (required)
  - b. dts - date time stamp - either in "Normal", ISO format or Excel datetime number, see DTSNote below (optional – if omitted stored at now, which is server time not local time)
  - c. dtsoffset - time offset from Universal (Zulu or GMT) time - e.g. "07:00" for PST or "-01:00" for CET, see DTS Offset notes below (optional)
  - d. dtsformat - the only valid value is excel - don't include for none Excel dates (optional)
- ii) Execute a POST against the general form as follows to write multiple values against multiple toris:-

<http://www.makehistori.net/command/multiwritexform>

- a. toriid-<n> - where <n> is a number 0..n to line up with dts-<n> and value-<n> - the value is toriid
- b. dts-<n> - date/time stamp - normal, ISO or Excel
- c. value-<n> - value
- d. dtsformat - the only valid value is excel - don't include for none Excel dates
- e. dtsoffset - time offset from Universal (Zulu or GMT) time - e.g. "07:00" for PST or "-01:00" for CET

### DTSNote

We record and use the timezone offset within MakeHistori

Examples of dts

RFC3339 / ISO (URL encoded)

dts=2009-04-12T11%3A06%3A32&value=99

(note %3A is encoding of ':' - so time in above is 11:06:32)

RFC3339 / ISO (with offset) (URL encoded)

dts=2009-04-12T11%3A06%3A32%2B-07%3A00&value=99

Excel

dts=39915.5229166667&dtsformat=excel&value=102

Excel with offset

dts=39915.5229166667&dtsformat=excel&dtsoffset=-07:00&value=102

Normal error prone - in the USA this would be 4th December, in Europe it would be 12th April...

dts=12/4/2009 12:34:00&value=102

Normal

dts=12 Apr 2009 19:30:03&value=103

Normal with offset

dts=12 Apr 2009 19:30:03 -05:00&value=104

## 7. Tori Rules

“Or how to write and read at the same time while being picky”

(Coming soon)

## 8. Create Tori

“Or how to make a new tori”

This requires the use of an API key so cannot be executed from the browser.

The general form to create a tori is as follows:-

<http://www.makehistori.net/command/createtori>

with the following additional attributes:-

- a. name - tori name, must be unique for your account
- b. worldpermissions – this specifies the access rights to the tori for everyone else 0: no access  
1: readonly 5: read/write
- c. description – a (url encoded) short description for the tori
- d. publish – specifies if this can be “discovered” (i.e. searchable) by other units. if T,t or 1 then it is world searchable
- e. dataformat – the type of data the tori is storing – text,numeric or power
- f. properties, this isn’t a switch but named properties can be specified when creating a tori.  
More information will be added to this topic later

If the create is successful then the new tori id is returned (numeric) – otherwise a message prefixed with + is returned.

An example using curl (this should all be on one line), substitute <API key> with your own api key and <toriname> with the name you wish to give the tori

```
\curl --header "X-MakeHistoriApiKey: <API key>" --data "name=<toriname>
&worldpermissions=5&dataformat=power&lowerrange=0&upperrange=9999"
"http://www.makehistori.net/Command/CreateTori"
```