

Waitlist Documentation

Waitlist Module Version: 1.008
02.08.2019

Requirements

Miva Merchant: **9.12.00 or higher.**

Store User Interface: **Miva Merchant CSSUI**

Installation

1. Log into your Miva Merchant Admin
2. Navigate to **Modules**
3. Click the **+** icon to add a Module
4. Click **Upload** and upload **TGWaitlist.mvc**
5. Press the **+ Add** button
6. Navigate to **System Extension Settings**
7. Click on the **Add/Remove Modules** tab
8. Look for **Waitlist** and click **Install**
9. The **Waitlist** module is now installed!

Upgrading from 1.004 and below, to 1.005 and above

Please Change the following page codes:

- **WatilistEmailTemplate** TO **WaitlistEmailTemplate**
- **WatilistEmailLogic** TO **WaitlistEmailLogic** (if you were using this)

Setting up the Waitlist Item

1. Navigate to **User Interface**
2. Click on the **Items** tab
3. Click the **+** icon to add an item
4. Set the code to **waitlist**
5. Set the module to **TGWaitlist**
6. Click the **+ Add** button
7. Go to the tab called **Pages**
8. Assign the item to the **PROD** page template

Setting up the Waitlist Scheduled Task

1. Navigate to **Store Settings**
2. Click the **Scheduled Tasks** tab
3. Click the **+** icon to add a scheduled task
4. Add a description to the new Task
 - a. Example: **Trigger Waitlist Emails**
5. Set **Trigger** as **waitlist_trigger**
6. Set **Trigger Delay** to **60** seconds
7. Select the operation as **Waitlist: Trigger Waitlist Emails**
8. Set up your schedule
 - a. Recommended: **Every: 2 Hour(s)**

Waitlist Module in the Admin

To see a list of customers, you can navigate to **System Extension Settings** and click on the **Waitlist** tab. This will show you 7 columns:

Date Added
Email
Product Code
Product Name
Variant Code
Variant Product Name
Current Stock

You can trigger emails from this view. If you select a waitlist to trigger, but that product (or variation) is not in stock, it will not send the notification out to that customer.

To view a full list of customers signed up for a specific product, you can use the **Advanced Search**.

You may only **delete** waitlists, and **trigger emails** in this view. To add customers to a waitlist, this must be done from the front-end of the website.

To view a list of waitlist emails that were already sent, you can navigate to **System Extension Settings** and click on the **Waitlist - Sent** tab. This will show you 7 columns:

Date Added
Date Sent
Email
Product Code
Product Name
Variant Code
VariantProduct Name

You may delete these entries if you would like to remove them.

Technical Details & Set-up

The waitlist form will need to be added to the **PROD** page, to render it on the product display page.

Form submission HTML

This is the actual Form. Make sure this is not placed within another form.

Required Inputs:

Action = WaitlistAdd

Waitlist_Product_Code

Waitlist_Email

Optional Inputs:

Waitlist_Variant_ID

Javascript (Form Submission & Code to show/hide the Waitlist form)

Place this near the footer of your **PROD** page. It is very important this is placed below all **attributemachine** items.

[For more technical details, click here.](#)

Waitlist Email Template

During installation, a page with the code of **WaitlistEmailTemplate** will be created. This uses a similar template to the Default Miva 9.6 email templates. This page will be rendered, and sent to customers when they receive their “Back in stock” notifications.

There are **two global variables** you can set.

g.Email_Subject - This is the Email Subject. By default (if nothing is specified), it will use this template: *[Product Name] is back in stock!*

g.Email_From - This is the email that will be used as the from email. By default (if nothing is specified), it will use the **Store Email**.

Waitlist with Miva's JSON API

To utilize the JSON API you will need to be on version **1.007** or higher of the module.

The following functions are available:

- Waitlist_Load_Query
 - Use this to load in all the waitlist entries
- Waitlist_Trigger_All
 - This will run a function to trigger an waitlist emails that need to be sent out.
- Waitlist_Load_Email
 - Load the waitlist entries for a specific email address.
- Waitlist_Load_Customer
 - Load the waitlist entries for a specific customer
- Waitlist_Add
 - Add a user to a waitlist

Miva JSON API - Waitlist_Load_Query

The following may be sorted/ filtered

| Code | Description |
|--------------|--|
| id | Waitlist ID; Unique |
| time_added | Timestamp of when a user signed up for the waitlist |
| product_id | Product ID |
| variant_id | Variant_ID |
| email | Email of the user who signed up for the waitlist |
| cust_id | Customer ID of the user who signed up for the waitlist |
| product_code | Product Code |
| variant_code | Variant Part(s) Product Code |
| product_name | Product Name |
| variant_name | Variant Part(s) Product Name |

Example Request

```
{
  "Store_Code": "YOUR_STORE_CODE",
  "Function": "Module",
  "Module_Code": "TGWaitlist",
  "Module_Function": "Waitlist_Load_Query",
  "sort": "email",
  "Filter":[
    {
      "name":"search",
      "value":[
        {
```



```
    "field": "product_code",  
    "operator": "EQ",  
    "value": "My_Product_Code"  
  }  
]  
}  
]
```

Miva JSON API - Waitlist_Trigger_All

Example Request

```
{  
  "Store_Code": "YOUR_STORE_CODE",  
  "Function": "Module",  
  "Module_Code": "TGWaitlist",  
  "Module_Function": "Waitlist_Trigger_All"  
}
```

Miva JSON API - Waitlist_Load_Email

Example Request

```
{  
  "Store_Code": "YOUR_STORE_CODE",  
  "Function": "Module",  
  "Module_Code": "TGWaitlist",  
  "Module_Function": "Waitlist_Load_Email",  
  "Email": "email@email.com"  
}
```

Miva JSON API - Waitlist_Load_Customer

Example Request

```
{  
  "Store_Code": "YOUR_STORE_CODE",  
  "Function": "Module",  
  "Module_Code": "TGWaitlist",  
  "Module_Function": "Waitlist_Load_Customer",  
  "Customer_ID": 5  
}
```

Miva JSON API - Waitlist_Add

Variant_ID and Customer_ID are optional.

Example Request

```
{  
  "Store_Code": "YOUR_STORE_CODE",  
  "Function": "Module",  
  "Module_Code": "TGWaitlist",  
  "Module_Function": "Waitlist_Add",  
  "Product_Code": "My_Product_Code",  
  "Email": "hello@email.com",  
  "Customer_ID": 2,  
  "Variant_ID": 2  
}
```

Uninstalling the Waitlist Module

To uninstall the module, you will need to make sure that the **Waitlist item** and **Waitlist Scheduled Task** are fully removed

Once these are both removed, navigate to **System Extension Settings**, and click on the **Add/Remove Modules** tab.

Find **Waitlist** and click the **Remove** button.

The module will remove the Waitlist Email Template (**WaitlistEmailTemplate**) once uninstalled, and remove all **Waitlist** data.