



Speedster for Magento 1

Notes for installing Fooman Speedster:

- **Install Fooman Speedster first on your test site**, before installing it on your live site. In the event that any errors do occur which need to be resolved during the installation process, it is far preferable to complete our troubleshooting instructions on your test site so your live site will remain unaffected
- Ensure your site meets all system requirements, and always test for existing Javascript errors on your site before installing Fooman Speedster.
- Make sure that `/lib/minify/m.php` is executable (permissions like 755 on the file itself and the containing folder should work) and `/var/ minifycache` is writeable
- Speedster has its own merging mechanism, please disable Magento's default Merging feature under **System > Configuration > Developer > Javascript and CSS**
- Minifying Javascript and CSS takes a while to compute. This only needs to be done once per Javascript/CSS combination and is then written to a cache. To prime the cache for a faster customer experience, browse each page type (homepage, product page, etc) in your store once
- Important: Don't disable the output of **Fooman_Speedster under Configuration > Advanced** - it will make your site unusable. To uninstall, either use Magento Connect or edit `/app/etc/modules/Fooman_Speedster.xml` (change true to false)
- Several developers have also successfully installed Fooman Speedster on different platforms by adapting the installation process. For further reading see the [ISAPI2](#) and [Nginx](#) threads

General installation guidance:

- The instructions below are for basic setups (one server which provides Apache, Mysql and PHP). For more advanced setups, the steps are still applicable but need to be adjusted to your particular setup. Your system administrator or web host can provide you with details
- [We don't recommend installing extensions via Magento Connect](#) - please follow the instructions below

Summary

[Step 1 - Turn off Magento's Compilation Mode](#)

[Step 2 - Turn on Magento's Configuration Cache](#)

[Step 3 - Install the Extension](#)

[Step 4 - Refresh Magento's Cache](#)

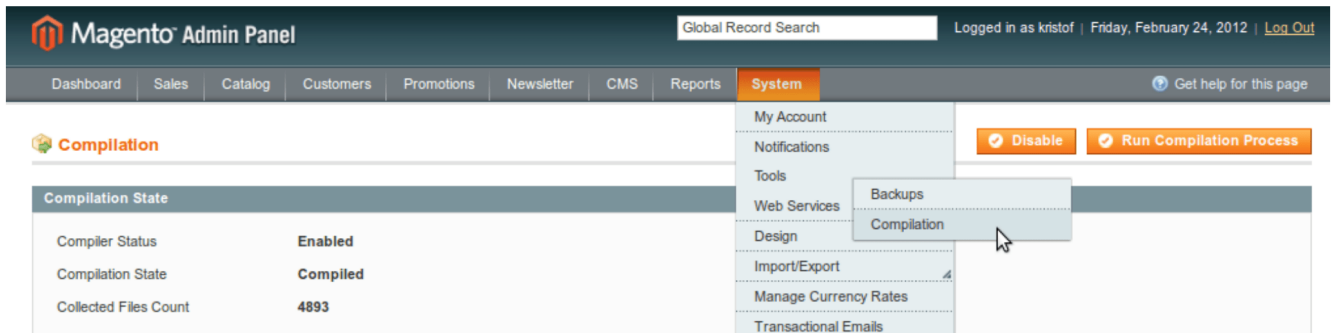
[Step 5 - Log Out of the Backend and Log Back In](#)

[Step 6 - Rerun Magento's Compilation Process](#)

If you're running a Magento multi store set up, please also follow [these instructions](#). In addition to following these instructions, also add a symlink to the lib folder.

Step 1 - Turn Off Magento's Compilation Mode

Check to see if Magento's compilation mode has been enabled. If it has, turn it off before installing an extension or making any code changes (go to **System > Tools > Compilation**). In a worst case scenario, not disabling compilation mode before installing an extension could make your site inaccessible (including the admin area). Yikes!



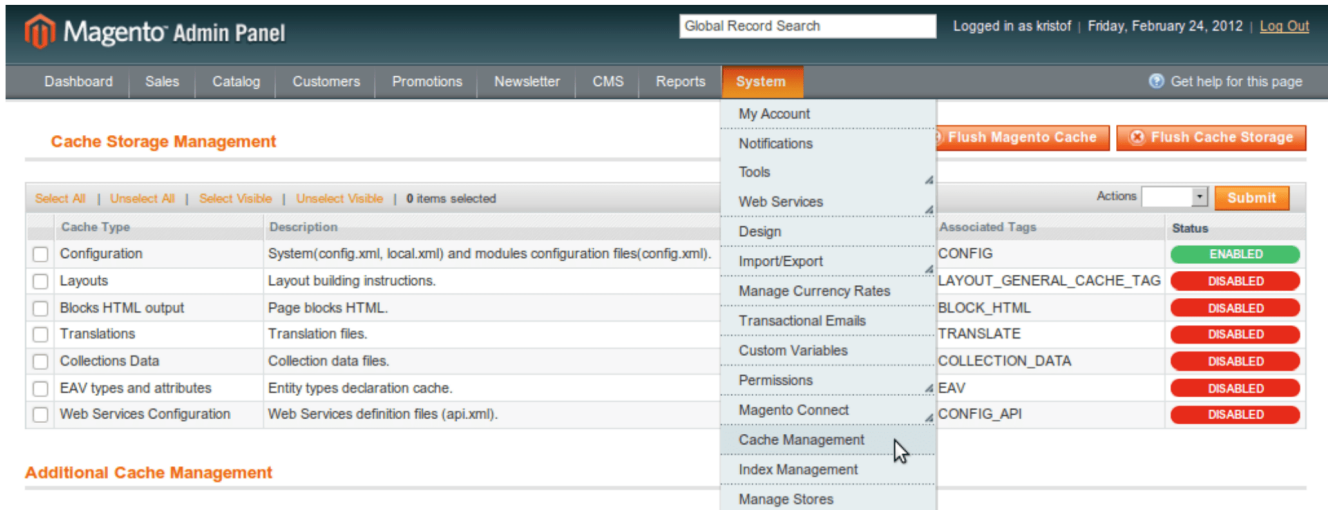
Command Line Equivalent:
php -f shell/compiler.php -- clear

As part of this step we also recommend:

- **Backing up your Magento files** so you can revert back to a previous version if something unexpected happens during installation. If this is your first backup, take additional care to verify that the backup is both complete and functional by performing a trial run of your recovery process
- **Refreshing the cache** to ensure your site is running up-to-date code, refresh your Magento cache (see Step 4). Verify that your current site works as expected ([this is a good list of what to look for](#)). If you identify any issues with your existing site, it's much easier to resolve these before adding a new extension
- [Taking your store offline](#) before taking a database backup (below). If we kept the store online and needed to rollback to this backup, we would lose any orders placed after the backup was taken. The alternative to this approach would be to keep the store live, but to re-add those orders manually if you needed to revert back to the old database. To take your site offline to customers, but retain access to the backend yourself, you can redirect users (excluding yourself) to a static page saying that the site is currently unavailable but will be back online shortly
- [Creating a database backup](#)

Step 2 - Turn On Magento's Configuration Cache

Turn on Magento's configuration cache – this will prevent Magento from trying to install the new extension too early. If the extension needs to make any database changes, triggering the installation too early could miss some database changes and create errors which are difficult to fix later on.



Cache Storage Management

Select All | Unselect All | Select Visible | Unselect Visible | 0 items selected

Cache Type	Description
☐ Configuration	System(config.xml, local.xml) and modules configuration files(config.xml).
☐ Layouts	Layout building instructions.
☐ Blocks HTML output	Page blocks HTML.
☐ Translations	Translation files.
☐ Collections Data	Collection data files.
☐ EAV types and attributes	Entity types declaration cache.
☐ Web Services Configuration	Web Services definition files (api.xml).

Additional Cache Management

Flush Magento Cache | Flush Cache Storage

Actions | Submit

Associated Tags | Status

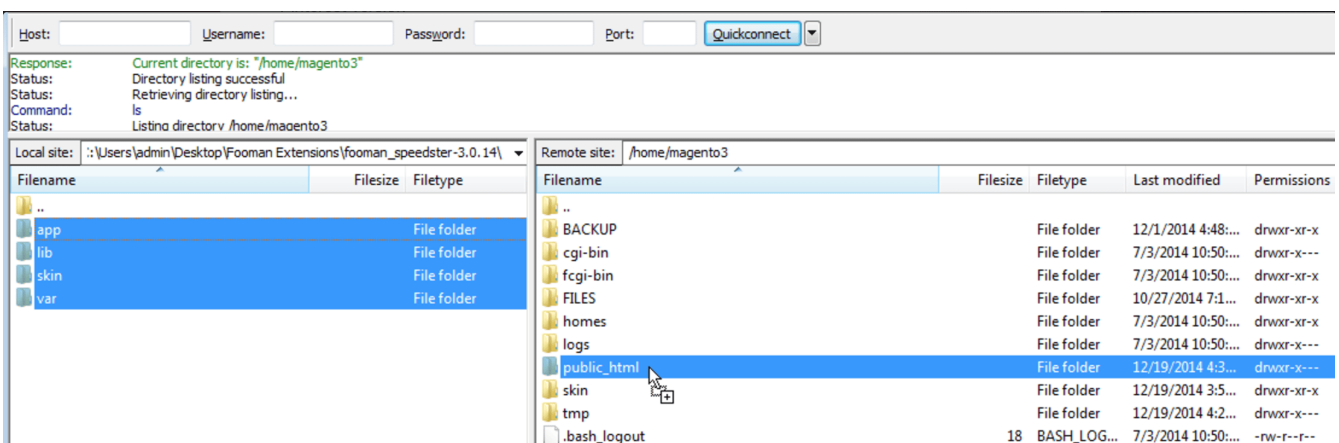
Associated Tags	Status
CONFIG	ENABLED
LAYOUT_GENERAL_CACHE_TAG	DISABLED
BLOCK_HTML	DISABLED
TRANSLATE	DISABLED
COLLECTION_DATA	DISABLED
EAV	DISABLED
CONFIG_API	DISABLED

Step 3 - Install the Extension

3.1 Download the extension, either from [your account on our website](#) or via the direct link sent to you in the order confirmation email. Once you have the files on your computer unzip the extension files into a separate folder.

3.2 Upload the extension by uploading the unzipped files to your Magento server via FTP or SFTP (an excellent option is to use the free opensource program [Filezilla](#)). To do this:

1. Connect to the server
2. Navigate to your Magento root folder
3. Upload the previously unzipped files to your root folder by dragging them from your local computer to your Magento root folder



Host: Username: Password: Port: Quickconnect

Response: Current directory is: "/home/magento3"
Status: Directory listing successful
Status: Retrieving directory listing...
Command: ls
Status: Listing directory /home/magento3

Local site: /Users/admin/Desktop/Fooman Extensions/fooman_speedster-3.0.14/ Remote site: /home/magento3

Filename	Filesize	Filetype	Last modified	Permissions
..				
app		File folder		
lib		File folder		
skin		File folder		
var		File folder		

Filename	Filesize	Filetype	Last modified	Permissions
..				
BACKUP		File folder	12/1/2014 4:48:...	drwxr-xr-x
cgi-bin		File folder	7/3/2014 10:50:...	drwxr-xr-x
fcgi-bin		File folder	7/3/2014 10:50:...	drwxr-xr-x
FILES		File folder	10/27/2014 7:1...	drwxr-xr-x
homes		File folder	7/3/2014 10:50:...	drwxr-xr-x
logs		File folder	7/3/2014 10:50:...	drwxr-xr-x
public_html		File folder	12/19/2014 4:3...	drwxr-xr-x
skin		File folder	12/19/2014 3:5...	drwxr-xr-x
tmp		File folder	12/19/2014 4:2...	drwxr-xr-x
.bash_logout	18	BASH_LOG...	7/3/2014 10:50:...	-rw-r--r--

Host: Username: Password: Port: Quickconnect

Status: Retrieving directory listing...
 Command: pwd
 Response: Current directory is: "/home/magento3"
 Status: Directory listing successful

Local site:	Remote site:
C:\Users\admin\Desktop\Fooman Extensions\fooman_speedster-3.0.14\	/home/magento3/public_html

Filename	Filesize	Filetype	Last modified
..			
app		File folder	12/19/2014
lib		File folder	12/19/2014
skin		File folder	12/19/2014
var		File folder	12/19/2014

Filename	Filesize	Filetype	Last modified
..			
app		File folder	12/19/2014 4:3...
downloader		File folder	11/26/2013
errors		File folder	11/26/2013
includes		File folder	11/26/2013
js		File folder	12/18/2014 11:...
lib		File folder	12/11/2014 8:4...
media		File folder	12/11/2014 3:3...
pkginfo		File folder	11/26/2013
shell		File folder	11/26/2013
skin		File folder	12/19/2014 4:3...
var		File folder	12/13/2014 12:...
.htaccess	5,673	HTACCESS...	12/11/2014 3:3...
.htaccess.sample	4,568	SAMPLE File	11/26/2013
api.php	2,834	PHP File	11/26/2013
cron.php	2,831	PHP File	11/26/2013
cron.sh	717	SH File	11/26/2013
favicon.ico	1,150	Icon	11/26/2013

Having issues locating your Magento Root Folder? Check [this FAQ](#).

3.3 Check the file permissions of the newly uploaded files: Files need to be readable by the webserver. Pay particular attention to the user and group of the files, and to the files' read/write permissions. It's not possible to give a generic recommendation for which settings to use because it depends entirely on your individual webserver setup. Compare file permissions of the newly uploaded files and folders against existing files on the server. Alternatively, ask your webhost or server administrator for the best settings to use.

Filename ^	Filetype	Filesize	Last modified	Permissions	Owner/Group
..					
app	Directory	20/02/10		drwxr-x—	apache apache
downloader	Directory	20/02/10		drwxr-x—	apache apache
errors	Directory	20/02/10		drwxr-x—	apache apache
includes	Directory	20/02/10		drwxr-x—	apache apache
js	Directory	20/02/10		drwxr-x—	apache apache
lib	Directory	20/02/10		drwxr-x—	apache apache
media	Directory	14/08/08		drwxr-x—	apache apache
pkginfo	Directory	20/02/10		drwxr-x—	apache apache
shell	Directory	20/02/10		drwxr-x—	apache apache
skin	Directory	20/02/10		drwxr-x—	apache apache
var	Directory	05/09/10		drwxr-x—	apache apache

Selected 1 directory.

The ideal scenario is to give the server the absolute minimum access rights needed to perform a given task. Granting universal read/write/execute (777) permissions can be unsafe and is generally not recommended.

Step 4 - Refresh Magento's Cache

Refreshing the Magento cache allows Magento to register the addition of the new extension and to run any necessary installation routines. Alternatively, turning off the cache will have the same effect.

The screenshot shows the Magento Admin Panel interface. At the top, there's a navigation bar with tabs for Dashboard, Sales, Catalog, Customers, Promotions, Newsletter, CMS, Reports, and System (which is currently selected). Below the navigation bar, there's a search bar and a login status indicator. The main content area is titled 'Cache Storage Management'. It features a table with columns 'Cache Type' and 'Description'. The table lists several cache types: Configuration, Layouts, Blocks HTML output, Translations, Collections Data, EAV types and attributes, and Web Services Configuration. To the right of the table, there are buttons for 'Flush Magento Cache' and 'Flush Cache Storage'. Below the table, there's a section for 'Additional Cache Management'. The 'System' menu is open, showing options like My Account, Notifications, Tools, Web Services, Design, Import/Export, Manage Currency Rates, Transactional Emails, Custom Variables, Permissions, Magento Connect, Cache Management (which is highlighted), Index Management, and Manage Stores.

Cache Type	Description
☐ Configuration	System(config.xml, local.xml) and modules configuration files(config.xml).
☐ Layouts	Layout building instructions.
☐ Blocks HTML output	Page blocks HTML.
☐ Translations	Translation files.
☐ Collections Data	Collection data files.
☐ EAV types and attributes	Entity types declaration cache.
☐ Web Services Configuration	Web Services definition files (api.xml).

Associated Tags	Status
CONFIG	ENABLED
LAYOUT_GENERAL_CACHE_TAG	DISABLED
BLOCK_HTML	DISABLED
TRANSLATE	DISABLED
COLLECTION_DATA	DISABLED
EAV	DISABLED
CONFIG_API	DISABLED

Step 5 - Log Out of the Backend and Log Back In

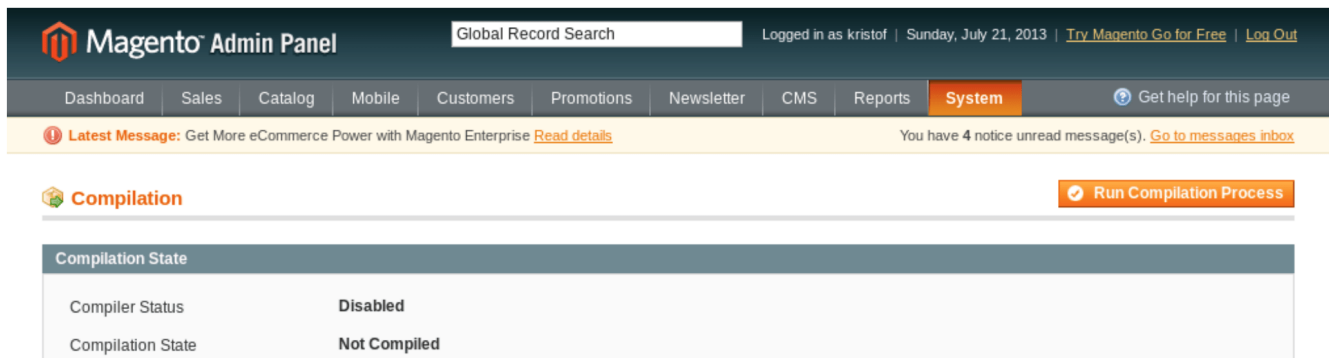
Log out and back into Magento to reload any access rules for the backend and prevent a potential “Access Denied” error on newly added admin areas that you haven’t been granted access to. Logging back in with a full administrator account will grant you the new permissions required.

Check your site to make sure your newly installed extension works as advertised, and the rest of your store still behaves as you would expect (you can use the same [checklist](#) we linked to earlier).

If you took your store offline before installing the extension, [bring it back online](#).

Step 6 - Rerun Magento’s Compilation Process

If you're using Magento’s compilation mode, you need to hit the “Run Compilation Process” button to update the compiled state with the newly installed extension (go to **System > Tools > Compilation**).



The screenshot shows the Magento Admin Panel interface. At the top, there's a header with the Magento logo, 'Admin Panel' text, a search bar, and user information. Below the header is a navigation menu with tabs like Dashboard, Sales, Catalog, Mobile, Customers, Promotions, Newsletter, CMS, Reports, and System (which is highlighted). A 'Latest Message' banner is visible below the menu. The main content area is titled 'Compilation' and features a 'Run Compilation Process' button. Below this is a table showing the 'Compilation State'.

Compilation State	
Compiler Status	Disabled
Compilation State	Not Compiled

Command Line Equivalent:
`php -f shell/compiler.php -- compile`

Installation Complete!

Troubleshooting

If you run into any issues, check out the [Installation Help](#) FAQ.

Uninstalling Fooman Speedster

If you installed Fooman Speedster via Magento Connect, then use Magento Connect to uninstall Fooman Speedster.

If you installed Fooman Speedster via FTP, you will need to delete all Fooman Speedster files/folders via FTP:

- app/etc/modules/Fooman_Speedster.xml
- app/code/community/Fooman/Speedster
- lib/minify
- skin/m/

Running the Speedster self-test

Once you have successfully enabled Fooman Speedster, you can also run the self-test from **System > Configuration > Fooman Extensions > Support**.

This version of the self-test will also include verification step 5 (this step is not included with the “enable” version of the self-test because it requires Speedster to already be enabled).

Verification steps 6 and 7 are not included in the self-test, as these are browser based tasks which are not able to be automatically tested. Please perform these tests manually.

How do I enable Speedster?

Fooman Speedster has an inbuilt self-test (which includes verification and troubleshooting steps 1-4). The self-test must run without any errors before you can enable Fooman Speedster under **System > Configuration > Fooman Extensions > Fooman Speedster**.

The self-test will highlight the most common issues that people have when configuring Fooman Speedster (it does not highlight all possible issues, because it is not possible to fully automate this process). If the self-test does not run successfully, Fooman Speedster will not be enabled. You will see an error message which tells you which verification and troubleshooting step/s failed. You must correct every error before you are able to enable Fooman Speedster.

I can't see any self-test result

Usually refreshing the cache and logging out/logging back in would fix this problem.

New Javascript/JS errors after installing Fooman Speedster

If you're seeing new Javascript (JS) errors that were not present before installing Fooman Speedster, these are most likely caused by either trying to minify already minified files, or minifying Javascript files that do not follow best practice coding style for Javascript.

To fix the double minification problem, replace the already minified file with the original non-minified Javascript. A Javascript file with .min.js in its file name is always a sure sign that this file has been minified. To find the origin of remaining Javascript errors, we suggest testing all Javascript files which are not part of the Magento base/default theme on www.jslint.com for code quality.

Javascript/JS is not loading correctly

This can be caused by a range of issues, so it's essential to perform all verification and troubleshooting steps below.

You should also test for existing Javascript errors on your site before installing Fooman Speedster (if you have already installed the extension, disable fooman Speedster to do this). Fooman Speedster relies on all Javascript on your store being completely error free. If your store contains existing Javascript errors, these will be magnified when installing Speedster and could cause issues with your store's frontend. The Speedster self-test does not include testing for existing Javascript errors on your site so it's important to do this separately.

Also check that your setup meets all system requirements for Fooman Speedster.

CSS is not loading correctly

The most common cause is incorrect permission settings (but other issues could also be the

cause). Perform the verification and troubleshooting steps listed below. Also check that your setup meets all system requirements for Fooman Speedster.

CSS styling has changed

A common issue is the use of @import directives in the original CSS files. During the minification process, the @import linked files end up being loaded first. If this is causing issues, remove the @import directives and instead load the files via an explicit layout xml instruction.

"Internal Server Error" error message

This is caused by incorrect permission settings. Change your permission settings to allow the file to be executed by the web server. Apply the same setting to the folder /lib/minify and the file m.php. Permission requirements differ from server to server, but most often permissions 755 or 775 will work. If in doubt, check which permissions are working for your main index.php in the Magento root folder.

Magento "404 Page Not Found" error message (fully styled)

This page can sometimes appear when you're running verification step 2. If redirected to Magento's fully styled, '404 Page Not Found' page, this indicates a permission issue. Please contact your Server Administrator or Hosting company to determine why the file is not accessible to you.

If you're seeing this error during the extension installation process, refer to our [Installation FAQ](#) for troubleshooting.

Failed URL rewriting verification

Perform the troubleshooting step 4.

I uninstalled Fooman Speedster, but am still experiencing CSS or JS issues

If you followed [these uninstall instructions](#) and are still experiencing CSS/JS issues (eg. CSS missing from your site), this indicates that some caching makes your system believe the extension is still in the system.

Please delete: app/etc/modules/Fooman_Speedster.xml and the contents of var/cache

If the issue persists, get in touch with your system admin person to work out what additional caching mechanism needs refreshing.

No speed improvements after installing Speedster

Fooman Speedster focuses on the frontend only and optimises CSS and Javascript.

Other speed issues that may be slowing down your site might be related to the server, the code it is running or even the theme not using Magento's default layout xml system. These are all additional things to optimise to speed up your store.

Are there any workarounds for the CSS rule limits on IE?

One possible workaround is to split the Fooman Speedster output into several chunks.

Chunking is determined by resulting url length. You could adjust this by editing `/app/code/community/Fooman/Speedster/Block/Page/Html/Head.php`

find public function `getChunkedItems($items, $prefix = "", $maxLen = 2000)` and change the 2000 to something smaller.

One thing to keep in mind while optimising performance is that using less rules is better.

Re-writing image locations

Fooman Speedster rewrites image locations based on Magento's fallback and/or your settings under **System > Configuration > Web** as appropriate.

CONNECT ERROR: Package not found

Please follow the instructions to manually install Fooman Speedster, rather than installing via Magento Connect.

Any other error message

If you encounter any other error messages not individually listed in the Support Centre, consult your server's error log for clues about the error source and refer to the appropriate troubleshooting step below.

Reporting other issues/bugs

If you experience any issues/bugs with this extension that aren't covered in this troubleshooting section:

1. Ensure you have read and followed all instructions and troubleshooting steps carefully
2. Run the self test to diagnose basic issues (see related article below)
3. Check that you can replicate the issue on Magento's default theme (with no other extensions installed), to rule out any issues caused by extension/theme conflicts
4. Clearly state at which stage of the installation process you are running into trouble (the net tab of the firebug Firefox extension is generally helpful in finding out what is not loading correctly)
5. If the issue persists, please feel free to report a bug on [Github](#)

We may contact you if we have an immediate fix or a workaround, but we can't always respond individually to every report that comes in.

Reporting Any Issues/Bugs

We are proud of our quality extension code - it's been widely tested and we stand by it 100%. If something does happen and you think you might be experiencing an issue or bug, please contact us via support@fooman.co.nz and we will help you out.