First, a few key points:

1. This is a bookmarklet, not an extension, so as far as I can read the new rules, this is perfectly acceptable. A bookmarklet is a fancy bookmark that runs a script, rather than loads a page. You will need to keep these handy, so enable your bookmark toolbar so you can easily click on them.
2. You have to click on the bookmarklet every time a task loads for it to work – slightly more extra work than RaterAide or LBTimer but, in the end, hopefully still as useful.
3. This logs the AET for each task – it won’t act as a session timer in the RA sense, but it will give you a count of all of the tasks you’ve completed and the overall AET time, so no pen and paper. And in the end, as we’re meant to work and submit to AET, you shouldn’t have any problems.
4. This bookmarklet doesn’t load a page, it loads a script (or rather several). There is only one version, with an autosubmit timer. I decided the upkeep of one without was too much.
5. If you need to send task information to a spreadsheet more than once, you can click the bookmarklet again. It will ask you if you want to resubmit. This might be useful if you accidently delete your task information from your spreadsheet, or you have an internet outage and your original task information didn’t send.
6. This will require a little bit of setup, but once its setup, you shouldn’t have to do any maintenance. A sample spreadsheet is provided that tries to replicate as much of the banned extensions as possible – with the benefit of working currency conversions! – but if you feel there is anything missing or you’d want to change, feel free. Do what you like with it – just make sure you leave the ‘Master Task Sheet’ alone, as the information the bookmarklet scrapes goes straight here. That includes the headers – if you’re brave and know Javascript, you can edit them in your bookmarklets. You can use arrayformula to scrape the information from there.
7. I’ve also created, in the sample spreadsheet, a month overview task page. By editing the numbers where it say ‘Month:’ (Cells P1 and Q 1) (At the moment of writing this, it will say 4 and 2018) to say, 6 2018 for June 2018, you will get a task overview for that month.
8. This has been developed with some considerable time, effort and in one or two places where my own skills were lacking and I had to outsource, cost so if you’d like to donate, feel free to get in touch. However, this script and the spreadsheet are provided **totally free of charge** and without support, although for the most part, I’ll be happy to answer questions where I can. I’ll also make the odd occasional update or addition, depending on time and any changes.
9. These scripts are provided as is and are to be used with the understanding they are provided without any kind of warranty or support of any kind.

Lets get started then!

The first thing you need to do is [grab a copy of my spreadsheet here.](https://docs.google.com/spreadsheets/d/18moFdVFEp_-qyhdYnfBINwJOr9C11tMEj7JeI099ibg/edit?usp=sharing) You will need to make a copy to your own Google Drive.

Once you’ve got a copy, open it. On your newly copied spreadsheet**, go to Tools,** and then **Script Editor.** Go to **Run, Run Function,** then **Setup.**

It will take a few seconds, but then it’ll ask for Authorization. **Click Review Permissions** and then sign in again with your Google account if it asks you to do so. You’ll get an error saying **‘This app isn’t verified.’** – ignore this, **click Advanced** and then **‘Go to Raterhub scraping script (unsafe)’** (don’t worry, it isn’t unsafe).

It’ll ask for permission to view and manage your spreadsheets in Google Drive – **click allow.**

Then go to **File, Save All.**

After this, go to **Publish** and then **Deploy as a Web App.** You will need to make sure that you have the settings as follows:

**Execute the app as: Me (**your email will be listed**).**

**Who has access to the app: Anyone, even anonymous.** The default setting is ‘Only Myself’ so you want to change this, otherwise your script won’t work.

Don’t be concerned by all of this – this just lets your script treat your spreadsheet as a glorified form to submit data to. Once you’ve completed this, **click Deploy.**

**It will give you a URL for your ‘web app’. This is important, as you need to edit the bookmarklet to add this URL, so save it.**

**Open up** **the text file**. This bookmarklet should work on all tasks, unless Google decide to change things later along and I have to release an emergency update.

You need to edit the contents of the file first with your Google Sheets Web App URL first, however. It will have a placeholder in place but it must follow this format. It is at the top of the file:

**var url = 'https://script.google.com/macros/s/YourIdentifierHere/exec?';**

Take note of the single quote marks, and the question mark at the end of ‘exec’. It’s important you don’t miss these. Don’t forget the semi-colon at the end, either.

Once you’ve done, you need to actually create it as a bookmarklet. **Go to** **https://mrcoles.com/bookmarklet/ -**  this is a bookmarklet creator, which compresses the file into something you can drag and drop into your bookmark toolbar.

**Delete the placeholder on the page** which says (alert(“test!”) and copy in the contents of the file you’ve edited. **Give it a proper name – name it whatever you like. I’ve called mine ‘Scrape it Now!’**

You will also **need to click ‘Include custom script’** and add https://cdn.jsdelivr.net/npm/easytimer@1.1.1/src/easytimer.min.js - this is the EasyTimer library that helps create the timers. **If you fail to do this, your bookmarklet won’t work.**

Once that’s done, **click** **convert to bookmarklet!** You can then **drag the little blue box that appears below to your bookmark bar** that has the name you’ve given the bookmarklet and voila!

Once you’re done, give it a whirl. Load a task and click to scrape. It should appear in your spreadsheet automatically.

☺ Happy rating.