Many websites using Search and Navigation (aka Episerver Find) perform unnecessary indexing which results in additional load on the search backend and indirectly slower search activities. To be clear, Search and Navigation is multi-tenant, so one client can harm another. In a worst-case scenario this could lead to search being completely unavailable or degraded for customers on a cluster. There are safeguard exceptions such as 429 (too many requests) and 413 (entity too large) however improper indexing can still cause major issues. Therefore, excessive indexing should be avoided whenever possible.

## Major pitfalls to avoid:

1. Do not run the Episerver Find Indexing or any other indexing job unless code changes have been made. Running the out of the box indexing job indexes every content item on the website (that is not excluded via Find conventions). Instead use the built-in event driven indexing which automatically indexes content upon content additions, changes, and deletions.

Many clients **completely** disable event indexing to improve the performance of their import job(s). Consequently, no content is indexed, and the indexing job must be run. Event indexing can be **partly** disabled to avoid the indexing job requirement with just two lines of code.

At the beginning of job code.

EPiServer.Find.Cms.EventedIndexingSettings.Instance.ScheduledPageQueueEnabled = false;

At the end of job code.

EPiServer.Find.Cms.EventedIndexingSettings.Instance.ScheduledPageQueueEnabled = true;

With those job modifications, changes are added to the indexing queue during the job. Once the job completes indexing automatically processes the items in the queue. This results in a significantly faster import and the indexing job is no longer needed after importing. With those changes don’t forget to re-enable (if it was disabled) event driven indexing.

<episerver.find.cms disableEventedIndexing = "false"/>

Or in code

EPIServer.Find.Cms.EventedIndexingSettings.Instance.EventedIndexingEnabled = true;

Speaking of jobs make sure they only run on the scheduler app service. Manually running the job ignores the *enableScheduler* setting mentioned in this [blog post](https://world.optimizely.com/blogs/Sergey-Vorushilo/Dates/2017/12/scheduled-jobs-setup-in-dxc-service/).

1. Avoid importing entire catalogs. Instead, only import changed items. This not only reduces the amount of indexing but also puts far less load on the database and reduces the time to import new or changed items.
2. Remove unnecessary types and properties from indexing. This reduces the index size, reduces load on the backend and speeds up all search requests (including indexing).
[Removing page types](https://docs.developers.optimizely.com/digital-experience-platform/v1.1.0-search-and-navigation/docs/indexing#customizing-pages-to-be-indexed)
[Ignoring properties](https://docs.developers.optimizely.com/digital-experience-platform/v1.1.0-search-and-navigation/docs/indexing-search-and-navigation#ignoring-properties)