

Platform - Authentication

Authentication Process

Printfection Authentication of Applications

During the Alpha release Printfection must explicitly grant access to individual accounts in order for the account to request API keys and develop applications using the PF API.

The first level of authentication occurs when an API key is requested. The request can be made from the 'Applications' -> 'My API Keys' menu option when logged into your Printfection account. The developer must know what the functionality of the application will be and request the correct permissions for the application. For example, an application which uploads only needs access to image sets and possibly the ability to add image sets but not root products or stores.

Once an API key has been requested, a Printfection admin will review the description of the application and the requested permissions. The Printfection admin will then decide to either grant the permissions requested or grant a different set of permissions. If the Printfection admin needs clarification, they will contact the developer directly. Lastly, the Printfection Admin will activate the application. Once the application has been activated it can be used to access the API.

The next step is to have a user authenticate the application in order for the application to access the user's account. This next level of authentication is based on the type of application as it's different for web and desktop applications.

User Authentication of Web-based Applications

The application will send a user attempting to use the application to the login page on Printfection's website using the arguments listed below. The base URL should be `http://www.printfection.com/app/authorize.php` instead of the normal `http://api.printfection.com` URL used for most other API calls.

Once the user logs in, accepts the permissions requested by the application, and agrees to the legal agreements they will be redirected to the post-authorization URL specified by the developer. An auth token will be included in the post-authorization URL. The application needs to then retrieve the session key using the `printfection.auth.getSession` method and the auth token. The auth token will expire shortly after it is created so the session key should be retrieved soon as possible. This session key can then be used in future requests to access a user's account.

User Authentication of Desktop Applications

The application should request an auth token using the `printfection.auth.createToken` method. Immediately after requesting the auth token the application should send the user to the login page on Printfection's website. The base URL should be `http://www.printfection.com/app/authorize.php`. The application must include the auth token as a parameter in the URL to the login page.

Once the user logs in, accepts the permissions requested by the application, and agrees to the legal agreements they will be asked to close the login window and return to the application. The application should have a way for the user to let it know they have authenticated the application on the Printfection website. As soon as the application receives this confirmation it needs to retrieve the session key using the `printfection.auth.getSession` method and the auth token. The auth token will expire shortly after it is created so the session key should be retrieved as soon as possible. This session key can then be used in future requests to access a user's account.

Required Permission

None

Parameters

The permissions string can be generated on the API keys page. Login to your account and select 'Applications' -> 'My API Keys'. Then select the API key you'd like to create the permissions string for. On the right side of the page is a link to create the JSON string. Copy and paste the generated JSON string to your application for use in the permissions argument listed in the table below. See the *Permissions* document for information concerning these permissions. At a minimum the permissions string must be: `{"required":{},{}, "suggested":{}}` which requests no permissions from the user.

Name	Type	Required/ Optional	Description
api_key	string	required	This is the developers API key.
version	float	required	The version to of the API to use. (This should be 1.0)
auth_token	string	required if desktop app	This should only be included if this is called by a desktop application. Desktop apps can get this via the <code>printfection.auth.createToken</code> method.
permissions	string	required	A JSON formatted array of permissions this application is asking for. The application cannot ask for higher permissions than what Printfection has given it.
PERMISSIONS: REQUIRED: stores	string	optional	Permission required for stores.
PERMISSIONS: REQUIRED: image_sets	string	optional	Permission required for image sets.
PERMISSIONS: REQUIRED: add_store	string	optional	Permission required pertaining to adding a store to the account.
PERMISSIONS: REQUIRED: add_image_set	string	optional	Permission required pertaining to adding an image set to the account.
PERMISSIONS:	string	optional	Permission suggested for stores.

SUGGESTED: stores			
PERMISSIONS: SUGGESTED: image_sets	string	optional	Permission suggested for image sets.
PERMISSIONS: SUGGESTED: add_store	string	optional	Permission suggested pertaining to adding a store to the account.
PERMISSIONS: SUGGESTED: add_image_set	string	optional	Permission suggested pertaining to adding an image set to the account.
api_sig	string	required	This is a signature of the request.

API Signature

The signature is comprised of a secret key, known only to the application developer, and all parameters sent in the request. This allows Printfection to verify the request was not modified between the application and Printfection's servers. The secret key can be found in the developer's account by going to 'Applications' -> 'My API Keys' and selecting the appropriate API key.

Follow these steps to generate a valid signature:

1. Take every argument in the request (aside from the api_sig argument) and sort them alphanumerically. They should not be urlencoded and should not contain the ampersand which separates arguments in a URL.
 - a. Make sure the values of each argument stay paired with the correct argument. For example, if your arguments are: dog=5&hippo=14&cat=12 the sorted arguments should look like: cat=12dog=5hippo=14.
2. Append the secret key and calculate the md5sum of the string.
 - a. So using our previous example we would take cat=12dog=5hippo=14 and append the secret key and calculate the md5sum which would look like this:
md5(cat=12dog=5hippo=142f43f0c832f658a7ef4c0552b31b73de)
3. Take the result of the md5sum and use it as the signature.
 - a. So the api_sig argument would be: api_sig= 6a33823107538bc8eb11feb0f5076f49

Sessions

Each time a user authenticates an application a session is created which grants the application permission to access the user's account. By default sessions last 24 hours. When authenticating the application users are given the option to stay logged in to the application. If this option is chosen then the session will never expire (assuming the session is actively used). Only one session per application per user will be valid at any given time.

If a user chooses not to stay logged in when authenticating a desktop application the session will extend itself 24 hours every time the session makes a call to the API. If a user chooses not to stay logged in

when authenticating web applications the session will never extend its 24 hour session length and requires the user to re-authenticate the application.