Privacy Browser Android - Feature #328

Investigate the possibility of storing the processed blocklists in SQLite databases to speed up app loading.

09/02/2018 04:17 AM - privacy guy

Status:	New	**Start date:**	09/02/2018
Priority:	3.x	**Due date:**	
Assignee:	Soren Stoutner	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour

Description

Hello, thank you for your great program. Unfortunately on my moto g2 it takes five seconds to start the gui.
Would it be possible for a better start time experience to load the gui in advance, that you can type the requested url, while Privacy Browser still loads other necessary content (such as blocklists). Something like this:

click on icon -> load gui -> type in url (during this time browser loads other essential stuff) -> render website

Thank you very much :)

History

#1 - 09/02/2018 02:30 PM - Rayleigh Rayleigh

I also experience long loading times. Just saying so it can be confirmed as not a one-off issue.

#2 - 09/03/2018 09:29 AM - Soren Stoutner

- Subject changed from decrease start time to Investigate the possibility of storing the processed blocklists in SQLite databases to speed up app loading.

- Assignee set to Soren Stoutner

- Priority changed from 2 to 3.x

As privacy guy already knows, but for the benefit of anyone else who may follow this issue, the long start time is caused by the loading of the blocklists as described at https://www.stoutner.com/privacy-browser/blocklists/ and https://www.stoutner.com/privacy-browser-2-8/. These blocklists are quite significant and require several seconds of CPU processing time depending on one's hardware.

Creating a semi-functional GUI while the blocklists are loading that doesn't introduce other bugs is a surprisingly difficult task because of the way Android's lifecycle and GUI thread are designed. I am already planning on creating a "Loading Blocklists" message as described in https://redmine.stoutner.com/issues/285 but this will also require a surprisingly large amount of code refactoring.

However, to get at the core of the problem, I have also been considering ways to speed up the processing of the blocklists so the delay is not significant. One possibility is to convert the blocklists from their native format to a SQLite database that more closely mirrors the data structures used by the ArrayLists. I am not sure this will speed things up, but it is likely enough that I plant to spend some time investigating the possibility. This is likely to be a time-consuming process, and as such, I am planning to delay it until after the addition of tabbed browsing, which is a much desired feature.

#3 - 10/13/2018 10:29 AM - Rayleigh Rayleigh

Seen as the issue is scheduled for the 3.x series, what about adding a (configurably dark/light) splash screen? It's a simple temporary workaround and it would be much more pleasant to see eg. the PB logo instead of a bright white screen for a few seconds.

#4 - 10/13/2018 09:29 PM - Soren Stoutner

If that were easy to do I would do it. However, because of the way Android is designed and the blocklists are currently implemented, Android has to finish processing the blocklists before any GUI elements are drawn, including any configuration for the background color. It is possible to redesign the blocklists so that they are loaded in the background as described in https://redmine.stoutner.com/issues/285, but it is a significant task that will likely break a lot of other things, and as such I am putting it off until after tabbed browsing in the 3.x series.

I'm really sorry for the inconvenience this causes for users on older devices. If I could think of an easy way to work around it I would.

#5 - 04/27/2019 12:11 PM - Soren Stoutner

- Priority changed from 3.x to Next Release

#6 - 05/23/2019 01:35 PM - Soren Stoutner

- Priority changed from Next Release to 3.x

SQLite can't handle string arrays natively, which makes me want to rethink the direction I should go.

https://stackoverflow.com/questions/9053685/android-sqlite-saving-string-array

The current internal structure of the blocklists uses string arrays. I could be possible to redesign them some other way, but that might end up slowing things down instead of speeding them up.

I could also look at compressing the string arrays into a single string or into a byte array, but that might also increase the total processing time instead of decreasing it.

I think I am going to sit on this a while longer before I make a decision on how to proceed.