TYPO3.Flow - Feature # 47951

Status:	New	**Priority:**	Should have
Author:	Alex no-lastname-given	**Category:**	Persistence
Created:	2013-05-05	**Assigned To:**	
Updated:	2013-05-21	**Due date:**	
PHP Version:			
Has patch:	No		
Complexity:			
Subject:	Warn if persistence stack is not empty at the end of a get-request		

Description

Because the flow framework follows a rfc-standard, persistanceManager::persistAll() is not executed automatically for a get request. The framework should at least throws an exception if the persistance stack is not empty at the end of a get request.

Related issues:

related to TYPO3.Flow - Feature # 51570: Unpersisted changes in Safe Requests...	**Resolved**	**2013-08-30**

History

#1 - 2013-05-08 12:51 - Bastian Waidelich

Hi Alex no-lastname-given, ;)

interesting idea (at least for development context).

The code that is currently responsible for this behavior is in the Package.php file of Flow:

```
$dispatcher->connect('TYPO3\Flow\Mvc\Dispatcher', 'afterControllerInvocation', function($request) use($bootstrap) {
  if (!$request instanceof \TYPO3\Flow\Mvc\ActionRequest || $request->getHttpRequest()->isMethodSafe() !== TRUE) {
    $bootstrap->getObjectManager()->get('TYPO3\Flow\Persistence\PersistenceManagerInterface')->persistAll();
  }
});
```

Technically this could probably be changed to s.th. like this:

```
$dispatcher->connect('TYPO3\Flow\Mvc\Dispatcher', 'afterControllerInvocation', function($request) use($bootstrap) {
  if ($request instanceof \TYPO3\Flow\Mvc\ActionRequest) {
    return;
  }
  $persistenceManager = $bootstrap->getObjectManager()->get('TYPO3\Flow\Persistence\PersistenceManagerInterface');
  if ($request->getHttpRequest()->isMethodSafe() !== TRUE) {
    $persistenceManager->persistAll();
  } elseif ($bootstrap->getContext()->isDevelopment() && $persistenceManager->hasChanges()) {
    // log exception
  }
});
```

Three things to keep in mind:

1. There is no "PersistenceManager::hasChanges()" method. Doctrine allows you to detect modifications AFAIK but we'd have to add this to our API

1. We can't throw an exception in the "afterControllerInvocation" slot because that's executed too late – it could be thrown before maybe or the exception could be logged instead

1. We currently only omit the "persistAll()" call for safe requests but the plan is not to load the fully fledged persistence layer for safe requests at some point (for performance reasons). But probably we need to be able to initialize it "lazily" anyways in order to allow people to call persistAll() manually

#2 - 2013-05-21 11:51 - Robert Lemke

- Target version set to 2.1

#3 - 2013-05-21 13:28 - Robert Lemke

- Target version deleted (2.1)