XTools

Release 3.1.25

Jan 12, 2018

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

Tools

1.1 Edit Counter

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

1.1.1 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).

1.1.2 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.

1.1.3 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.

1.1.4 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.

1.1.5 Month counts

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

1.1.6 Latest global edits

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

1.1.7 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.

1.2 Page History

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

1.2.1 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.

1.2.2 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.

1.2.3 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).

1.2.4 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.

1.2.5 (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.

1.2.6 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.

1.2.7 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.

1.3 Pages Created

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

1.4 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).

1.5 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.

1.5.1 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.

1.6 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.

1.7 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 2

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.

2.1 Project API

API endpoints related to a project.

2.1.1 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

2.1.2 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

2.1.3 Admins and user groups

`GET /api/project/admins_groups/{project}`

Get a list of users who are admins, bureaucrats, CheckUsers, Oversighters, or stewards of the project and list which of these user groups they belong to.

Parameters:

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

Example:

Get administrative users of the French Wikipedia:

https://xtools.wmflabs.org/api/project/admins_groups/frwiki https://xtools.wmflabs.org/api/project/admins_groups/fr.wikipedia.org

2.1.4 Admin statistics

`GET /api/project/adminstats/{project}/{days}`

Get users of the project that are capable of making 'admin actions', along with various stats about the actions they took. Time period is limited to one month.

Parameters:

- `project` (**required**) - Project domain or database name.
- `days` - Number of days before present to fetch data for (default 30, maximum 30).

Example:

Get various statistics about actions taken by admins of the French Wikipedia over the past week:

https://xtools.wmflabs.org/api/project/adminstats/frwiki/7 https://xtools.wmflabs.org/api/project/adminstats/fr.wikipedia.org/7

2.2 User API

API endpoints related to a user.

2.2.1 Simple edit count

`GET /api/user/simple_editcount/{project}/{username}`

For the given account, get the user ID, live and deleted edit count, local user groups and global user groups.

Parameters:

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

- `username` (**required**) - Account's username.

Example:

Get basic statistics about Jimbo Wales on the English Wikipedia.

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

2.2.2 Number of pages created

`GET /api/user/pages_count/{project}/{username}/{namespace}/{redirects}/{deleted}`

Get the number of pages created by the user in the given namespace.

Parameters:

- `project` (**required**) - Project domain or database name.
- `username` (**required**) - Account's username.
- `namespace` - Namespace ID or `all` for all namespaces.
- `redirects` - One of 'noredirects' (default), 'onlyredirects' or 'all' for both.
- `deleted` - One of 'live', 'deleted' or 'both' (default).

Example:

Get the number of mainspace, non-redirect pages created by Jimbo Wales on the English Wikipedia.

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

Get the number of article talk pages created by Jimbo Wales that are redirects.

https://xtools.wmflabs.org/api/user/pages_count/en.wikipedia/Jimbo_Wales/1/onlyredirects

2.2.3 Pages created

`GET /api/user/pages/{project}/{username}/{namespace}/{redirects}/{deleted}/{offset}`

Get the pages created by the user in the given namespace.

Parameters:

- `project` (**required**) - Project domain or database name.
- `username` (**required**) - Account's username.
- `namespace` - Namespace ID or `all` for all namespaces.
- `redirects` - One of 'noredirects' (default), 'onlyredirects' or 'all' for both.
- `deleted` - One of 'live', 'deleted' or 'both' (default).
- `offset` - Which page of results to show. If there is more than one page of results, `continue` is returned, with the subsequent page number as the value.

Example:

Get the mainspace, non-redirect pages created by Jimbo Wales on the English Wikipedia.

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

Get the article talk pages created by Jimbo Wales that are redirects.

https://xtools.wmflabs.org/api/user/pages/en.wikipedia/Jimbo_Wales/1/onlyredirects

2.2.4 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` - 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

2.2.5 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

2.2.6 Edit summaries

`GET /api/user/edit_summeries/{project}/{username}/{namespace}`

Get statistics about a user's usage of edit summaries.

Parameters:

- `project` (**required**) - Project domain or database name.
- `username` (**required**) - Account's username.
- `namespace` - Namespace ID or `all` for all namespaces.

Example:

Get Jimbo Wales's edit summary statistics on the English Wikipedia.

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

2.2.7 Top edits

`GET /api/user/top_edits/{project}/{username}/{namespace}/{article}`

Get the top-edited pages by a user, or get all edits made by a user to a specific page.

Parameters:

- `project` (**required**) - Project domain or database name.
- `username` (**required**) - Account's username.
- `namespace` - Namespace ID or `all` for all namespaces. Defaults to the mainspace. Leave this blank if you are also supplying a full page title as the `article`.
- `article` - Full page title if `namespace` is omitted. If `namespace` is blank, do not include the namespace in the page title.

Example:

Get the top edits made by Jimbo Wales in the mainspace.

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

Get the top edits made by Jimbo Wales in the userspace.

https://xtools.wmflabs.org/api/user/top_edits/en.wikipedia/Jimbo_Wales/2

Get the top edits made by Jimbo Wales to the page Talk:Naveen Jain.

https://xtools.wmflabs.org/api/user/top_edits/en.wikipedia/Jimbo_Wales//Talk:Naveen_Jain https://xtools.wmflabs.org/api/user/top_edits/en.wikipedia.org/Jimbo_Wales/1/Naveen_Jain

2.3 Page API

API endpoints related to a single page.

2.3.1 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 3

Installation

For contributors, see *Development* for additional, more detailed instructions specific to setting up a local development environment. The prerequisites listed below still apply.

3.1 Prerequisites

XTools requires the following to run:

- A recent version of Linux or Unix (such as MacOS). Windows servers are supported, however; you must enable the `app.load_stylesheets_from_cdn` if you want it to look nice.
- PHP 5.6 or newer. If you have done PHP development on your machine before, you might already have these popular libraries installed:
  - JSON must be enabled.
  - ctype needs to be enabled.
  - A MySQL-like database, and PDO including the driver for the database you want to use (e.g. PDO_MYSQL.
  - cURL must be enabled. On some environments you may need to enable this in your php.ini file. Look for a line like `;extension=php_curl.dll` and uncomment it by removing the leading `;`.
- Composer 1.0.0+
- Node and npm (tested with versions 6.2.1+ and 3.9.3+, respectively).

3.2 Instructions

1. Download the latest release into a web-accessible location. For contributors, you should develop off of the master branch.

2. Ensure that `var/` and all files within it (other than `var/SymfonyRequirements.php`) are writable by the web server.
3. Run `composer install`. You will be prompted to enter database details and other configuration information. See *Configuration* for documentation on each parameter.
4. Create the XTools database: `php bin/console doctrine:database:create` and run the migrations with `php bin/console doctrine:migrations:migrate`. This is actually only used for usage statistics (e.g. see xtools.wmflabs.org/meta). XTools will run without it but doing so may cause silent failures, as the requests to record usage are made with AJAX.
5. With each deployment or pull from master, you may need to dump the assets and clear the cache. Use `php bin/console assetic:dump` to generate the CSS and JS, and `php bin/console cache:clear` to clear the cache. For a production environment be sure to append `--env=prod` to these commands. You must also clear the cache whenever you make configuration changes.

In production, you may find that further server-level configuration is needed. The setup process for Wikimedia Cloud VPS (which runs on Debian Jessie) is documented on Wikitech. This may be of assistance when installing XTools on similar Linux distributions.

3.3 Single wiki

If you are running XTools against a single wiki, make sure you using the following *configuration options*:

- `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.

3.4 Wiki family

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

You will also need a database table that contains meta information about your wikis. It can live wherever you want; just set the `database_replica_*` variables accordingly in `parameters.yml`. XTools was built for one resembling the WMF database.

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 4

Configuration

As part of the installation of XTools, `composer install` or `composer update` may prompt you for configuration options.

4.1 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. If you are *developing against the WMF replicas* through an SSH tunnel, the value should probably be `127.0.0.1`.
- **database_replica_port** - Port for the server with the MediaWiki databases. If you are developing against the WMF replicas through an SSH tunnel, the value should the MySQL port that was forwarded.
- **database_replica_name** - Database name of any one of the MediaWiki databases (usually the default, or the 'meta'; it doesn't matter which). For installations connecting to the WMF replicas, this could for example be `metawiki_p`.
- **database_replica_user** - Username for the MediaWiki databases. If you are developing against the WMF replicas, you should use the credentials specified in the `replica.my.cnf` file, located in the home directory of your Toolforge account.
- **database_replica_password** - Password for the user for the MediaWiki databases.

The 'meta' database:

- **database_meta_name** - Database name for the server with the `wiki` table (this is not required if `app.single_wiki` is set). If connecting to the WMF replicas, the value should be `meta_p`.

For WMF installations, you should also specify credentials for the `tools-db` database server, which contains data from other tools (e.g. checkwiki):

- **database_toolsdb_host** - MySQL host name (`127.0.0.1` if connecting to the replicas via SSH tunnel).
- **database_toolsdb_port** - MySQL port number.
- **database_toolsdb_name** - Username to connect as (should be the same as `database_replica_user`).
- **database_toolsdb_password** - Password to use for the user (should be the same as `database_replica_password`).

4.2 Authentication and Email

OAuth is used to allow users to make unlimited requests, and to allow them to see *detailed statistics* of their own account. The credentials need to be requested from `Special:OAuthConsumerRegistration` on your default wiki. This requires the OAuth extension. If this extension is not installed or you are developing locally, you can leave these options blank.

- **oauth_key** - Oauth consumer key.
- **oauth_secret** - Oauth consumer secret.
- **mailer_transport** - Software for the mailer. For development installations, you can leave all the mailer configuration options blank.
- **mailer_host** - Hostname for the mailer.
- **mailer_user** - Username for the mailer software.
- **mailer_password** - Password for the mailer software.

4.3 Caching

These options are available if you wish to use a cache provider such as Redis. However, XTools will function well using only the file system for caching.

- **cache.adapter** A cache adapter supported by DoctrineCacheBundle. *file_system* is the default and should work well.
- **cache.redis_dsn** The DSN of the Redis server, if `redis` is used as the `cache.adapter`. If you're not using Redis, this parameter can be ignored.

4.4 Wiki configuration

The parameters ensures paths to your wiki(s) are properly constructed, and for communicating with the API.

- **wiki_url** - Full URL of the wiki, used only if `app.single_wiki` is set to `true`. The title of pages are attached to the end.
- **api_path** - The API path for the projects, usually `/w/api.php`.

- **default_project** - The base URL of whatever wiki you consider to be the "default". This will be the default value for the "Project" field in the forms. On the Wikimedia installation, `en.wikipedia.org` is used because it is the most popular wiki. For single-wiki installations, the "Project" field in the forms are hidden, but you still need to provide this value for `default_project`.
- **opted_in** - A list of database names of projects that will display *restricted statistics* regardless of individual users' preferences. For developers working off of the replicas, use `enwiki_p`.

4.5 Application

- **secret** - A secret key that's used to generate certain security-related tokens, and as the secret for the internal API. This can be any non-blank value. 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** - This is used to broadcast a notice at the top of XTools. Set to `1` to turn this feature on.
- **app.noticeStyle** - Style of the notice banner, correlating to the Bootstrap contextual classes. Available options include `danger`, `warning`, `info` and `success`.
- **app.noticeText** - Message shown to the user. If you provide a valid i18n message key, it will display that message instead.
- **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 Cloud Services environment. If you are developing against the WMF replicas through an SSH tunnel, set this to `true`.
- **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. For WMF installations, this parameter is obsolete and can be left blank, as the new replicas do not suffer from noticeable lag.
- **app.rate_limit_time** - Used for *rate limiting*. This parameter is the 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.
- **languageless_wikis** - This should be left blank for any non-WMF installation. This is used only to convert legacy XTools URL parameters to the modern equivalents, listing any wikis where there is no specific language

associated with it. "meta.wikimedia.org" is intentionally not included. Developers may also leave this value blank.

4.6 Tools

Selectively choose which tools to enable within XTools.

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

CHAPTER 5

Development

If you are only looking to contribute to XTools as a developer, it is recommended that you use a Toolforge account to connect to the WMF replicas so that you'll have live data that matches production. Requesting an account should be the first thing you do, since it involves an approval process. Having access to Toolforge may benefit you in working on other Wikimedia projects, too.

5.1 Overview

- XTools is based on Symfony 3, which is a full MVC framework. We use Twig as our template engine.
- All the PHP lives in `src/`.
  - There is a single bundle called `AppBundle`, which contains the controllers, event listeners, helpers and Twig extensions.
  - Models and repositories live in `src/Xtools`. Repositories are responsible for fetching data and the models handle the logic.
- Views and assets live in `app/Resources`.
  - In `app/Resources/views`, there is a separate directory for each controller. The `index.html.twig` files are the form pages (example), and `result.html.twig` pages are the result pages (example). Some tools like the Edit Counter have multiple result pages.
- Routes are configured using the `@Route` annotation.
- By convention, each tool has it's own controller that handles requests, instantiates a model, sets the repository, and returns it to the view. Not all the tools follow this convention, however. Each tool is also registered within `app/config/tools.yml`.
- XTools was built to work on any MediaWiki installation, but it's target wiki farm is Wikimedia. Some features are only available on the Wikimedia installation, which is what all the `app.is_labs` checks are for.

5.2 Running Development server

First make sure you meet the *Prerequisites*, 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 application or session data; any changes you make are visible after refreshing the page. However when you modify the `app/config/parameters.yml` file or other things in `app/config`, you may need to clear the cache with `php bin/console c:c --no-warmup`.

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

5.3 Developing against WMF replicas

If you want to use the WMF database replicas (recommended), first make sure you have an account and shell access. Then you can open up the necessary tunnels and a shell session with:

```
ssh -L 4711:enwiki.web.db.svc.eqiad.wmflabs:3306 -L 4712:tools-db:3306 your-
↪username@tools-dev.wmflabs.org
```

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-db-username-here>
database_replica_password: <your-db-password-here>
database_toolsdb_host: 127.0.0.1
database_toolsdb_port: 4712
database_toolsdb_name: tools-db
```

Change the `your-*-here` bits to your own values, which you can find in your `replica.my.cnf` file in the home directory of your account on **'Toolforge'_**.

5.4 Table mappings

The replicas have 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'
```

For the `logging` table, sometimes we use `logging_userindex` and other times `logging_logindex` (depending on what we're querying for). This is handled in the code via the `getTableName()` method in `Repository.php`.

5.5 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')`.

5.6 Style Guidelines

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

5.7 Tests

Tests are located in the `tests/` directory, and match the `src/` directory structure. They are built with PHPUnit. Repositories only handle fetching data and do not need to be tested. Controllers also interact with the database, and while tests are most welcomed for these, they will not run on the continuous integration server (Travis and Scrutinizer) due to limitations.

There are also tests for linting, phpDoc blocks, and file permissions.

Use `composer test` to run the full suite, or `./vendor/bin/phpunit tests/` to run just the unit tests.

5.8 Writing the docs

We use ReadTheDocs. 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 running `make clean && 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

5.9 Releases

Releases are made by tagging commits in the master branch. 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 `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

5.10 Additional Help

- Email: `tools.xtools` @ `tools.wmflabs.org`
- IRC: #wikimedia-xtools (Direct link - Requires an IRC client)
- MediaWiki talk page: Talk:XTools

CHAPTER 6

Administration

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

6.1 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.

6.2 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`

6.3 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 7

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

7.1 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.

7.2 How to opt out

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

CHAPTER 8

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.

8.1 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...*.

8.2 Admin Score

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

8.3 Admin Stats

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

8.4 Article Info

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

8.5 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....

8.6 Bash

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

8.7 Blame

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

8.8 Pages

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

8.9 RFX

RFX Read more about RFX....

8.10 RFX Vote

@TODO Read more about RFAVote....

8.11 Simple Counter

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

8.12 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 9

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.