Operations descriptions

Documentation of the different operation primitives that the protocol describes

Introduction

Common for all operations is that they follow the flow described in Message flow and that all operations are targeted at a specific Collection, the only exception is the GetStatus operation as it regards components.

Operations that regard collections all act on a single file and some also all files. When working with all files in a collection, the protocol supports a paging mechanism to ensure that a large number of results won't cause problems for the various components (pillars, clients, messagebus etc.).

The following sections describe the major functionalities of each operation, for detailed fields in the protocol see Message format.

The descriptions of the various operations here does not cover the exact details about the expected behavior and corner cases, but just the broader conceptual parts. Details are kept in integration tests and acceptance tests in the reference code for two reasons:

1. To ensure documentation and reality (implementation) does not drift apart
2. Enable automatic verification of components and their interaction.

The tests can be found here.

- DeleteFile — Operation to remove a file from one or more pillars in a collection
- GetAuditTrails — Operation to retrieve audittrails for a specific collection from one or more contributors
- GetChecksums — Operation to get checksums from one or more pillars in a given collection.
- GetFile — Operation to retrieve a single file from a collection.
- GetFileIDs — Operation to get fileIDs from one or more pillars in a given collection.
- GetFileInfos
- GetStatus — Operation to retrieve status from components in a repository
- PutFile — Operation to ingest a file on one or more pillars in a collection.
- ReplaceFile — Operation to replace a file on one or more pillars in a collection.

Single data object for get, put, delete and replace

It is only possible to perform get,put,replace and delete operations on a single file.

The reason for this choice is to have a simple protocol. Requests on multiple data objects would complicate data transmission of the different files as well as error handling.