

Personalization and Cache

How Personalization and Cache work, in a few words

- **Out of the box with 5.3:** cache is bypassed for pages with variants. (we removed the contention issues that can occur when doing header negotiation).
 - When installing the Personalization module, your cache configuration will be automatically adapted (if it is using the default cache policy) to use a new CachePolicy and Store CacheExecutor, which simply result in bypassing the cache for pages which have variants or are personalized. (info.magnolia.personalization.cache.BypassUncacheableEntriesPolicy and info.magnolia.personalization.cache.BypassVariantsCacheStore respectively)
- **Possible:** customize cache key to include your traits
 - We can't make that work for everyone without a lot of effort (would need to be configurable, magic, or dumbly add all traits),
 - We don't believe this is a good solution: it only works if a majority of pages are personalized AND you have a small amount of traits AND those traits have a small amount of possible values. (It would be completely pointless with a "date" trait, and even just 2 traits with 2 possible values would increase the cache size by 4)
- **Future:** we will implement another caching mechanism which should allow caching pages with variations while maintaining a reasonably sized cache.
 - It will require for trait detectors to be executed before the cache filter, which means performance of these detectors will be paramount to this mechanism working properly.
 - Tech details: cache policy will work with an extra indirection. See sequence diagram below.
- **Future:** we are investigating component-level caching and Edge Side Includes. See [Concept - partial caching](#).

