

Platform - Permissions

Description

Although there is only one set of permissions, permissions can be constrained at both the application level and the user (or session) level. A user cannot give an application more permissions than the application itself has available to it. An application requests a set of permissions from Printfection which are either granted or denied. Once an application has been granted a set of permissions by Printfection it can then ask a user to grant the application access to their account based on those permissions, or a subset of the permissions. The user can further constrain permissions to certain objects in their account.

For instance, Printfection may grant an application delete access (which includes both read and write access) on stores. A user who wants to use the application may then be asked, by the application, to give the application write access on the user's stores. The user may then grant the application write access to one of their stores and read access to another one of their stores. They do not grant the application any access (aka no access) to the remainder of their stores. This means the application now has write and read access to those two stores respectively and can do nothing with any of the other stores the customer owns.

If an application gains additional permissions from Printfection, all users of the application must explicitly grant those new permissions. For instance, if an application is only granted permission to access image sets, then, at a later date, is granted permission to access stores by Printfection, it will not be able to actually access any stores until each user edits their permissions for that application and directly allows the application permission to access stores in their account. In this instance the application should request that each user re-authenticate the application with the new permissions, assuming the application does in fact want permission to access stores.

The best way to require a user to re-authenticate an application is to delete the current session using the `printfection.auth.deleteSession` method then ask them to authenticate the application again. This will create a completely new session with a new set of permissions.

Application Permissions

After Printfection grants permissions to an application, the granted permissions will show up in the developer's account for each application. On the application's page, the developer can then create the permission argument used when authenticating a user. This JSON string has both suggested and required permissions. Users wishing to use that application must select at least the required permissions in order to add the application. The user will see what permissions are suggested but may choose not to grant the suggested permissions. If the suggested permission is higher than the required permission, the suggested permission will be pre-selected, otherwise the required permission will be pre-selected. The user will always have the option to set an object's permission to no access. If the application decides

that it requires a higher level of permission than the user initially granted, it will have to ask the user to re-authenticate with the new permissions.

At any time, users have the ability to change any of the permissions in their account for every application they have an active session with. There are no required permissions when the user changes their permissions in their account. This means a user could change the permission for an application below the application's required permissions. The application must compensate for this and ask the user to re-authenticate the application with the required permissions.

Lifespan

Permissions granted by a user exist as long as the session does. (See Sessions in the Authentication document.) Users can also log into their account and change the permissions for any application they have an active session with at any time. Users can also remove any session, thus removing access to their account for an application, at any time.

Possible Permissions

image_sets

Printfection grants or denies an application the ability to request access to image sets when users authenticate the application. During authentication, users can constrain access by individual image sets. Having access to an image set gives the application access to all images within the image set.

read

Allows access to the image sets and images PFQL tables for all image sets the application has permission to access.

write

This allows uploading of images and modifying information associated with an image or image set, such as the image set's name and an image's keywords. This includes the read permission. If an application has access to multiple image sets images can be moved between the image sets. The images sets must have the same level of permission.

delete

Allows deleting of an image or image set. Images can only be deleted if the image does not exist on any products, or as a section image, or as a store logo. Write access is required on products, sections, or stores to remove the image from each object. This includes both the write and read permissions.

stores

Printfection grants or denies an application the ability to request access to stores when users authenticate the application. During authentication, users can constrain access by individual stores.

Having access to a store gives an application access to all sections and products within the store. A user's personal products exist in a store called 'My Personal Products' which is treated like all other stores.

read

Allows access to the stores, store_sections, and products* PFQL tables. This also allows access to the product pricing methods.

write

This allows modifying an entire store and all objects in a store. This means the store itself can be modified, as well as the sections in the store and the products in the store. Although sections and products cannot be deleted, images on products can be removed as well as section images can be removed. Furthermore, some values can be set to NULL, such as a product or section description, which removes the value. This also allows creating of products and sections within a store. If an application has access to multiple stores then products can be moved between the stores. The stores must have the same level of permission. This includes the read permission.

delete

This allows deleting of products and sections from a store as well as shutting down the entire store. This includes both the write and read permission.

add_store

write

This allows an application to add a store to a customer's account. The application will immediately be granted delete permission to the new store.

add_image_set

write

This allows an application to add an image set to a customer's account. The application will immediately be granted delete permission to the new image set.

rootproducts

read

Most applications which create or edit products will want access to root product information. This includes access to all root_product* PFQL tables and all RootProducts methods. The root_prouduct* tables include sizing and colors for root products as well as possible positioning and sizing of images on products. This is only given to applications by Printfection. Users of the application have no say in this permission as it does not affect a user's data.

carts

The cart methods, explained in the *Checkout* document, allow an application to use a cart hosted by Printfection for Printfection products. Granting only read access is meaningless as applications must first be able to create and modify a cart before reading information about a cart becomes necessary.

write

Applications can access all cart* methods as well as the cart* PFQL tables. There is no need to delete a cart.

sessions

read

All applications are given access to the sessions and permissions tables which allow the application to figure out exactly what kind of permissions it has for a specific session. For instance, an application can find out what stores it has access to and what permission it has per store. The application will always be constrained to the data pertaining to the current session when referencing these tables.