In the following, the “local file system” is assumed to be a clone of a GitHub repository.

# Workflows

The following workflows are desired for working with Dokimion.

## Backup a project on a server to the local file system

The backup files can be used for:

* Restoring individual test cases if they get corrupted or deleted.
* Restoring a whole project.

## Copy a whole project from one server to another

Do we need to be able to specify a subset of the test cases?

If so, would filtering using project attributes be helpful?

Would backing up the project from one server to the file system and then restoring that project to the other server be “good enough”? Two steps instead of one. The files would not need to be put into GitHub.

## Restore a whole project from the local file system

Do we need to be able to specify a subset of the test cases?

If so, would filtering using project attributes be helpful?

## Restore a test case from the local file system

## Create a new test case in Github and send it to a server

This involves hand editing XML files. Perhaps a dedicated editing tool would help?

# Tools

Currently, I have two tools:

* Updater, which saves and restores individual test cases. (But it doesn’t work if a test case is deleted.)
* Cloner, which copies a whole project from one server to another. (But only if the project never existed on the target server.)

I propose obsoleting the Cloner and extending the Updater tool to:

* Save/restore the project data, not just test case data
* Support server to server copies

# Changes to Dokimion API

We need to be able to undelete a test case. Currently, if we try to PUT the “deleted” field to “false”, the API returns a “Not found” error. I propose allowing PUTs to deleted test cases, but not GETs. Also, the PUT to a deleted test case should not perform the “last modified time” check.