

Visbo ReST Server and UI Testing

Here we describe the Testing Environment for the ReST Server and the UI

Testing of the ReST Server

We use Postman (settings of the Postman [Settings of Postman.txt](#)) as the test tool to verify if the VISBO ReST Server works as expected.

We manage the following areas:

- 00 Flow

this is the complete flow of testing VC, VP, VPV & VPF from creation of the objects to the manipulation, the deletion and the access for different user types. Verifying that the permission system works and all the ReST calls works as expected.

- 98 Cleanup

this is the cleanup flow, that is mainly focused on destroying deleted test VCs, so this is on the one hand side also a test of the destroy function, but mainly a cleanup of the system to get rid of the many test VCs, VPs, VPVs and VPFs

The flows are part of the ReST Server package in git, so it could always verified what kind of tests were done for an older version. To allow multiple users to work with the same flow, we have formerly used individual flows in the personal workspace and asked other users to adopt their flow in the same form. This was error prone but a working solution for the option to use postman for free. In the meantime postman allows small teams of 3 users in the free plan, so we change to the team approach.

The rule is now, that there is a teamspace, where the flows were managed. The flow should not be changed anymore directly, but the developer of the test make a fork of the flow with his initials as label.on, we will use merge / pull requests to integrate the changes into the main flow. It is still a good idea to mark the areas where a user wants to change postman tests with the initial in the folder name of the postman test folder.

To setup the test environment for postman the following steps are required

1. Install postman client latest version
2. If no flows are present in teamspace, import the flow from the ReST Server GIT
3. Fork the required flow from teamspace to the personal workspace
4. Import the environments from ReST Server GIT if required

5. Run the complete flow or a part of it
6. Adopt the postman flow for new features or missing checks
7. After finish generate a pull request
the flow has to be moved to teamspace for this during creating the pull request
8. Fork the original flow from teamspace again, to have all the changes integrated

As the environment contains sensitive information like usernames and passwords, I have not found a solution to use the flow from team space but use the environment from local workspace, so I decided to fork and merge it between teamspace and workspace.