XTools

Release 3.1.7

Oct 06, 2017

Table of Contents

XTools is a suite of statistics tools for MediaWiki wikis, users, pages, and more. It is in operation for Wikimedia wikis and can also be installed for any MediaWiki installation.

Quick links:

- Demonstrations:
  - Wikimedia installation: xtools.wmflabs.org
  - Development installation: xtools-dev.wmflabs.org
- This documentation: xtools.readthedocs.io
- Source code: github.com/x-tools/xtools
- Issue tracker: phabricator.wikimedia.org
- IRC: #wikimedia-xtools on Freenode

CHAPTER 1

Pre-requisites

XTools requires the following to run:

- A recent version of Linux (Windows servers are supported, however; you must enable the `app.load_stylesheets_from_cdn` if you want it to look nice).
- PHP 5.5.9+ (not tested on PHP7)
  - JSON must be enabled.
  - ctype needs to be enabled
  - You must have `date.timezone` set in `php.ini`.
  - PDO including the driver for the database you want to use
  - Curl must be enabled.
- Composer 1.0.0+
- Node and npm (tested with versions 6.2.1 and 3.9.3, respectively)

Databases

1. One or more project databases. These should be current mediawiki installations. The meta database should point to them.
2. A Meta database. If you are running more than one wiki (`app.is_single_wiki` set to false), information on each wiki must be stored in a meta database. XTools uses one modeled after the WMF Labs database..

   This database must live on the same machine as the project databases.

   See the *installation documentation* for more details if you don't already have this database available.
3. An optional Tools' database, where other MediaWiki tools store their data.

CHAPTER 2

Installation

To install XTools, please follow these steps:

1. Download the repository into a web-accessible location. If you're using Git, the `master` branch is always stable so you can clone that.
2. Ensure that `var/` and all files within it (other than `var/SymfonyRequirements.php`) are writable by the web server.
3. Run `composer install` and be prompted to enter database details and other configuration information.
4. Open XTools in your browser; you should see the XTools landing page.

To update the cache after making configuration changes, run `./bin/console cache:clear`.

Single wiki

To use XTools for a single wiki, set the following variables in `parameters.yml`:

- `app.single_wiki` to `true`
- `wiki_url` to the full URL of your wiki
- `api_path` to the path to the root of your wiki's API

Wiki family

To use XTools for a family of wikis, set `app.single_wiki` to `false` in `parameters.yml`.

You will also need to create a new database table to record the meta information about your wikis. It can live wherever you want; just set the `database_replica_*` variables accordingly in `parameters.yml`.

The table must be called `wiki` and have the following structure:

```
CREATE TABLE `wiki` (
    `dbname` varchar(32) NOT NULL PRIMARY KEY,
    `lang` varchar(12) NOT NULL DEFAULT 'en',
    `name` text,
    `family` text,
    `url` text
);
```

(The WMF version of this table can be browsed at Quarry #4031.)

CHAPTER 3

Configuration

As part of the installation of XTools, `composer install` or `composer update` may prompt you for configuration options. This is a definition of those options.

Databases

XTools' own database:

- **database_host** - Hostname for the server with the XTools database
- **database_port** - Port for the server with the XTools database
- **database_name** - Database name of the XTools database
- **database_user** - Username for the XTools database
- **database_password** - Password for the user for the XTools database

The projects' databases:

- **database_replica_host** - Hostname for the server with the MediaWiki databases
- **database_replica_port** - Port for the server with the MediaWiki databases
- **database_replica_name** - Database name of any one of the MediaWiki databases (usually the default, or the 'meta'; it doesn't matter which).
- **database_replica_user** - Username for the MediaWiki databases
- **database_replica_password** - Password for the user for the MediaWiki databases

The 'meta' database:

- **database_meta_name** - Database Name for the server with the meta_p table (this is not required if `app.single_wiki` is set)

Other tools' database (e.g. checkwiki):

- **database_toolsdb_host** - MySQL host name

- **database_toolsdb_port** - MySQL port number
- **database_toolsdb_name** - Username to connect as
- **database_toolsdb_password** - Password to use for the user

Authentication and Email

The Oauth details need to be requested from `Special:OAuthConsumerRegistration` on your default wiki.

- **oauth_key** - Oauth consumer key
- **oauth_secret** - Oauth consumer secret
- **mailer_transport** - Software for the mailer
- **mailer_host** - Hostname for the mailer
- **mailer_user** - Username for the mailer software
- **mailer_password** - Password for the mailer software

Application

- **secret** - A secret key that's used to generate certain security-related tokens, and as the secret for the internal API. If you are using a separate API server (as explained in the *administration* section), this parameter must have the same value on both the app server and API server.
- **app.noticeDisplay** - Display the notice or not
- **app.noticeStyle** - Style of the notice banner. Available options: "error," "warning," "succeess," "info."
- **app.noticeText** - Message shown to the user. If you provide a valid intuition key, it will display that message instead
- **app.replag_threshold** - Number of seconds to consider the replicas as "lagged", and show a warning to the user that the data may be out of date
- **app.load_stylesheets_from_cdn** - Whether to load our stylesheets and scripts from a CDN. This is required if XTools is installed on a Windows server
- **app.single_wiki** - Point XTools to a single wiki, instead of using a meta database. This ignores database_meta_name above.
- **app.is_labs** - Whether XTools lives on the Wikimedia Foundation Labs environment. This should be set to false.
- **app.rate_limit_time** - Number of minutes during which `app.rate_limit_count` requests from the same user are allowed. Set this to `0` to disable rate limiting.
- **app.rate_limit_count** - Number of requests from the same user that are allowed during the time frame specified by `app.rate_limit_time`. Set this to `0` to disable rate limiting.
- **app.multithread** Set to 1 to speed up the Edit Counter and other tools by making multiple asynchronous queries. This requires a multithreaded server (such as Apache), so you should set this to 0 if you are using the default Symfony server in your development environment. It may also be possible to forward all requests to `/api` to a dedicated API server. See the *administration* section for more. You must also set the `app.base_path` parameter for multithreading to work.

- **app.base_path** The base URL of your XTools installation, including the protocol. This parameter is required if `app.multithread` is turned on.
- **app.max_page_revisions** - Set a maximum number of revisions to process for pages. This is to safeguard against unnecessarily consuming too many resources for queries that will most surely timeout. Set this to *0* to disable all limitations.
- **app.max_user_edits** - Querying a user that has more edits than this will be rejected. This is to safeguard against unnecessarily consuming too many resources for queries that will most surely timeout. Set this to *0* to disable all limitations.
- **wiki_url** - URL to use if app.single_wiki is enabled. The title of pages is attached to the end.
- **api_path** - The API path for the project, usually /w/api.php
- **opted_in** - A list of database names of projects that will display *restricted statistics* regardless of individual users' preferences

Tools

- **enable.ec** - Enable "Edit Counter" tool
- **enable.articleinfo** - Enable "Article Information" tool
- **enable.pages** - Enable "Pages Created" tool
- **enable.topedits** - Enable "Top Edits" tool
- **enable.blame** - Enable "Article Blamer" tool
- **enable.autoedits** - Enable "Automated Edits" tool
- **enable.adminstats** - Enable "Admin Statistics" tool
- **enable.adminscore** - Enable "Admin Score" tool
- **enable.rfa** - Enable "RfX Analysis" tool
- **enable.rfavote** - Enable "RfX Vote Calculator" tool
- **enable.bash** - Enable "Quote Database" tool
- **enable.sc** - Enable "Plain, Dirty, Simple Edit Counter" tool
- **enable.es** - Enable "Edit Summaries" tool

CHAPTER 4

Opting in to restricted statistics

Some statistics are considered private by some users, such as the times of the day or year that they edit most or the pages they've made most contributions to.

Although the data for these statistics is made available via MediaWiki's API, users must explicitly opt in to make the aggregate forms available in XTools. Alternatively, a whole project can be opted in via the `opted_in` *configuration variable*.

The affected tools are as follows:

- *Edit Counter*:
  - Monthly counts bar chart
  - Timecard punch chart
- *Top Edits*:
  - Top edits per namespace

How to opt in

To opt in, a user must create `User:<username>/EditCounterOptIn.js` on each wiki they want to opt in for. This page should be created with any content (it just has to have *some* content).

To opt in on all projects, they must create `User:<username>/EditCounterGlobalOptIn.js` on the default project (or, in the case of the WMF Labs installation, on Meta Wiki). Again, the actual content of this page is irrelevant.

How to opt out

To opt out the relevant user page (single-wiki or global; see above) should be blanked or deleted.

CHAPTER 5

Tools

Edit Counter

The edit counter tool provides detailed summary statistics about a single user on a single project.

General Statistics

The general statistics section contains lots of statistics about the user and their work on the project, as well as some data about other projects that they're active on.

Firstly, some basic **user information**: ID, username, and group membership (including globally, if CentralAuth is installed).

Then, **Edit counts** are displayed for:

- the last day, week, month, year, and all time (the latter also including addition counts of deleted edits);
- edits made with or without comments;
- edits that have been deleted;
- small (under 20 bytes) and large (over 1000 bytes) edits;
- minor/non-minor edits (as recorded by the user); and
- what semi-automating tools they used to edit.

Also, dates of activity on the project (earliest and latest) are displayed, and what this duration is in days.

Averages (per day) are given for some of the above metrics.

Next, **Page counts** are shown:

- pages created (note that this shows *all* pages created, including those created as redirects during a page move; the *Pages Created* tool excludes these);
- pages imported, moved, deleted, and undeleted;

- total number of unique pages edited.

And lastly, **Log counts** are summarized:

- the number of times the user has thanked another user;
- pages reviewed, patrolled, protected, and unprotected;
- users blocked and unblocked;
- files uploaded (and also those uploaded to Commons, for the WMF Labs installation).

Namespace totals

Total edit counts in each namespace (from all time): a table ordered in decreasing number of edits; and a pie chart showing the relative number of edits.

Timecard

A 'punchcard' chart showing what days of the week and hours of the day the user made most edits. The times given are in UTC.

Year counts

A bar chart showing total edit counts made in each year, with each bar being divided into namespace sections so that it's possible to get an idea of how a user's namespace activity has changed over the years.

Month counts

The same as the year counts, except the columns are months instead of years.

Latest global edits

A list of the user's thirty most recent edits from all projects.

Automated edits

A summary table of the number of edits the user has made with any of the known semi-automated editing tools, sorted in decreasing order.

Page History

The Page History tool, also known as "ArticleInfo", provides detailed statistics about the revision history of a page.

General Statistics

The general statistics section contains an overview of the statistics of the page. This includes basic figures like the page size, number of editors, types of editors, number of edits, and various averages.

On WMF wikis, the "Wikidata ID" field also shows the number of *sitelinks*. This figure refers to the number of sister projects that have a page about the same subject.

For supported projects on WMF wikis, you may see additional information such as the *assessment* of the page, pageviews and the number of *bugs*.

Beneath the numerical statistics are three charts. The first shows the number of edits made by registered accounts compared to logged out users (IPs). The second chart shows the number of edits that were marked as minor compared to major edits (not marked as minor). The last chart shows the number of edits made by the top 10% of all editors to that page, compared to the bottom 90%. The *top editors* are ranked by the amount of content they've added to the page.

Top editors

The top editors section shows various information about users and bots who have edited the page. There are two pie charts comparing the top editors by *number of edits* and by *added text*. XTools does not count bot accounts as a top editor. Instead, they are listed in the *bot list* table.

By number of edits

The *Top 10 by edits* chart compares the number of edits each top editor made. The percentages shown in parentheses refer to the number of edits the user made in relation to total number of edits made to the page.

By added text

Added text refers to any positive addition of content that was not reverted with the next edit. This is because users who fight vandalism (for instance) will otherwise appear to have added a lot of content to a page, when in actuality they just undid an edit that removed a lot of content. Going by edits that weren't reverted, we have a better idea of the users who made meaningful contributions.

Note however that the Page history tool only detects reverts if it happened with the very next edit, and not a later edit.

The "Top 10 by added text" pie chart compares each of the 10 top editors. The percentages shown in parentheses refer to the amount of content that user added compared to all content that was added to the page.

Top editors table

The first table shown lists the top editors (non-bots) and various statistics about their contributions to the page. The last two columns show specialized calculations. *Average time between edits* (atbe) is the average number of days between each of the user's edits to the page. This is starting with the date of their first edit and the date of their last edit to the page. *Added (bytes)* refers to the number of bytes of text the user added to the page.

By default only the first 20 editors are shown. You can expand to show all editors using the link on the bottom row of the table.

You can also export this data as wikitext using the link just above the table.

Bot list

The "Bot list" shows lists all of the bots that edited the page, ranked by edit count. A message is shown indicating if the bot is no longer a bot, and links to the account's user rights log.

The list is by default limited to the top 10 bots. You can expand to show all bots using the link on the bottom row of the table.

Year counts

This section breaks down editing activity by each year.

The chart compares the number of edits, IP edits and minor edits over time. The yellow line represents the total size of the article as it changed over time (the right Y-axis denotes the values).

The table lists various statistics for each individual year. *Log events* shows which logged events occurred during that year. The types of events XTools looks for include deletions (e.g. page was deleted then restored), page moves, protections that were applied, and stable settings (also known as pending changes protection).

Month counts

This section breaks down editing activity by each month. There is a small graph shown for each month, which compares the number of total edits made to IP edits and minor edits.

(Semi-)automated edits

This lists all the known (semi-)automated tools that were used to edit the page. For more information on how this works, see the documentation on the AutoEdits tool.

Assessments

Some WMF wikis have a system of rating the quality of a page, known as an "assessment". This section lists any known assessments of the page from each WikiProject, based on PageAssessments data.

Bugs

This section lists any issues with the page that were automatically detected. This includes missing basic Wikidata, such as the description, and _CheckWiki errors. For both, a table is shown explaining each issue and how to fix it. The "priority" indicates how important it is to fix the given issue according to CheckWiki, where 1 is the highest priority. "Notice" indicates where in the wikitext the issue lies.

Pages Created

This tool provides information about pages that a given user has created.

Top Edits

This tool queries a single project and displays

- a user's most-edited articles in one or all namespaces; or
- all of a user's edits on one article (in chronological order).

Admin Score

The Admin Score tool is intended to give a very brief overview of how admin-worthy a user is. This tool was originally developed by ScottyWong for use on the English Wikipedia.

Algorithm

AdminScore takes the following factors into account

Activity	Multiplier
Account Age (days)	1.25
Edit Count	1.25
Has user page	1
Page Patrols	1
Blocks applied	1.4
Participation in AFDs	1.15
Recent activity (730 days)	0.9
Participation at AIV	1.15
Use of edit summaries	0.8
Namespaces	1
Pages Created (live)	1.4
Pages Created (deleted)	1.4
Participation at RFPP	1.15

All factors are capped at 100, making a total possible admin score 1300.

Bash Quote

The Bash Quote tool contains humorous quotes about MediaWiki development and general software development wisdom. Bash pulls all quotes from `app/quote.yml`, which means you can replace the quotes with any relevant to your wiki.

Inside the Bash Quote tool, there is a "random" link which will give a random quote from the database. There's also the ability to search based on a quote ID. Or, if you'd rather have humour in large batches, there is an "All" button which shows all of the quotes currently in the database.

Simple Counter

The Simple Counter is a quicker way than *Edit Counter* to get a brief overview of a user's contributions.

It displays a user's total number of edits (live, deleted, and a grand-total), as well as their username, ID, and group membership.

CHAPTER 6

Development

To contribute to the development of XTools, you may fork us on GitHub. A few things to be aware of first:

1. XTools is based on Symfony 3. We use Twig as our template engine. Symfony is a full MVC system. a. The controllers are located at `src/AppBundle/controller`. They are sorted by tool. b. The twig templates are located at `app/resources/views`. They are sorted by tool.
2. We use the `@Route` syntax to configure routes.
3. Every tool requires a twig directory and one controller. Also, core parts of XTools require the tool to be registered within *app/config/tools.yml*.

Style Guidelines

- It's called "XTools", with two capital letters.
- XTools conforms to PSR2 coding standards; use `./vendor/bin/phpcs` to check your code.
- Functions and routes must begin with the tool name.
- Version numbers follow Semantic Versioning guidelines.

Running Development server

First make sure you meet the *Pre-requisites*, and then follow these steps:

1. Clone the repository: `git clone https://github.com/x-tools/xtools.git && cd xtools`
2. Run `composer install` and answer all the prompts.
3. Create a new local database: `./bin/console doctrine:database:create` (or `d:d:c`).
4. Run the database migrations: `./bin/console doctrine:migrations:migrate` (or `d:m:m`)

5. Launch PHP's built-in server: `./bin/console server:run` (or `s:r`).
6. Visit `http://localhost:8000` in your web browser.
7. You can stop the server with `./bin/console server:stop` (or `s:s`)

The *Simple Counter* is the simplest tool and should work as soon as you set up XTools. Test it by going to http://localhost:8000/sc and put in `Jimbo Wales` as the Username and `en.wikipedia.org` as the Wiki. After submitting you should quickly get results.

The development server does not cache data; any changes you make are visible after refreshing the page. When you edit the `app/config/parameters.yml` file, you'll need to clear the cache with `./bin/console c:c`.

Assets can be dumped with `./bin/console assetic:dump`, and if you're actively editing them you can continually watch for changes with `./bin/console assetic:watch`.

The logs are in `var/logs/dev.log`. If things are acting up unexpectedly, you might try clearing the cache or restarting the server.

Developing against WMF databases

If you want to use the WMF database replicas, open two tunnels with:

```
ssh -L 4711:enwiki.labsdb:3306 tools-login.wmflabs.org -N -l your-username-here
ssh -L 4712:tools.labsdb:3306 tools-login.wmflabs.org -N -l your-username-here
```

And set the following in `app/config/parameters.yml`:

```
app.is_labs: 1
database_replica_host: 127.0.0.1
database_replica_port: 4711
database_replica_name: meta_p
database_meta_name: meta_p
database_replica_user: your-uxxxx-username-here
database_replica_password: your-password-here
database_toolsdb_host: 127.0.0.1
database_toolsdb_port: 4712
database_toolsdb_name: toollabs_p
```

(Change the `your-*-here` bits to your own values, which you can find in your `replica.my.cnf` file on Tool Labs.)

Table mappings

Tool Labs has different versions of tables that utilize indexing to improve performance. We'll want to take advantage of that.

- Go to the config directory with `cd app/config`
- Create the file table_map.yml from the template: `cp table_map.yml.dist table_map.yml`
- Set the contents of the file to the following:

```
parameters:
  app.table.archive: 'archive_userindex'
  app.table.revision: 'revision_userindex'
  app.table.logging: 'logging_logindex'
```

Sometimes we want <code>logging_userindex</code> and not the logindex. This is handled in the code via the <code>getTableName()</code> function in [https://github.com/x-tools/xtools/blob/master/src/Xtools/Repository.php#L144 Repository.php].

Caching

Caching should happen in helpers, with appropriate times-to-live.

Every helper should extend HelperBase, which has `cacheHas()`, `cacheGet()`, and `cacheSave()` methods. These should be used in this pattern:

```
public function doSomething($input)
{
    $cacheKey = 'something.'.$input;
    if ($this->cacheHas($cacheKey)) {
        return $this->cacheGet($cacheKey);
    }
    $something = 'big query here';
    $this->cacheSave($cacheKey, $something, 'P1D');
    return $something;
}
```

The cache key can be anything, so long as it is unique within the current class (the `cache*()` methods prepend the classname, so you don't have to). The TTL syntax is from the DateInterval class (e.g. `P1D` is one day, `PT1H` is one hour).

The above methods are just wrappers around a PSR-6 implementation, intended to reduce the repetition of similar lines of code. You can, of course, retrieve the underlying CacheItemPoolInterface whenever you want with `$container->get('cache.app')`.

Writing the docs

We use ReadTheDocs; it's great.

To build this documentation locally, you need `python-sphinx` installed, as well as the `sphinx_rtd_theme` theme.

Then, it's simply a matter of runnign `make html` in the `docs/` directory, and browsing to `xtools/docs/_build/html/` to view the documentation.

Documentation sections use the following (standard Python) hierarchy of section symbols:

- # with overline for parts
- * with overline for chapters
- = for sections
- - for subsections

Releases

Before tagging a new release:

- update the version numbers in `docs/conf.py` and `app/config/version.yml`;

- check the copyright year in `README.md`, `docs/conf.py`, and `app/Resources/views/base.html.twig`;
- if assets were modified, bump the version number in config.yml under framework/assets/version;
- update the branch of the stable build status badge in `README.md`; and
- update `RELEASE_NOTES.md` with any notable new information for the end user.

Then tag the release (follow the Semantic Versioning guidelines, and annotate the tag with the above release notes) and push it to GitHub.

Lastly, update the `version` and `updated` parameters at https://www.mediawiki.org/wiki/XTools

Additional Help

Please contact User:Matthewrbowker or User:MusikAnimal if you need help. Or, you are welcome to visit us on IRC (Direct link - Requires an IRC client).

CHAPTER 7

Administration

Once you have XTools up and running, depending on how much traffic you receive, you might want to implement measures to ensure stability.

Adding rate limiting

Rate limiting can safeguard against spider crawls and bots that overload the application.

To configure, set the following variables in `parameters.yml`:

- `app.rate_limit_time: 10` where `10` is the number of minutes `app.rate_limit_count` requests from the same user to the same URI are allowed.
- `app.rate_limit_count: 5` where `5` is the number of requests from the same user that are allowed during the time frame specified by `app.rate_limit_time`.

Using the above example, if you try to load the same page more than 5 times within 10 minutes, the request will be denied and you will have to wait 10 minutes before you can make the same request. This only applies to result pages and the API, and not index pages. Additionally, no rate limitations are imposed if the user is authenticated.

Any requests that are denied are logged at `var/logs/rate_limit.log`.

You can blacklist user agents and URIs using the request_blacklist.yml file.

Offloading API requests

XTools features a rich public API. In addition, the internal API used for the Edit Counter can be very expensive in terms of resources. If you expect your XTools installation will receive a lot of traffic, you might consider setting up a dedicated API server so that resources on the main app server are not hogged.

This documentation covers how to set up forwarding so that all requests to /api go to the API server, assuming you are using Apache in a Linux environment.

1. Install libapache2-mod-proxy-html and libxml2-dev: `sudo apt-get install libapache2-mod-proxy-html libxml2-dev`
2. Enable the necessary modules (if some are already enabled it will simply make sure they are active):

```
sudo a2enmod proxy
sudo a2enmod proxy_http
sudo a2enmod proxy_ajp
sudo a2enmod rewrite
sudo a2enmod deflate
sudo a2enmod headers
sudo a2enmod proxy_balancer
sudo a2enmod proxy_connect
sudo a2enmod proxy_html
sudo a2enmod xml2enc
```

3. In your Apache coniguration, within the `<VirtualHost>` block, add this to the bottom:

```
ProxyPreserveHost On
ProxyPass /api http://X.X.X.X:80/app.php/api
ProxyPassReverse /api http://X.X.X.X:80/app.php/api
```

   ...replacing `X.X.X.X` with the IP of the API server.

4. Finally, restart apache with `sudo apachectl -k graceful`

Killing slow queries

Some queries on users with a high edit count may take a very long time to finish or even timout. You may wish to add a query killer to ensure stability.

If you are running on a Linux environment, consider using pt-kill. A query killer daemon could be configured like so:

```
pt-kill --user=xxxx --password=xxxx --host=xxxx \
       --busy-time=90 \
       --log /var/www/web/killed_slow_queries.txt \
       --match-info "^(select|SELECT|Select)" \
       --kill --print --daemonize --verbose
```

This will kill any SELECT query that takes over 90 seconds to finish, and log the query at `/var/www/web/killed_slow_queries.txt`.

Note that pt-kill requires libdbi-perl and libdbd-mysql-perl.

CHAPTER 8

API

A public API is available for getting basic statistics. Below is each avaiable endpoint. All data is returned as JSON, in addition to other formats as noted.

Project API

API endpoints related to a project.

Normalize project

`GET /api/project/normalize/{project}`

Get the URL, database name, domain and API path of a given project.

Parameters:

- `project` (**required**) - Project domain or database name.

Example:

Basic access information about the English Wikipedia.

> https://xtools.wmflabs.org/api/project/normalize/enwiki https://xtools.wmflabs.org/api/project/normalize/en.wikipedia https://xtools.wmflabs.org/api/project/normalize/en.wikipedia.org

Namespaces

`GET /api/project/namespaces/{project}`

Get the localized names for each namespace of the given project. The API endpoint for the project is also returned.

Parameters:

- `project` (**required**) - Project domain or database name.

Example:

Get the namespace IDs and names of the German Wikipedia.

https://xtools.wmflabs.org/api/project/namespaces/dewiki https://xtools.wmflabs.org/api/project/namespaces/de.wikipedia https://xtools.wmflabs.org/api/project/namespaces/de.wikipedia.org

User API

API endpoints related to a user.

Automated edit counter

`GET /api/user/automated_editcount/{project}/{username}/{namespace}/{start}/{end}/{offset}/{tools}`

Get the number of (semi-)automated edits made by the given user in the given namespace and date range. You can optionally pass in `?tools=1` to get individual counts of each (semi-)automated tool that was used.

Parameters:

- `project` (**required**) - Project domain or database name.
- `username` (**required**) - Account's username.
- `namespace` (**required**) - Namespace ID or 'all' for all namespaces.
- `start` - Start date in the format `YYYY-MM-DD`. Leave this and `end` blank to retrieve the most recent data.
- `end` - End date in the format `YYYY-MM-DD`. Leave this and `start` blank to retrieve the most recent data.
- `tools` - Set to any non-blank value to include the tools that were used and thier counts.

Example:

Get the number of (semi-)automated edits made by Jimbo Wales on the English Wikipedia.

https://xtools.wmflabs.org/api/user/automated_editcount/en.wikipedia/Jimbo_Wales

Get a list of the known (semi-)automated tools used by Jimbo Wales in the mainspace on the English Wikipedia, and how many times they were used.

https://xtools.wmflabs.org/api/user/automated_editcount/en.wikipedia/Jimbo_Wales/0///1

Non-automated edits

`GET /api/user/nonautomated_edits/{project}/{username}/{namespace}/{start}/{end}/{offset}`

Get non-automated contributions for the given user, namespace and date range.

Parameters:

- `project` (**required**) - Project domain or database name.
- `username` (**required**) - Account's username.
- `namespace` (**required**) - Namespace ID or 'all' for all namespaces.
- `start` - Start date in the format `YYYY-MM-DD`. Leave this and `end` blank to retrieve the most recent contributions.

- `end` - End date in the format `YYYY-MM-DD`. Leave this and `start` blank to retrieve the most recent contributions.
- `offset` - Number of edits from the start date.

Example:

Get the newest non-automated mainspace contributions made by Jimbo Wales on the English Wikipedia.

https://xtools.wmflabs.org/api/user/nonautomated_edits/en.wikipedia/Jimbo_Wales/0

Page API

API endpoints related to a single page.

Article info

`GET /api/page/articleinfo/{project}/{article}/{format}`

Get basic information about the history of a page.

Parameters:

- `project` (**required**) - Project domain or database name.
- `article` (**required**) - Full page title.

Example:

Get basic information about Albert Einstein.

https://xtools.wmflabs.org/api/page/articleinfo/en.wikipedia.org/Albert_Einstein

CHAPTER 9

The tools

Here is a brief overview of all of the tools, with links to more detailed information. See the main menu in the side bar for more.

Edit Counter

Edit Counter provides summary information about a user and their activity on a project, such as the total numbers of certain types of edits; their most-edited pages; what semi-automating tools they've used to edit; and lots more. *Read more about Edit Counter...*.

Admin Score

Find out how admin-worthy a user is. *Read more about Admin Score...*.

Admin Stats

Statistics about administrators' actions. Read more about Admin Stats....

Article Info

Get various statistics about the history of a page. *Read more about Article Info...*.

Auto Edits

Explore the edits made by various semi-automated editing tools, from the point of view of pages or of users. Read more about Auto Edits....

Bash

A collection of humourous or insightful quotations about MediaWiki. *Read more about Bash...*.

Blame

Find out who last changed a given part of a page. (Blame is currently not implemented.) Read more about Blame....

Pages

Information about pages that have been created by a user. *Read more about Pages...*.

RFX

RFX Read more about RFX....

RFX Vote

@TODO Read more about RFAVote....

Simple Counter

A simpler but quicker way to view edit counts (than Edit Counter, above). *Read more about Simple Counter...*.

Top Edits

View the pages that a user has edited most often, or all of their edits on one page. *Read more about Top Edits...*.

CHAPTER 10

Help

For more help with XTools, there are several places you can ask:

- IRC (direct link requires an IRC client) — to chat with the developers and other users.
- Phabricator — if you've found a bug.