

---

# Table of Contents

Introduction	1.1
Requirements	1.2
Limitations	1.2.1
Installation	1.3
Uploading the script	1.3.1
Creating a MySQL database	1.3.2
Running the installer	1.3.3
Adding the tracking code	1.3.4
Debugging	1.3.5
Basic usage	1.4
Usage tips	1.5
API	1.6
Client side API	1.6.1
Changelog	1.7
Support	1.8
Donate	1.9

# Quick Start Guide



## Quick links

- [Official userTrack website](#)
- [Installation](#)
- [Support](#)
- [Donate](#)

## Get

If you have not yet purchased *userTrack*, you can view the [prices here](#). If you have already purchased *userTrack*: **THANK YOU**, every purchase increases userTrack's [update frequency](#), [documentation](#) and [support](#).

## Install

To get started with userTrack you first have to install it. After reading the [requirements](#) you can go directly to the [installation section](#).

## Use

***userTrack*** has recorded data but you are not sure how to interpret it? Read the [usage tips](#) to see the recommended ways to take advantage of the recorded data.

# Requirements

## Web-development knowledge

If you know how to do the following tasks you should have no problems [installing userTrack](#) on your server in under 5 minutes. If you don't know how to do any of those steps, don't worry as they are explained in the installation guide or you can contact the [support](#) for help.

- Upload files to your server
- Create a database
- Include a JavaScript file in your website pages

## Server requirements

- PHP (**pdo\_mysql** extension enabled)
- MySQL
- Access to create a database and upload files to the server

## Client requirements

- HTML5 doctype
- userTrack's **tracker.js** tracking script file has to be included on all the pages you want to track.

## Other recommendations

- *Google Chrome* browser is required for accessing the userTrack dashboard.
- Large monitor resolution is preferred for improved experience (1080p)

## Limitations

- Interactions over or inside **iframes** are **not** recorded (*browsers do not propagate events from iframes to the parent document*)
- The script is not suitable for websites with very **high traffic**. Because `_userTrack` is **self-hosted** the maximum number of recordings stored depends on your server configuration, usually it is a good idea to limit the recordings to only store latest 1000-2000 user data.
- Page content, versions or updates are not stored. This happens because `_userTrack` only stores the URL of the recorded pages, this means that the page content is only loaded when the record is being played and it is always the latest/current version of the page at that URL.
- As a consequences of the previous point: if a user has to **log-in** in order to **access** a specific area of the website you will **not** be able to view recordings on that area as long *as you are not logged in as well*. Also, if you are logged in as a different user you may see their actions done on your user panel.



# 1. Installation

## 1.1 Standalone version

userTrack was designed to run on any server or shared hosting as long as it meets the [requirements](#).

The installation can be started in 4 simple steps:

1. [Upload the script](#)
2. [Create a new MySQL database for userTrack](#)
3. [Run the auto installer](#)
4. [Add the tracking code to your site](#)

## 1.2 WordPress version

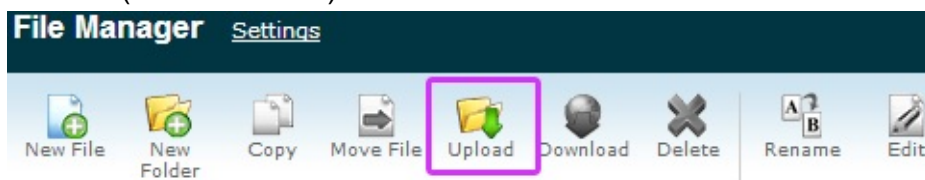
1. Install the **wp-usertrack.zip** archive as a WordPress plugin.
2. Click **Run auto setup** in the userTrack dashboard to auto-detect settings.
3. Give access to view data for the current **domain** name to the **admin** user.
4. Make sure to check the "Pages to track" menu to set on which pages to automatically insert the **tracker.js** file.

# Installaion

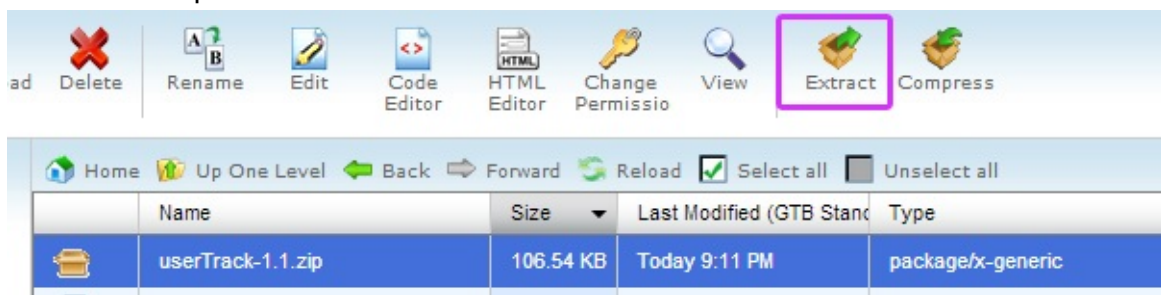
1. **Upload the script**
2. [Create a new MySQL database for userTrack](#)
3. [Run the auto installer](#)
4. [Add the tracking code to your site](#)

## 1.1 Uploading the script

1. **Upload**, using cPanel or FTP, the **userTrack.zip** archive to the root folder of your website (recommended).



2. **Extract** the zip file and delete it afterwards.



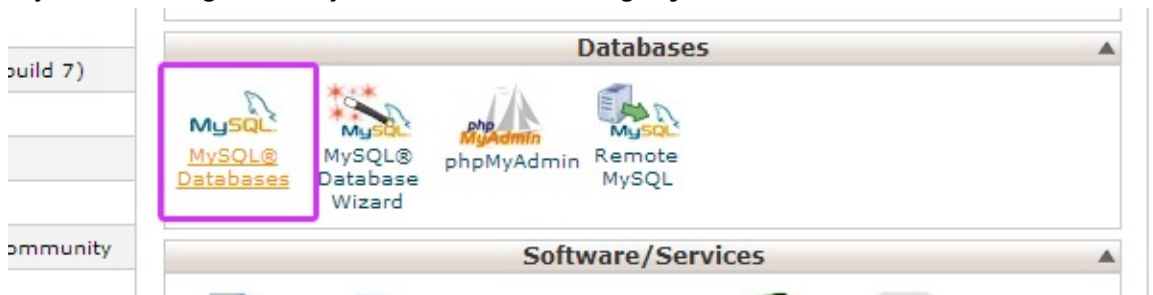
# Installation

1. [Upload the script](#)
2. **Create a new *MySQL* database for userTrack**
3. [Run the auto installer](#)
4. [Add the tracking code to your site](#)

## 1.2 Creating the database

First step is the auto-installer will be to **create a MySQL database**. To be a step ahead, you can already create a database for userTrack.

If you are using cPanel you can create it using **MySQL® Databases**.



Make sure you remember **database name**, **username** and **password**.

Note that you have to create a database, create an user and add that user to the database. Mark the "**ALL PRIVILEGES**" checkbox when asked.

# Installaion

1. [Upload the script](#)
2. [Create a new \*MySQL database\* for userTrack](#)
3. **Run the auto installer**
4. [Add the tracking code to your site](#)

## 1.3 Auto installer

Access **[www.yourdomain.com/userTrack/install.php](http://www.yourdomain.com/userTrack/install.php)** and follow the instructions.



# Installaion

1. [Upload the script](#)
2. [Create a new MySQL database for userTrack](#)
3. [Run the auto installer](#)
4. **Add the tracking code to your site**

## 1.4 Tracking a domain

1. To track a new domain you have to add it first, from the userTrack dashbaord, using the **add domain** button that appears next to the user that should have access to view data for this domain.
2. On each page that you want to track you have include this script (actual link provided in your dashboard):

```
<script src="//www.yourdomain.com/userTrack/tracker.js" defer></script>
```

Add it just before the end of the **head** tag, like so:

```
<head>
...
...
<script src="//www.yourdomain.com/userTrack/tracker.js" defer></script>
</head>
```

If you want different tracking settings for each domain tracked you can change the settings from **tracker.js**, saving it with another name (eg: tracker-domain1.js) and include this copy on the other domain instead of the original tracker.js.

## 1.4 Debugging (Troubleshooting)

1. If the script does not record anything or simply doesn't work, you can:
  - i. Open Google Chrome's **dev panel** by pressing CTRL + SHIFT + J
  - ii. Select the **Console** tab.
  - iii. See if there any **errors** in the console.
2. If the script is still not working :( contact me at **support [at] tips4design**.



(under construction)

## 2. Usage and features

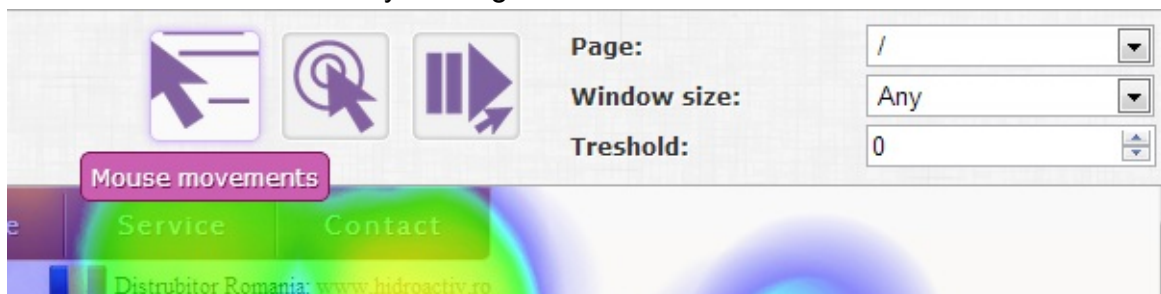
Note: This guide assumes that **userTrack** was installed in the `public_html` root of your site, in a folder called **userTrack**.

### 2.1 Accessing the interface

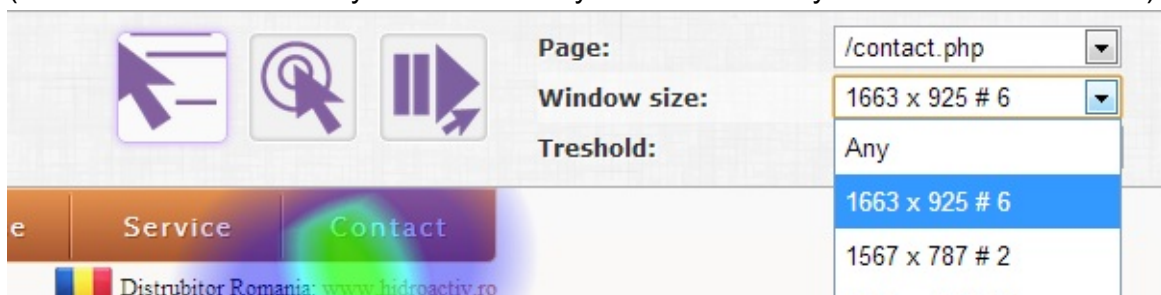
1. You first have to login. To login go to **yourdomain.com/userTrack** and enter your password. . By default the username is **admin** and password is **letmein**
2. Make sure to change the default password after first logging in.

### 2.2 Mouse Movement heatmaps

1. You can enter this section by clicking the first button.



2. You may now choose which page to show data from. You can also choose a resolution (this is recommended as your website may render differently on different resolutions)



3. Higher activity regions on the page are shown using the red color.

## 2.3 Mouse Clicks heatmaps

1. You can enter this section by clicking the second button.
2. Same as 2.2 but this heatmap displays areas with most number of clicks.

## 2.4 Scroll heatmaps

1. You can enter this section by clicking the third button.
2. The scroll heatmap is currently based on mouse movements. Higher activity regions are shown with the red color.

## 2.5 Record Playback

1. You can enter this section by clicking the fourth button.
2. Choose a session from the list and click **PLAY RECORD** to start the playback.
3. Once the user visits another page the record corresponding to that page will be automatically selected and played.
4. To delete a record double click the X icon. You can stop playing a record by clicking "STOP"
5. Important note: Due to performance issues the last page a user visits is not fully recorded. When a user enters a page the recording of the last page is sent. This saves a lot of db usage by reducing the number of requests sent to the server.

### 2.5.1 Autoplay

1. You can start autoplaying a list of visits.
2. In the clients list select one record by clicking it and click "START AUTOPLAY"
3. All records present in that list **including and above** the selected record will be played on after another.

## 2.6 Settings

4. You can access this menu by going to "Change tracker settings" on the main dashboard.
5. Every change you make here is saved directly into **tracker.js** file.
6. Every time you open the menu the settings value you see are loaded directly from the **tracker.js** file.

7. Place mouse over a setting name or value to see a small description.

## 2.7 Admin settings

1. You add users by toggling the admin panel and clicking the "Add user" button. You can then change his name or level simply by clicking inside the table
2. To change the limit for recordings simply click the "Record limit: 100" area to edit it.

## 2.8 Remove data from database

1. To delete all stored recordings go to the records tab and in the dialog opened in the lower right corner you have options to delete either stored session, or all data stored in the database including heatmaps
2. You can also delete specific clients (users) with all their corresponding data by selecting the users from the list and clicking the trash icon

(under construction)

# Usage tips

## Heatmaps

Here are a few tips on how to make use of the heatmap data generated by userTrack.

### 1. Unused call-to-action elements

If you have buttons, links or other interactive elements that the user is supposed to click on but using the heatmap data you noticed **low to no activity** on them you should consider:

1. If it's not a core element of your site you can **entirely remove the element** as it has no real-world use to the user and it occupies space and is a distraction for the user.
2. If the element is actually important for your site and you want the users to click it try making the element more visible by:
  - i. Ensuring there is a high contrast between the element and the background.
  - ii. The element has enough white space around it and is not very close to other important call-to-action elements.
  - iii. Make sure the element's text accurately represents what will happen if the user clicks it.

## Recordings

### 1. Gray "play" button and fake visitors

If there's a gray `play` button next to a user it usually means:

1. The visitor was actually a **bot** or fake visitors. You see this gray button a lot if, for example, you advertise on Facebook which currently sends a lot of fake visitors or bots.
2. The visitor left the page immediately, before any data could be sent to the server. This could also happen if the user opens your site in a background tab and closes that tab without ever viewing your site.

### 2. Panic clicking

If a user clicks multiple times in the same spot and nothing happens it might be because he would expect for something to change when he clicks that spot. Don't make elements look like buttons if they are not one. This could also mean that the button is not working properly or it's buggy.

## Other stats

- I find very important the list of referrers presented in the main dashboard. Notice which websites bring the most traffic to your page and try *milking* them as much as possible.



## API

userTrack has both a [client API](#) and a service API.

### Client API

The [client API](#) offers some JavaScript functions that you can use while the client is recorded to add additional data to the recording or affect the way the recording is done.

### Service API

The service API describes the REST API used by the userTrack dashboard to view, filter and modify the stored data. This API is **not yet fully documented**, but you can have a look through the *userTrackAPI.js* file to see all API methods implemented.





## Client side API

### 1. Disable tracking yourself

- i. `UST.disableRecord()`
- ii. `UST.enableRecord()`

### 2. Dynamically tag clients

- i. `UST.addTag('tag_name')`

### 3. Force send the recorded data

- i. `UST.forceSendData()`

## 1. Disable tracking yourself

### Example

```
// Starting with the next page load the tracking will be disabled
UST.disableRecord();

// Removes the "disabled" flag.
// Starting with the next page load you will be tracked
UST.enableRecord();
```

### Instructions

- To disable tracking yourself or other client you can call **UST.disableRecord()** from the JavaScript console or from a JS file.
- To re-enable tracking the current client call **UST.enableRecord()**
- The tracking enabled/disabled flag is stored in the **localStorage**. As long as the browser localStorage is not cleared/reset this setting will persist. (eg: you won't be tracked again unless you clear all your browser data/cookies/localStorage)

## 2. Tag visitors dynamically

### Example

```
// Call this when the scrollbar reaches the end
UST.addTag('scrolled_to_footer')

// Call this when user adds an item to the basket
UST.addTag('add_to_basket_timestamp: ' + Date.now())
```

## Instructions

- Sometimes you might want to save additional data for each tracked user (such as username or whether they clicked a button or not).
- To add a tag the function **UST.addTag()** is provided. It has only one parameter which is the tag value. For example after calling this: **UST.addTag("username\_John")** The tag will be saved for the current recording and you will be able to find it more easily.
- You can also enable auto-tagging when certain elements are clicked. Add the HTML5 attribute **data-UST\_click\_tag** to set what tag will be added if the current element is clicked. Example: `<input type="text" data-UST_click_tag="clicked_input" />` . When this input is clicked the tag **clicked\_input** will be added for this session. You will then be able to, for example, see only the recordings of users who clicked that specific input.

## Notes

1. Once written the value of **tags can not be changed**.
2. Each tag is **unique** and can appear **only once**. If you call `UST.addTag()` again with the same tag name, nothing will happen.
3. If you want more tags with similar name you can either add a random value or an unique id to each tag name.

## 3. Force send the recorded data

### Example

```
// A request to send *new* data to the server will be created
UST.forceSendData();
```

## Instructions

- userTrack sends data to the server in an efficient way: it **batches** multiple actions made by the users and sends all of them at once in a single request. This means that sometimes you have recorded data stored on the client that hasn't been sent yet (it's queued to be sent in a specific amount of time). You can call **UST.forceSendData()** to reset the **time left** to send the next batch to **0**, thus forcing userTrack to send the data as soon as possible.
- One **use case** of this method is if you want to make sure the last seconds of the user's

visit are recorded by calling **UST.forceSendData()** inside window `onbeforeunload` callback.

- You could also call this function after the user makes an important action that you want to make sure has been sent to the server (eg: `user clicks a specific button` ). Note that in this case you should also use `UST.addTag()`

# Changelog

---

*Note that the version updates might be posted here before the new version has actually been released.*

## userTrack 2.3.1 (11 July 2017)

- **NEW! Text selection** is now recorded. If the user selects some text or element during his visit, userTrack will now also record and playback those actions.
- **NEW! Local timezone.** The date and time of the recordings will now always be shown in the current timezone (the timezone of the device used to view the recording).
- Small UI improvements.
- **Bug fixes:**
  - Fixed the previous bug mentioned in the 2.3.0 notes regarding cross-domain tracking.
  - Fixed a bug where sometimes click positions would not be stored as integers.
  - Fixed a bug where the unique path to an element was not correctly generated. Also compressed the unique path string to reduce memory usage.
  - Fixed a bug where **position: fixed**; elements would not be fixed during playback (due to a bug with Chrome iframes).

## userTrack 2.3.0 (1 July 2017)

---

**NOTE!!:** There is a bug in this version (2.3.0) which stops the script from working cross-domain. To fix the bug please remove **line 251** from **tracker.js**, more exactly this has to be removed: `xhr.setRequestHeader('X-Requested-With', 'XMLHttpRequest');`

---

- **NEW! jQuery** dependency has been **removed**. Now, tracker.js no longer requires jQuery to be included.
  - **NEW! Autoplay mode.** You can now start playing a list of recordings from the clients list. Once one user session was played the next one will automatically start.
  - [API] **NEW!** [UST.forceSendData\(\)](#) allows you to immediately queue the sending of all stored data that has not been yet sent to the server.
-

- **Performance** improvements:
  - As jQuery is no longer used most tracking should be faster as it uses pure JavaScript
  - If you only included jQuery of userTrack this is a huge performance boost.
  - Heatmap should only be drawn once (previously sometimes it might be drawn twice after the page was loaded).
  - A *passive* scroll listener is used to track scroll.
- Security improvements.
- **Bug fixes:**
  - Fixed a bug where text input was sometimes not recorded.
  - Fixed thumbnails not displaying in the main dashboard.

## userTrack 2.2.0 (12 June 2017)

- NEW! Recordings that haven't been watched show a "new" icon.
- **Performance** improvements:
  - A new format to store data has been implemented (replacing JSON) which lead to **~70% less storage used** for recordings and heatmap data.
  - AJAX requests for storing data have been replaced by tracking pixel requests (reducing network usage)
- **Bug fixes:**
  - Viewport size is more accurately recorded.
  - Dropdown option select should be correctly recorded.
  - Other small bug fixes related action recording.
- **Development:** E2E tests have been created for us to use in order to make sure userTrack changes and new features don't break existing functionalities. This doesn't affect you in any way, just means that new userTrack releases will be more frequent and less likely to introduce bugs.

## userTrack 2.1.0 (22 May 2017)

- NEW! Interface update.
- NEW! The ability to scale the size of the heatmap points.
- NEW! Ability to scale the entire iframe/heatmap. This means you can easily view heatmaps for users who had larger screen resolution than yourself.
- NEW! Changing the page by clicking inside the iframe will also update the heatmap data. This means you can now change the page for which to view the heatmaps for by simply navigating through your website.
- Started refactoring most of the codebase.

- Many other UI/UX improvements.
- Dashboard performance improvements.
- Fixed a bug where sharing a recording did not work if the tracked site was from a different domain.

## **userTrack 2.0.3 (24 March 2017)**

- The API to get visitors IP has been replaced with a local implementation.
- Improved IP2Location API usage, performance and accuracy.
- City of visitor is also displayed while hovering over the country flag.
- Created the foundation for a new and improved permissions system.
- Security improvements.

## **userTrack 2.0.2 (20 January 2017)**

- NEW! Share recording. You can now share a recording via a public link.
- NEW! IPv6 support for country flag detection.
- Performance improvements
- UI bug fixes
- Starting with this version the database structure will auto-update if necessary. This means that you will most likely be able to keep your old data while upgrading to a new version.

## **userTrack 2.0.1 (11 December 2016)**

- Several UX improvements.
- WordPress version bug fixes (click position is displayed correctly regardless the visibility of the WP admin bar)
- WordPress version UX improvements.

## **userTrack 2.0.0 (5 November 2016)**

- NEW! Basic mobile tracking support.
- NEW! Window resize event is also tracked.
- Disable MySQL 5.7 FULL GROUP BY error bug.

## **Older major updates**

- **userTrack 1.9** (20 June 2016)
- **userTrack 1.8** (09 February 2016)
- **userTrack 1.7** (4 April 2015)
- **userTrack 1.6** (8 September 2014)
- **userTrack 1.5** (11 February 2014)
- **Version 1.4** (16 July 2013)
- **Version 1.3** (01 April 2013)
- **Version 1.2** (18 February 2013)
- **Version 1.0** (01 February 2013)

# Support

If you have any trouble installing this script or have bugs to report feel free to send an e-mail at **support [at] tips4design.com**.

*PS: I am currently working on creating a support platform with CodeCanyon integration, stay tuned for the awesomeness to come :)*



# Any help is appreciated

Although userTrack is not free, being a product (with a one-time fee and no monthly charge) that is **very** affordable, the earnings are nowhere near enough to even cover the development costs. If you like userTrack and want to support the further development of the script you can:

## 1. Spread the word

The best and easiest way to **help userTrack grow** is to tell your friends about it. Even better, you can **earn money** while doing so using [Envato's referral program](#).

## 2. Extend your support period

You can [extend](#) the initial 6 months [support period](#) included from your [CodeCanyon account](#). By extending the support period before it expires you save around 30%.

## 3. Support the patreon page

With [Patreon](#) you can help by donating a monthly amount while obtaining different benefits such as patron-only posts, the ability to request features to be implemented and even access to a new free API platform that is currently under construction.

Visit [my patreon page](#) (currently under construction).

## 4. Donate via PayPal

All donations are appreciated and every donation received will be invested into improving userTrack.