TYPO3.Flow - Feature # 46318

Status:	New	**Priority:**	Should have
Author:	Norbert Sendetzky	**Category:**	Cache
Created:	2013-03-15	**Assigned To:**	
Updated:	2013-03-15	**Due date:**	
PHP Version:	5.3		
Has patch:	No		
Complexity:	medium		
Subject:	[caching framework] Extend cache interface to handle multiple entries		

Description

The current BackendInterface and FrontendInterface is not suitable for retrieving or storing several entries at once because the interfaces are limited to hand over cache entries one by one. Especially for the Redis and the database backend, being able to retrieve several entries at once speeds up the operation drastically.

One option would be to extend the get(), set() and remove() methods to also allow arrays of identifiers. The advantage would be that the interfaces don't have to be changed but on the other side, all cache backends have to be adapted to be able to accept arrays. Another disadvantage is that the returned values for single entries and lists of entries must be different.

The other option is to add new methods to the interface:
- getList(array $entryIdentifiers)
- setList(array $entryIdData, array $tags = array(), $lifetime = NULL)
- setList(array $entry) // alternative way: data, tags and lifetime in associative array
- removeList(array $entryIdentifiers)

Then, we can provide default implementations in AbstractBackend that call the get(), set() or remove() methods in a loop for all backends that are not able to fetch, store or delete a bunch of entries. The Redis and PDO backend can overwrite these methods to provide their own optimized implementations.
There are two alternatives for setList(): Either the tags and the lifetime is the same for all entries (easier to understand and less error prone) or data, tags and lifetime is stored in an associative array per entry. The last option would be much more flexible and this might be a better solution for different use cases.

Can the caching interfaces extended in on of these ways and are you willing to integrate a patch for this?

History

#1 - 2013-03-15 12:18 - Christian Kuhn

- Project changed from Core to TYPO3.Flow

- Category deleted (Caching)

- Target version deleted (6.1.0)

Moved the issue to Flow since the cf in TYPO3 CMS is a backport of the flow part. Such bigger changes should be implemented in FLOW in the first place.

#2 - 2013-03-15 12:19 - Christian Kuhn

- Category set to Cache

- Has patch set to No