

Queue-it

- [Overview](#)
- [Wordpress Implementation - Part 1](#)
- [Wordpress Implementation - Part 2](#)

Overview

Queue-it is a virtual waiting room solution that is used to control the follow of traffic for event ticket sales.

It was implemented following issues with the Puffing Billy website and CustomLinc booking widgets struggling with capacity.

It is to be utilised for Polar Express, Day Out With Thomas x 2 and Train of Lights.

Puffing Billy's Queue-it instance can be accessed at <https://puffingbilly.go.queue-it.net/>

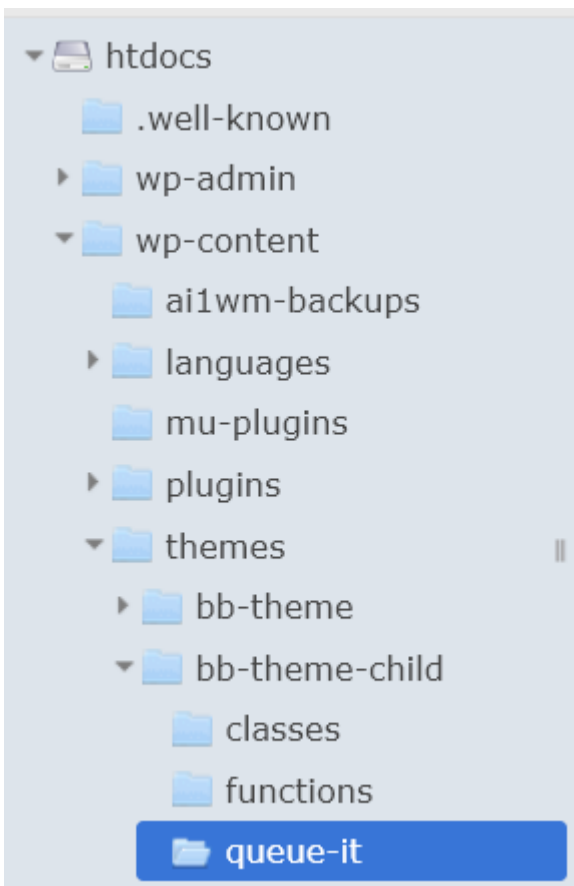
Wordpress Implementation - Part 1

Queue-it is implemented server side using PHP. The required PHP files can be downloaded via the Queue-it GitHub repository located at <https://github.com/queueit/KnownUser.V3.PHP>








The *WP File Manager* plugin is used to access the WordPress folder structure and upload the required files. It can be located in the left hand menu after logging into WordPress.



The Queue-it files should be placed in the following folder:



The required PHP files are below. The file `integrationconfig.json` is generated via the Queue-it portal and uploaded.

Name	Permissions	Modified	Size	Kind
 integrationconfig.json	read and write	Oct 30, 2023 02:30 PM	778 b	Application
 IntegrationConfigHelpers.php	read and write	Oct 18, 2023 02:31 PM	11 KB	PHP source
 KnownUser.php	read and write	Oct 18, 2023 02:29 PM	22 KB	PHP source
 Models.php	read and write	Oct 18, 2023 02:29 PM	3 KB	PHP source
 QueueITHelpers.php	read and write	Oct 18, 2023 02:29 PM	6 KB	PHP source
 UserInQueueService.php	read and write	Oct 18, 2023 02:31 PM	10 KB	PHP source
 UserInQueueStateCookieRepository.php	read and write	Oct 18, 2023 02:31 PM	8 KB	PHP source

Wordpress Implementation - Part 2

Once the required php files have been uploaded to the web server, additional php code must be added to WordPress to enable the queuing functionality. The code is accessed via PBR's private GitHub repository at <https://github.com/pbr-it/queue-it>

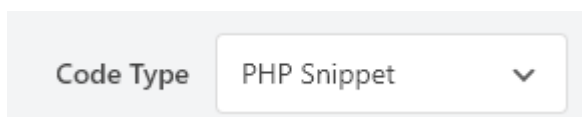
Credentials are located in the IT password store.

Before adding the code, ensure that the variable `$secretKey` is set to the secret key specified in the Queue-it portal.

The Code Snippets Wordpress plugin is used to add the code to Wordpress.

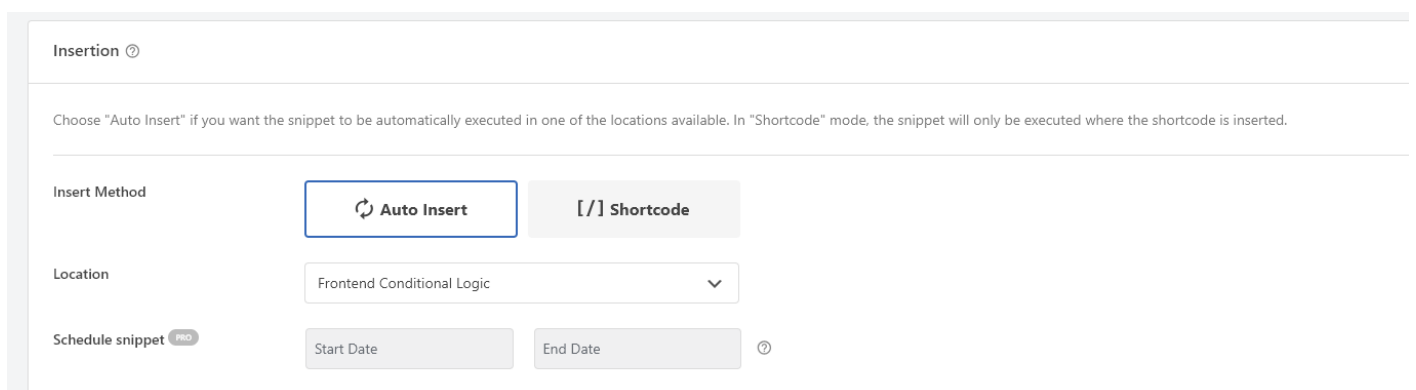
After opening the plug in, create a new snippet.

Add the code to the code preview window, ensuring that 'Code Type is set to PHP Snippet'



A screenshot of the 'Code Type' dropdown menu in the Code Snippets plugin. The dropdown is open, showing 'PHP Snippet' as the selected option with a downward arrow icon.

Ensure that the insertion options are the same as below. Selecting 'Frontend Conditional Logic' ensures that the code is only executed by visitors and not when attempting to edit the page.



A screenshot of the 'Insertion' section in the Code Snippets plugin. It shows the 'Insert Method' with 'Auto Insert' selected (indicated by a blue border) and '[/] Shortcode' as an alternative. The 'Location' dropdown is set to 'Frontend Conditional Logic'. Below, there are fields for 'Start Date' and 'End Date' under the 'Schedule snippet' section, which is marked as 'PRO'. A help icon is visible next to the 'End Date' field.

Under 'Smart Conditional Logic', specify the criteria that should apply for the code to run,

Smart Conditional Logic

Using conditional logic you can limit the pages where you want the snippet to be auto-inserted.

Enable Logic



Conditions

Show



This code snippet if

Page URL



Contains



polar-express



AND

+ Add new group

Save the code snippet.

At this point the code is in place to make the call to Queue-it at page load.

Remaining configuration is performed via the Queue-it portal.