Base Distribution - Task # 53257

Status:	Closed	**Priority:**	Should have
Author:	Bastian Waidelich	**Category:**	Low Level
Created:	2013-10-31	**Assigned To:**	Bastian Waidelich
Updated:	2014-12-16	**Due date:**	
Subject:	Content Repository API		

Description

We need a solid foundation for all kinds of interactions with the TYPO3 Content Repository in order to streamline client/server communication and provide open endpoints for 3rd parties.

Some measures towards a stable API I could imagine:

1. Encapsulate NodeDataRepository calls into a public service (e.g. "NodeRepository")

2. Merge functions with similar functionality

If you look at the list of "lookup" interactions below you can see that we do similar jobs multiple times. I would suggest following changes:

- Replace Workspace arguments by ContextInterface arguments (and provide some easy way to retrieve a context from a workspace if we see the need for it). This will also render arguments like "$includeRemovedNodes" obsolete.

- The service should only return NULL, NodeInterface or QueryResultInterface<NodeInterface> if possible (Until we rewrote the queries to use DQL we could create a NodeQueryResult that implements a subset of the QueryResultInterface)

3. Extend the Neos "Node REST API" so that it provides common operations (using the NodeRepository)

4. Replace existing client/server interactions (ExtDirect, possibly some Backend*Controller) by interactions with the REST API

History

#1 - 2013-10-31 18:47 - Bastian Waidelich

The current direct interactions with the NodeDataRepository that should be replaced by proper API calls:

{{toc}}

Lookup

Find one node by identifier & workspace

```
NodeDataRepository::findOneByIdentifier($identifier, Workspace $workspace);
```

Usages:

TYPO3.Neos:

```
Module\Management\WorkspacesController::indexAction();
Service\NodeController::showAction();
TypoScript\ConvertNodeUrisImplementation::convertNodeIdentifierToUri();
```

TYPO3.TYPO3CR:

```
Domain\Model\Workspace::publishNode()
TypeConverter\NodeConverter::convertFrom()
```

Find one node by path and context

```
NodeDataRepository::findOneByPathInContext($path, ContextInterface $context);
```

Usages:

TYPO3.TYPO3CR:

```
Domain\Model\Node::getParent()
Domain\Model\NodeData::getParent()
```

Find one node by path and workspace

```
NodeDataRepository::findOneByPath($path, Workspace $workspace);
```

Usages:

TYPO3.TYPO3CR:

```
Domain\Model\NodeData::getNode()
```

Find first node by parent and type in context

```
NodeDataRepository::findFirstByParentAndNodeTypeInContext($parentPath, $nodeTypeFilter, ContextInterface $context);
```

Usages:

TYPO3.TYPO3CR:

```
Domain\Model\Node::getPrimaryChildNode()
Domain\Model\NodeData::getPrimaryChildNode()
```

Find nodes by parent and type in context

```
NodeDataRepository::findByParentAndNodeTypeInContext($parentPath, $nodeTypeFilter, ContextInterface $context, $limit = NULL, $offset = NULL);
```

Usages:

TYPO3.TYPO3CR:

```
Domain\Model\Node::getChildNodes()
Domain\Model\NodeData::getChildNodes()
```

Find nodes by parent node and type

```
NodeDataRepository::findByParentAndNodeTypeRecursively($parentPath, $nodeTypeFilter, Workspace $workspace, $limit = NULL, $offset = NULL, $includeRemovedNodes = FALSE)
```

Usages:

TYPO3.Neos:

```
Service\PluginService::getNodes()
```

Find nodes by workspace

```
NodeDataRepository::findByWorkspace(Workspace $workspace);
```

Usages:

TYPO3.Neos:

```
Service\PublishingService::getUnpublishedNodes()
```

Find nodes on path by context

```
NodeDataRepository::findOnPathInContext($pathStartingPoint, $pathEndPoint, ContextInterface $context, $nodeTypeFilter = NULL);
```

Usages:

TYPO3.TYPO3CR:

```
Domain\Model\Node::getClosestAncestor()
Domain\Model\Node::getNode()
Domain\Service\Context::getNodesOnPath()
```

Find arbitrary nodes (custom queries)

```
NodeDataRepository::createQuery(); // …
```

Usages:

TYPO3.Neos:

```
Domain\Service\NodeSearchService::findByProperties()
```

Counting

Count nodes by parent and type in context

```
NodeDataRepository::countByParentAndNodeType($parentPath, $nodeTypeFilter, Workspace $workspace, $includeRemovedNodes = FALSE);
```

Usages:

TYPO3.TYPO3CR:

```
Domain\Model\Node::getNumberOfChildNodes()
Domain\Model\NodeData::getNumberOfChildNodes()
```

Count nodes by workspace

```
NodeDataRepository::countByWorkspace(Workspace $workspace)
```

Usages:

TYPO3.TYPO3CR:

```
Domain\Model\Workspace::getNodeCount()
```

Interaction with single nodes / miscellaneous

Add a new node

```
NodeDataRepository::add($nodeData)
```

Usages:

TYPO3.TYPO3CR:

```
Domain\Model\Node::materializeNodeData()
Domain\Model\NodeData::createSingleNode()
Domain\Model\Workspace::initializeObject()
```

Update a node

```
NodeDataRepository::update($nodeData)
```

Usages:

TYPO3.TYPO3CR:

```
Domain\Model\Node::update()
Domain\Model\NodeData::setName()
Domain\Model\NodeData::setIndex()
Domain\Model\NodeData::update()
Domain\Model\NodeData::remove()
```

Remove a single node:

```
NodeDataRepository::remove($node);
```

Usages:

TYPO3.Neos:

```
Module\Management\WorkspacesController::discardNodeAction()
Module\Management\WorkspacesController::publishOrDiscardNodesAction()
Module\Management\WorkspacesController::discardWorkspaceAction()
```

TYPO3.TYPO3CR:

```
Domain\Model\NodeData::remove()
Domain\Model\Workspace::publishNode()
```

Remove all nodes:

```
NodeDataRepository::removeAll();
```

Usages:

TYPO3.Neos:

```
Command\SiteCommandController::pruneCommand()
Setup\Step\SiteImportStep::importSite();
```

Change node index

```
NodeDataRepository::setNewIndex(NodeData $node, $position, NodeData $referenceNode = NULL)
```

Usages:

TYPO3.TYPO3CR:

```
Domain\Model\NodeData::moveBefore()
Domain\Model\NodeData::moveAfter()
Domain\Model\NodeData::moveInto()
Domain\Model\NodeData::createSingleNode()
```

Manually persist nodes

```
NodeDataRepository::persistEntities();
```

Usages:

TYPO3.Neos:

```
Service\BackendRedirectionService::getAfterLoginRedirectionUri()
```

TYPO3.TYPO3CR:

```
Domain\Model\NodeData::setName()
```

#2 - 2013-12-04 19:03 - Bastian Waidelich

- Status changed from New to Accepted

- Assigned To set to Bastian Waidelich

#3 - 2014-12-16 11:38 - Bastian Waidelich

- Status changed from Accepted to Closed

- Remaining (hours) set to 0.0

In the meantime the CR has been changed drastically and IMO a rewrite from scratch (maybe based on CQRS?) is more realistic than a substantial refactoring