Basic integration

Integrating using the SDK is really simple.

First you need to obtain a copy of the SDK from this page:
https://nitropack.io/download/plugin/nitropack-php-sdk

Unzip it and upload it to your site in a directory of your preference. Most frameworks have a vendor directory suitable for adding third party SDKs.

Next you need to configure your API credentials and load the **bootstrap.php** file from the SDK. We suggest that you load the SDK as early in your project as possible. In many cases the index.php file is the best candidate. Adapt the following lines of code to your needs in order to load the SDK:

```
define("NITROPACK_HOME_URL", "Your home page URL");
define("NITROPACK_SITE_ID", "your site ID");
define("NITROPACK_SITE_SECRET", "your site secret");
include_once “<path to the sdk>/bootstrap.php”;
```

This is it. The bootstrap file will now connect to NitroPack and start handling optimizations and cache serving for you. You can configure the cache behavior from your dashboard at https://nitropack.io/

The SDK will store its config and the cache files in ***<path to the sdk>/NitroPack/SDK/data/*** and it will also create an empty **nitropack_webhooks** file at ***<path to the sdk>/nitropack_webhooks***. It is recommended that you exclude these paths from your version control system.

If your site has a login system and you need to disable caching for logged in users you can do it in one of these two easy ways

1. Load the SDK only when the visitor is not logged in (possibly worse performance compared to option 2)
2. Configure your site to set a cookie whenever a visitor logs in and remove the said cookie upon logging out. Then configure this as an “Excluded cookie” in your NitroPack settings. This is typically faster because it doesn’t require your application to start session or open DB connections.

URL Tagging

The basic integration is great for basic sites which are rarely being updated. However dynamic sites like publishing platforms, e-commerce, etc. typically need to be able to purge specific parts of the cache instead of purging the entire cache. NitroPack achieves this through URL tagging. You can assign one or more tags to a page which can later be used to purge the cache for pages containing a specific tag.

For example you can tag your home page with "page:home", your category pages with "page:category", product pages with "page:product" and so on. You can also add tags for the items on a specific page. For example a product page might also have a tag "product:128", you can then purge that product's page whenever you update the product in your system.

Tags must follow the naming rules described in this document:
https://docs.nitropack.io/#api-Tags-Add

Adding tags can be done with the helper function **nitropack_add_tag($tag)**

Purging the cache

NitroPack provides two ways to update your cache:

1. By purging - can be done with the **nitropack_sdk_purge($url, $tag, $reason)** function
2. By invalidating - can be done with the **nitropack_sdk_invalidate($url, $tag, $reason)** function

In both cases you can update the cache by providing a URL, a tag or both. If none is provided the entire cache is going to be updated. The reason parameter accepts a human readable message describing the reason of the purge. This goes into the service usage logs, so you can later review the behavior of your integration.

The difference between purging and invalidating is explained in this document:
https://help.nitropack.io/en/articles/3281310-what-is-the-difference-between-invalidate-and-purge

Enjoy NitroPack :)