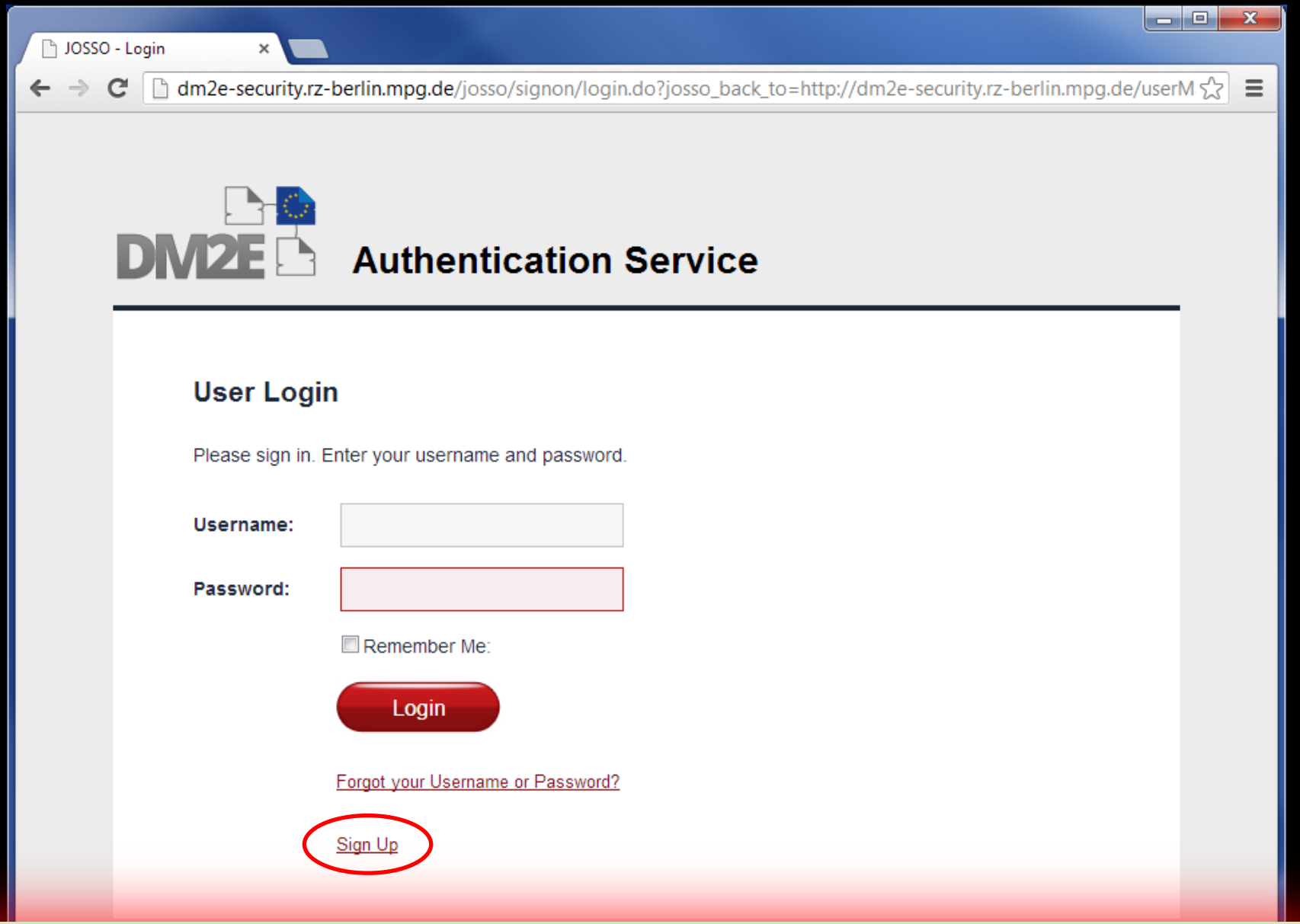
A Short OmNom Tutorial

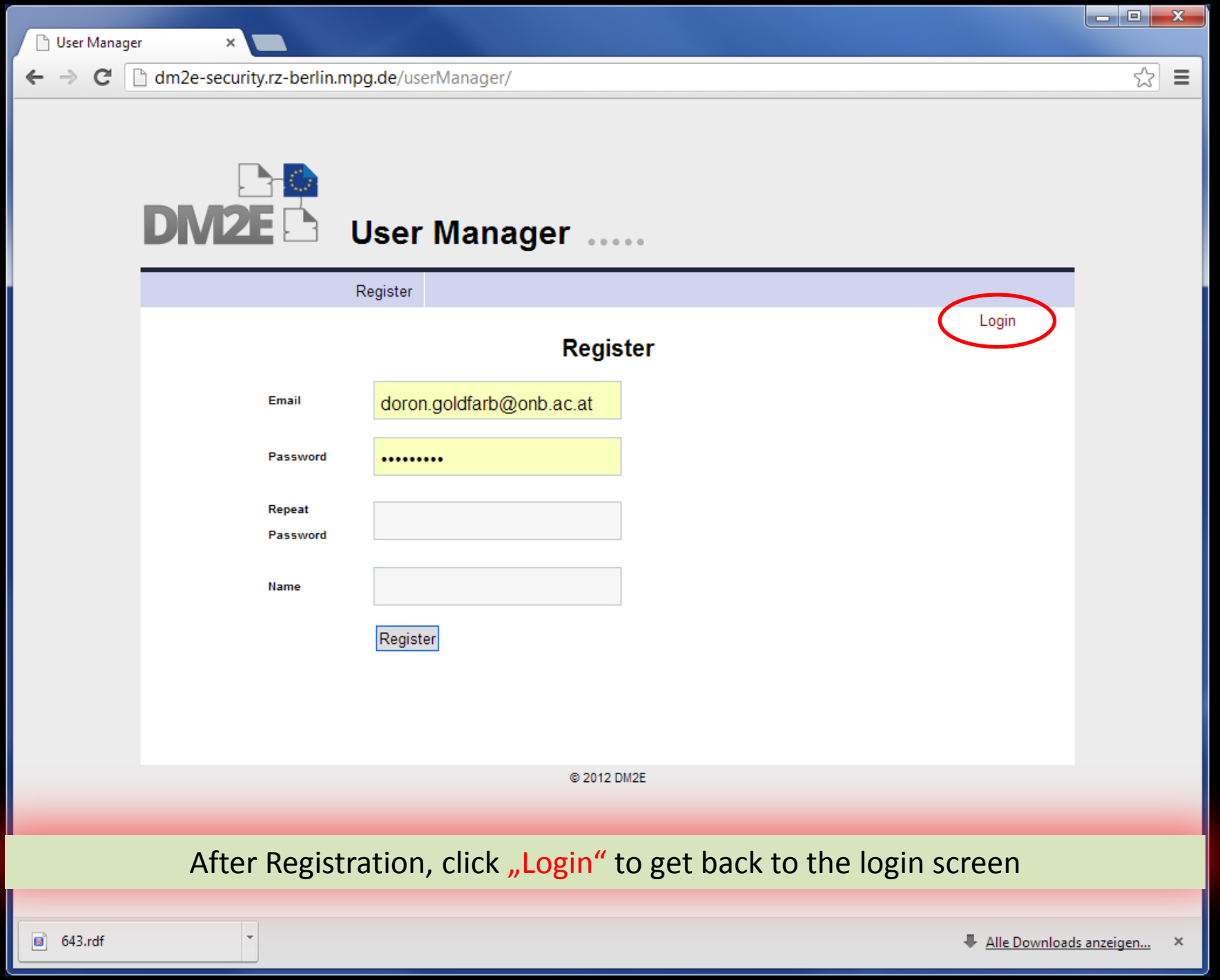
Doron Goldfarb

Austrian National Library

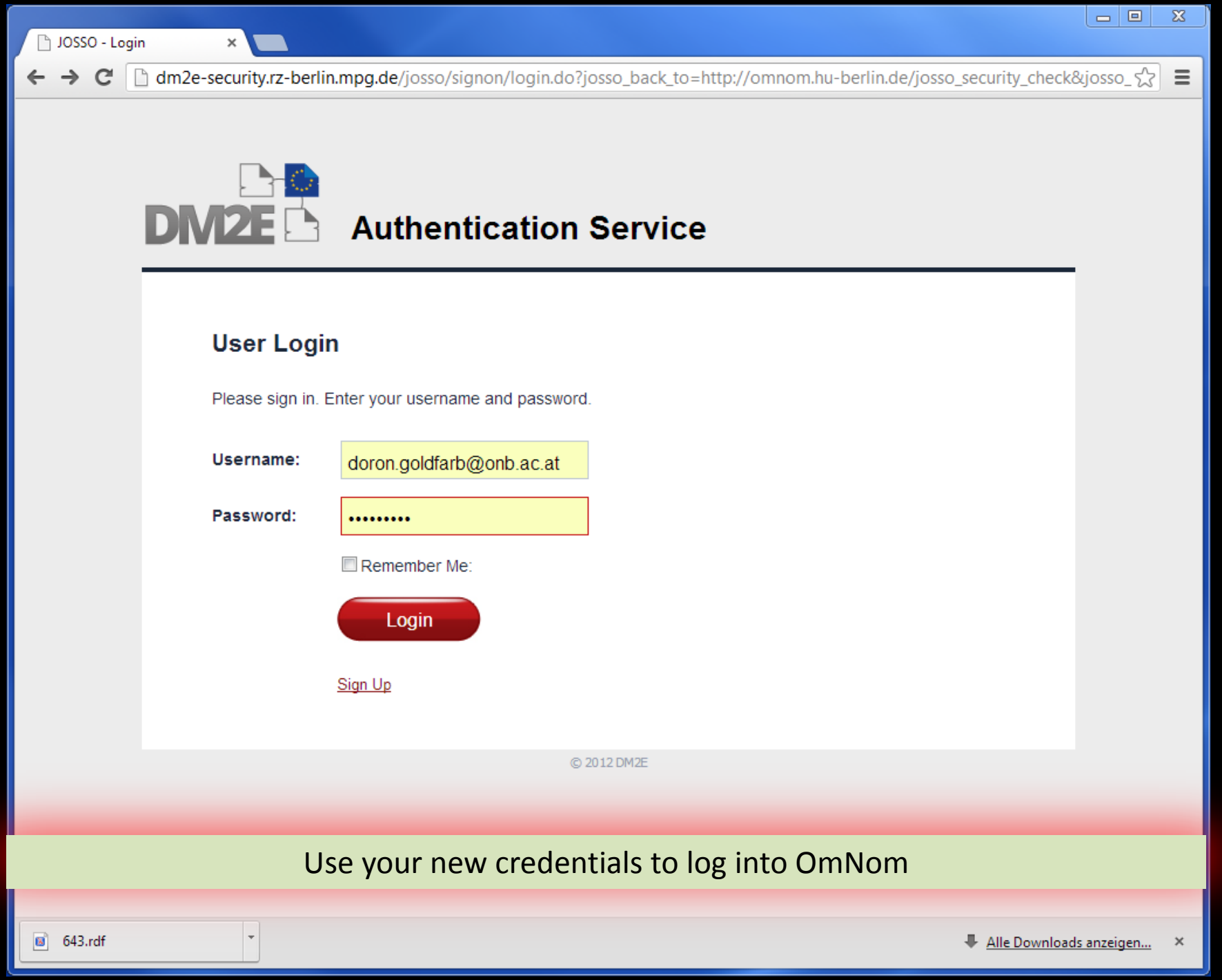
Disclaimer: This is meant as a short step by step tutorial for new users to get acquainted with OmNom in order to participate in its evaluation.
OmNom is subject to constant enhancement. Expect things to change regularily!



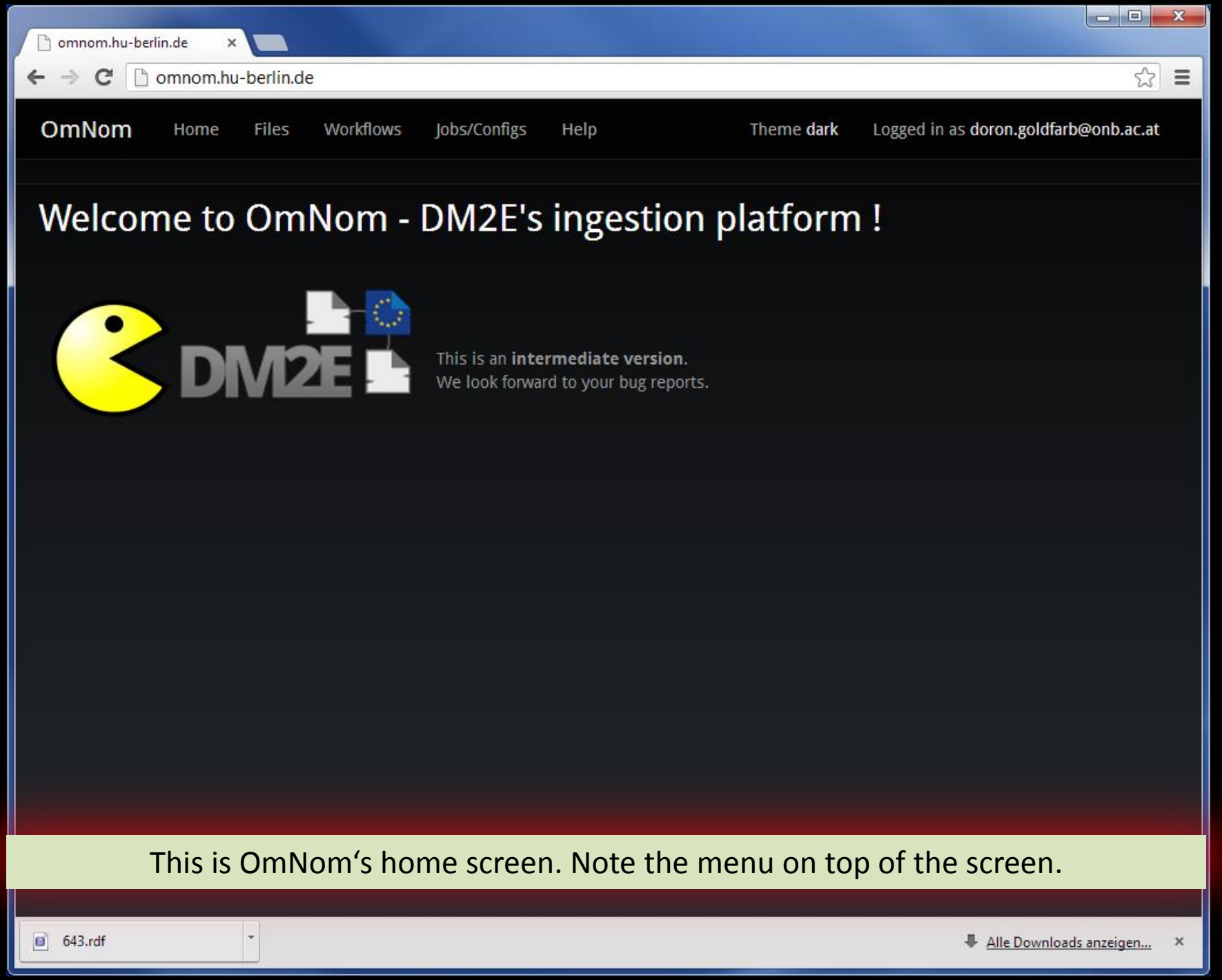
When you first visit http://omnom.hu-berlin.de you will be redirected to the DM2E authentication service. Klick „Sign Up“ to register a new user.



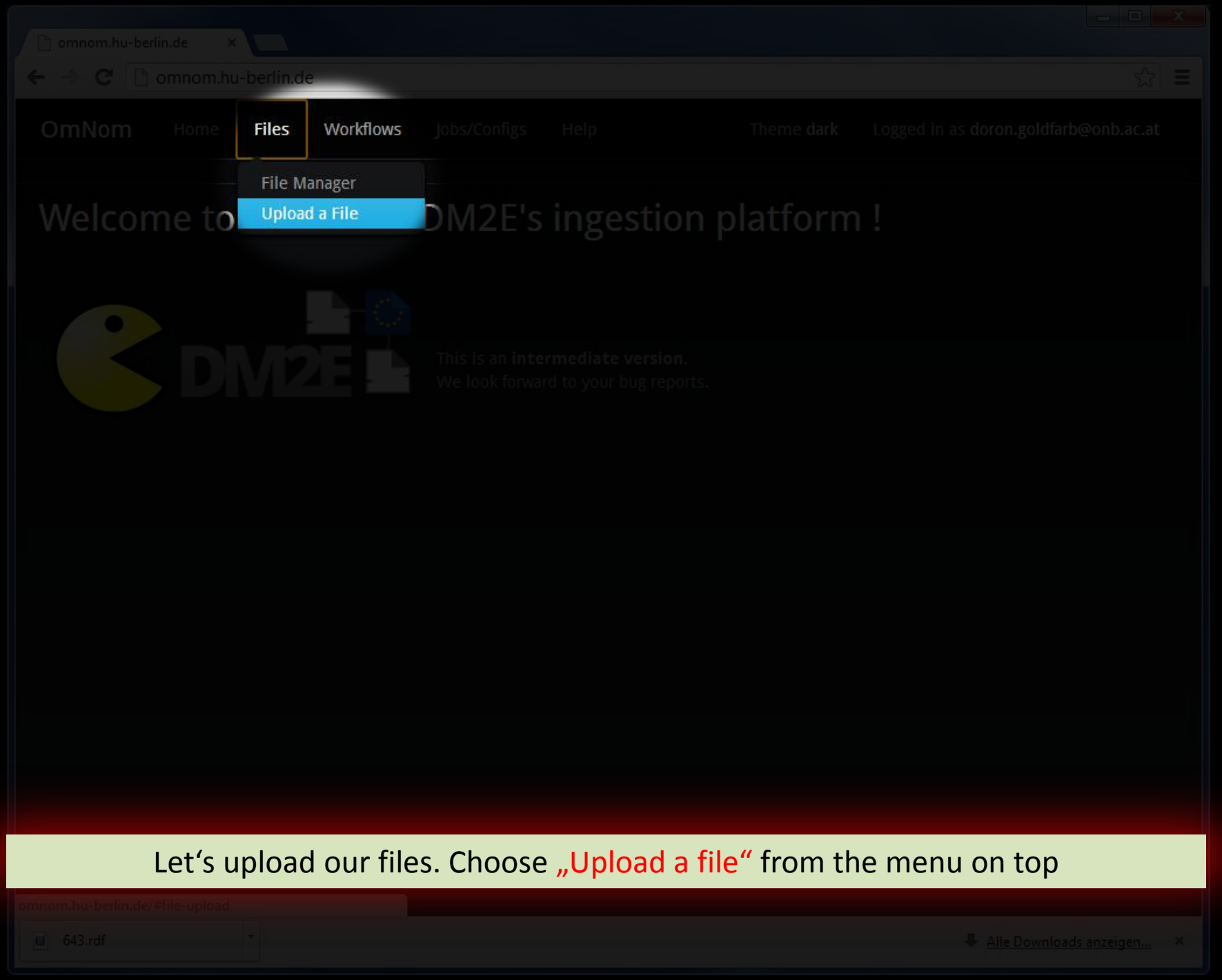
After Registration, click „Login“ to get back to the login screen



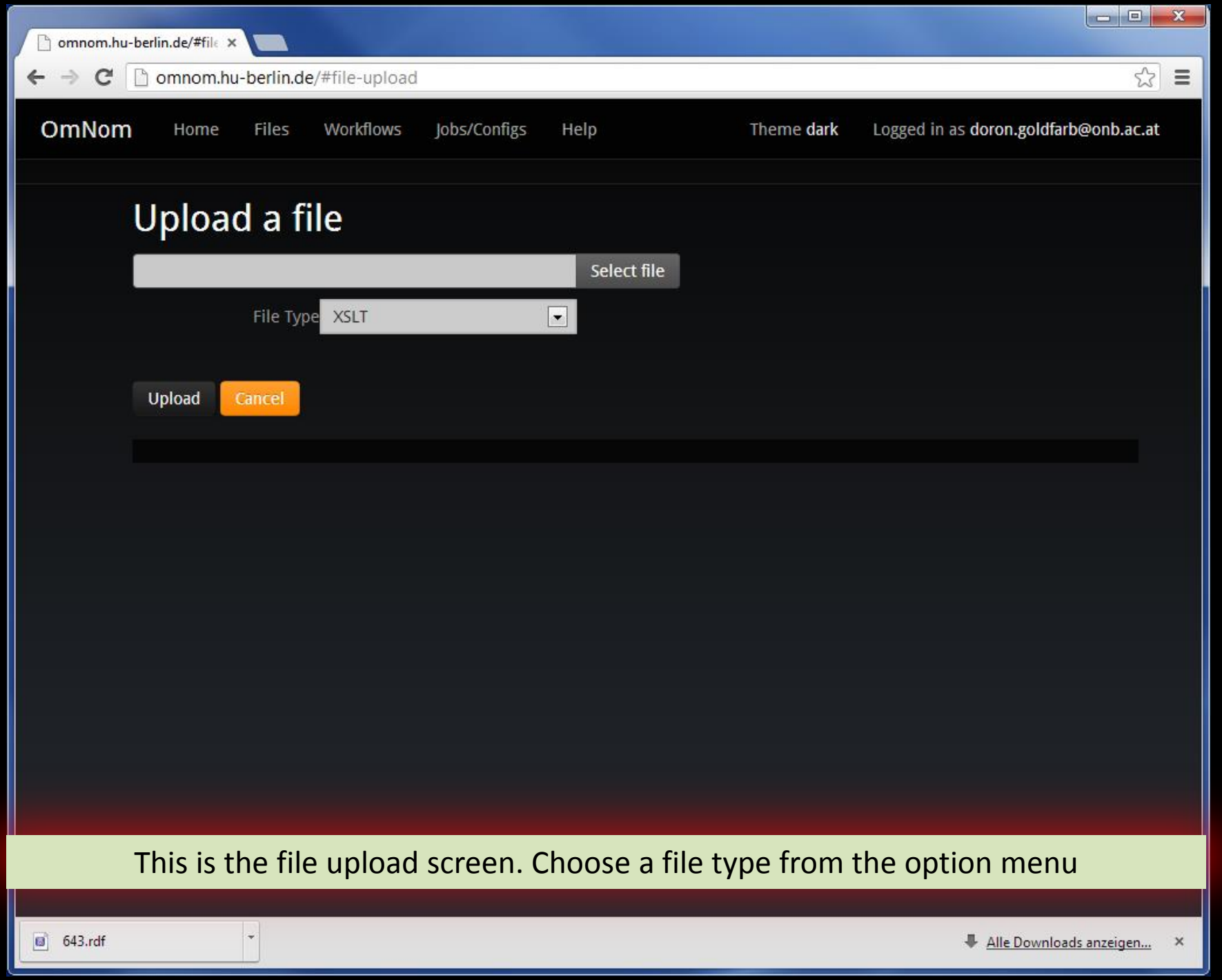
Use your new credentials to log into OmNom



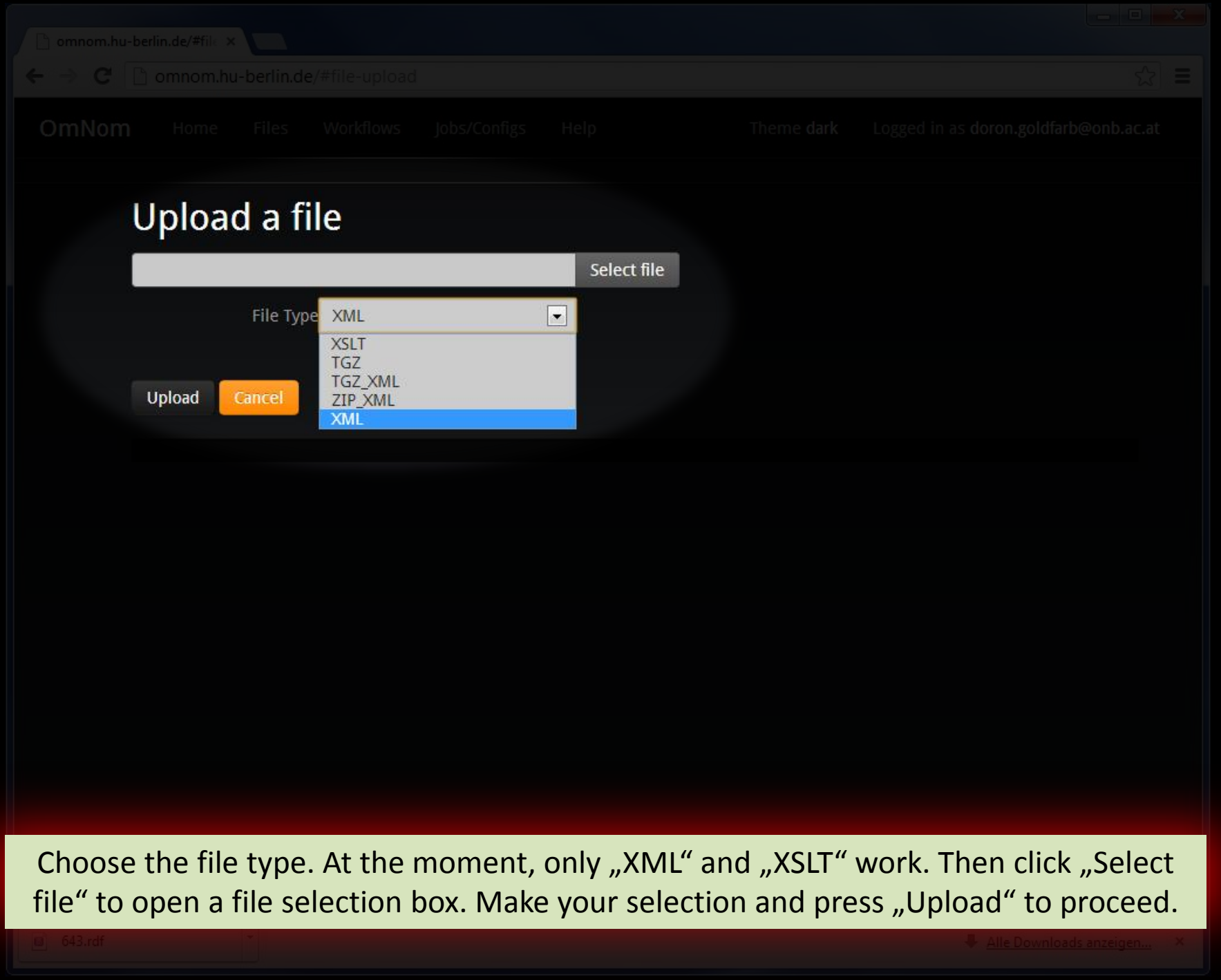
This is OmNom‘s home screen. Note the menu on top of the screen.



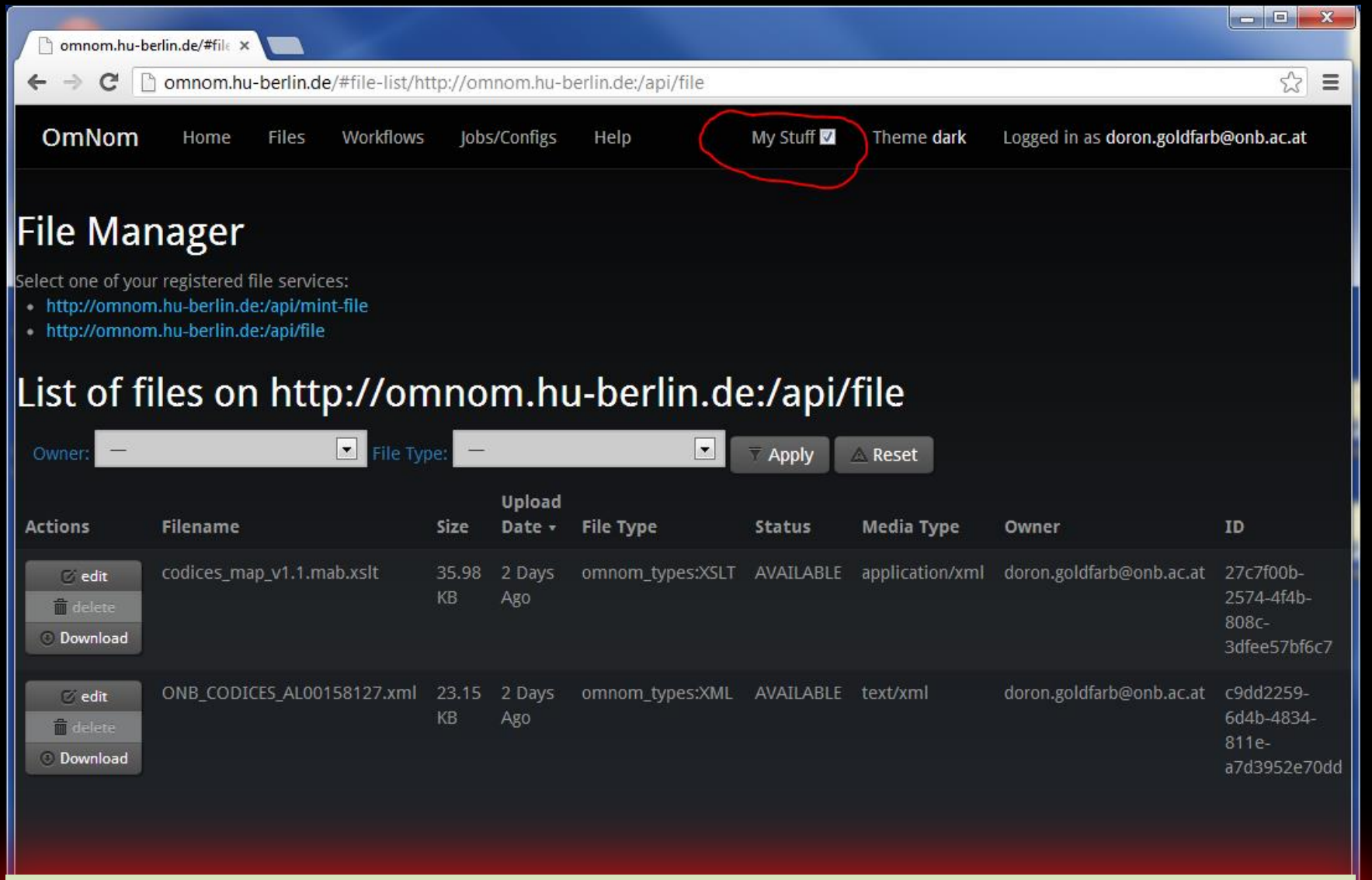
Let‘s upload our files. Choose „Upload a file“ from the menu on top



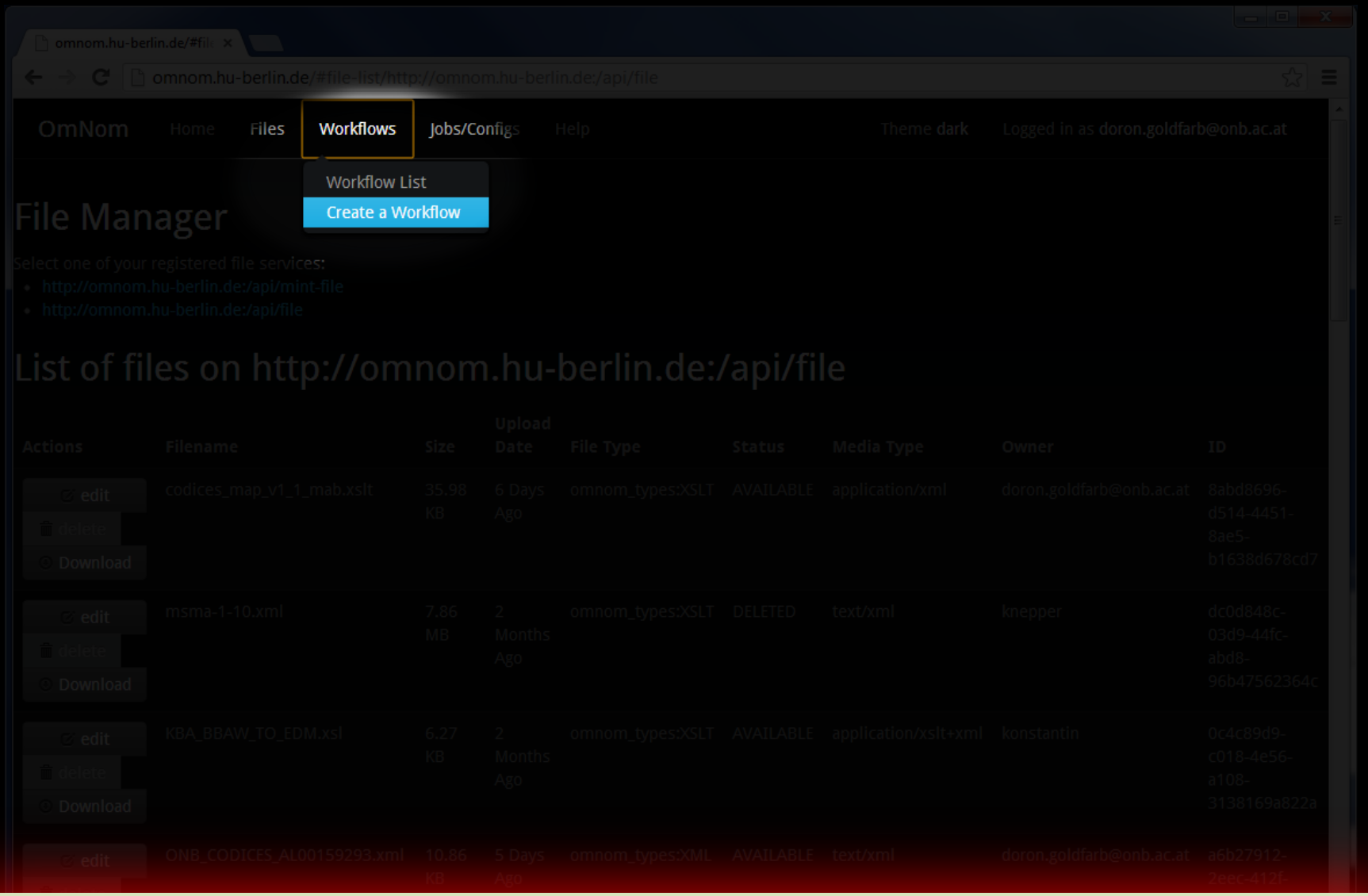
This is the file upload screen. Choose a file type from the option menu



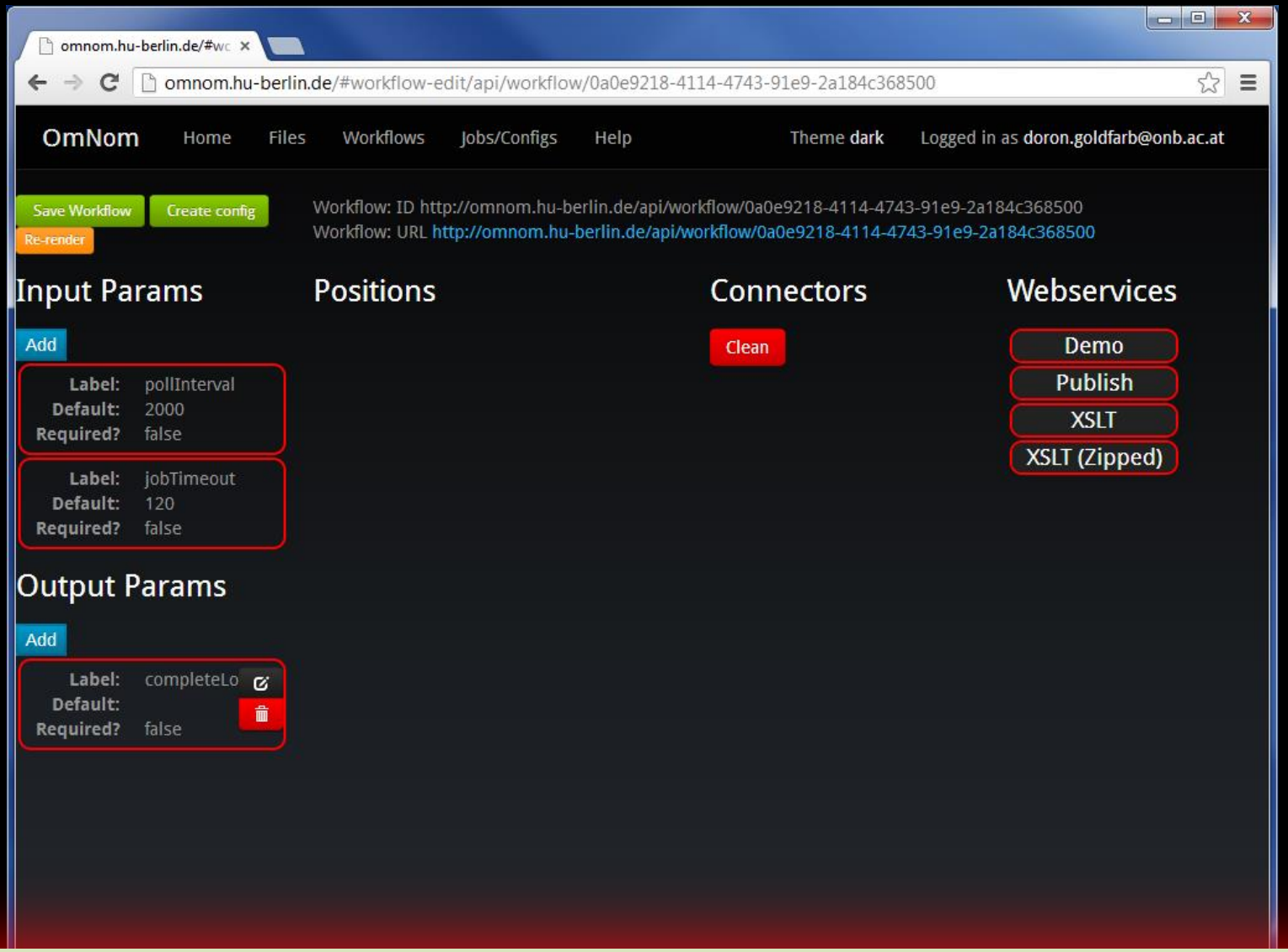
Choose the file type. At the moment, only „XML“ and „XSLT“ work. Then click „Select file“ to open a file selection box. Make your selection and press „Upload“ to proceed.



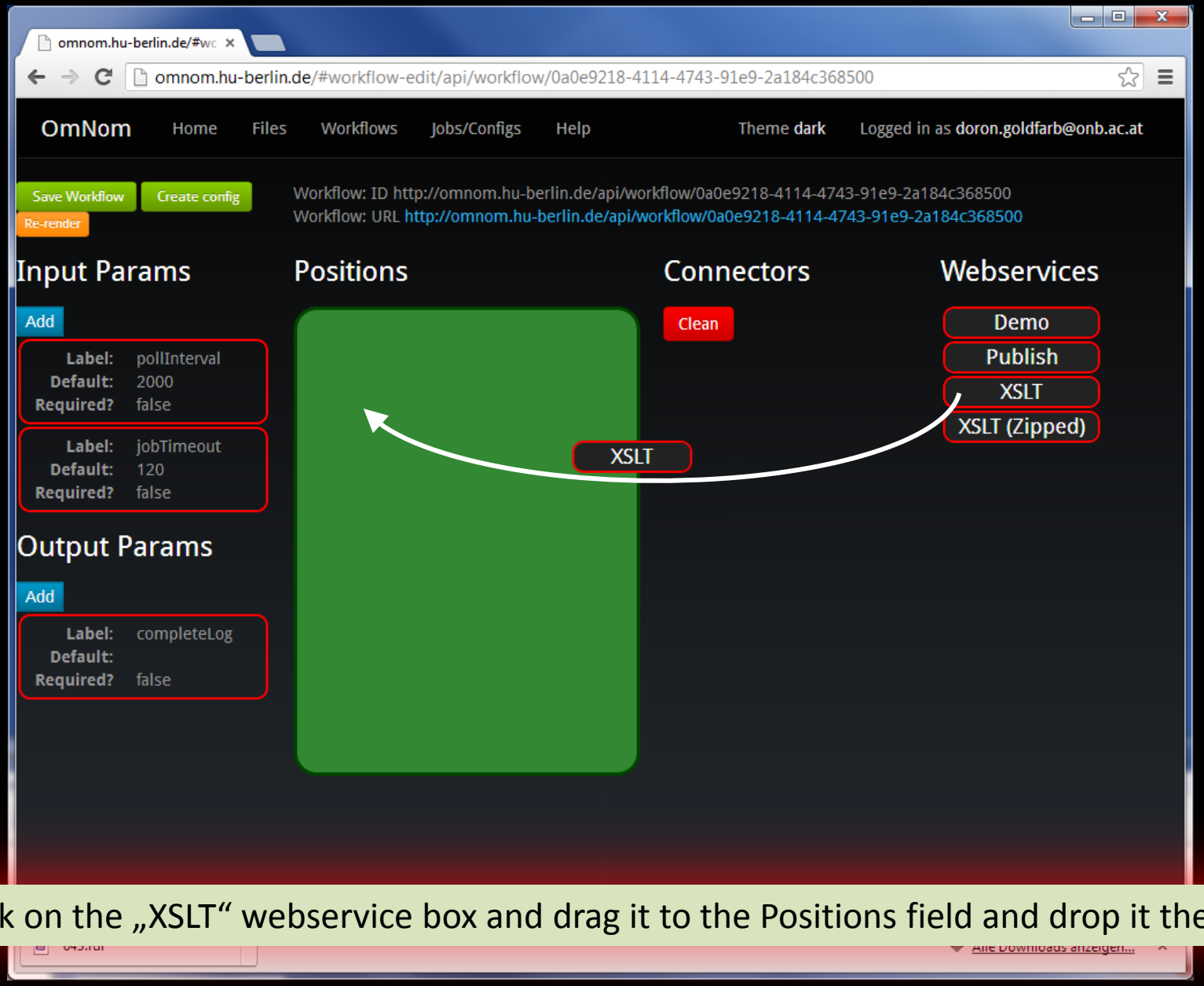
Once your file has been uploaded, you will be redirected to the file manager view for uploaded files. For your convenience, you can tick the „My Stuff“ tickbox on top of the screen in order to limit the displayed files to those uploaded by your user only.



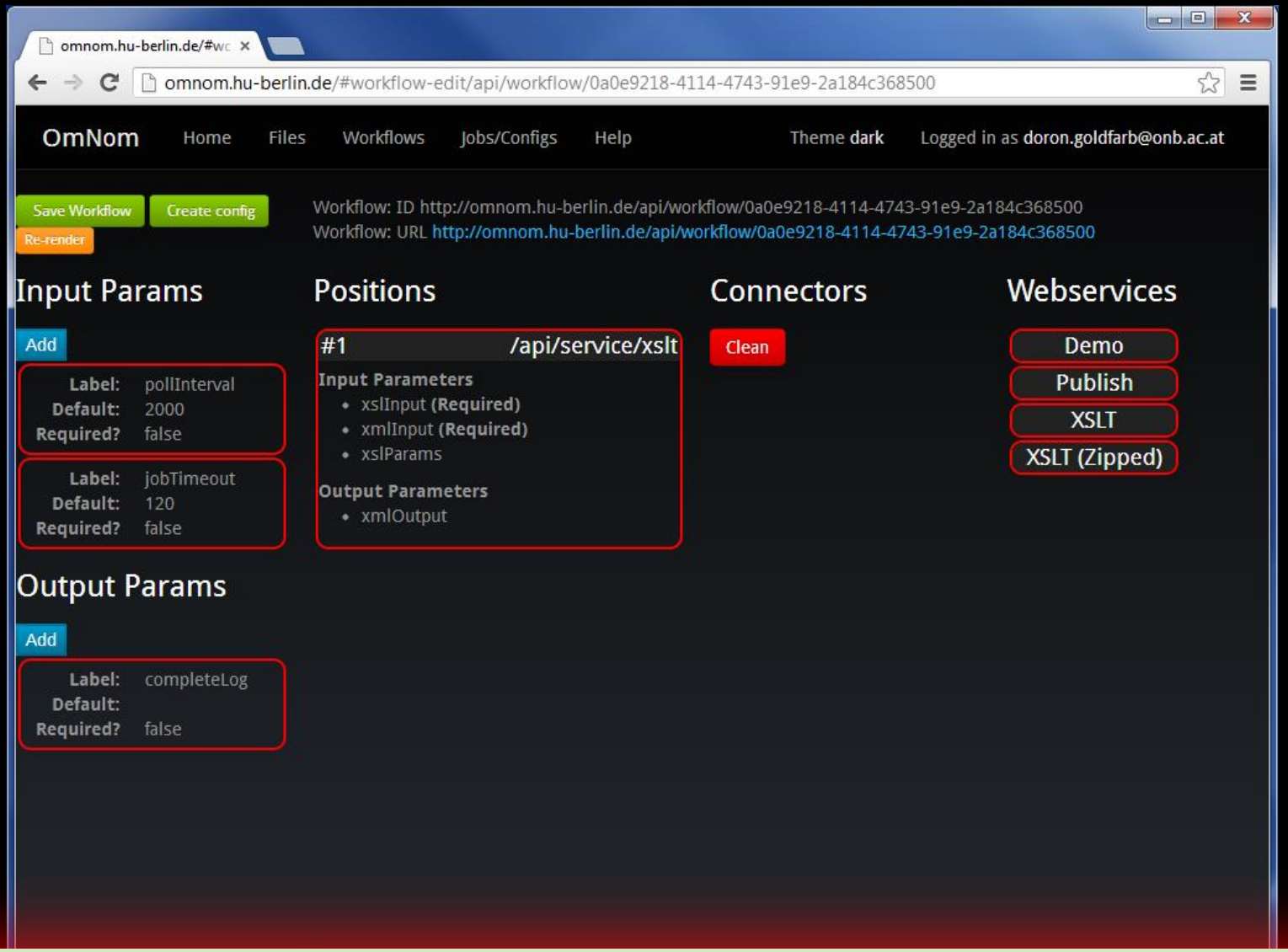
Now we will define a workflow that applies XSLT transformations to source XML files and publishes them to the DM2E Triplestore. Choose „Create a Workflow“ from the „Workflows“ option menu on top of the screen.



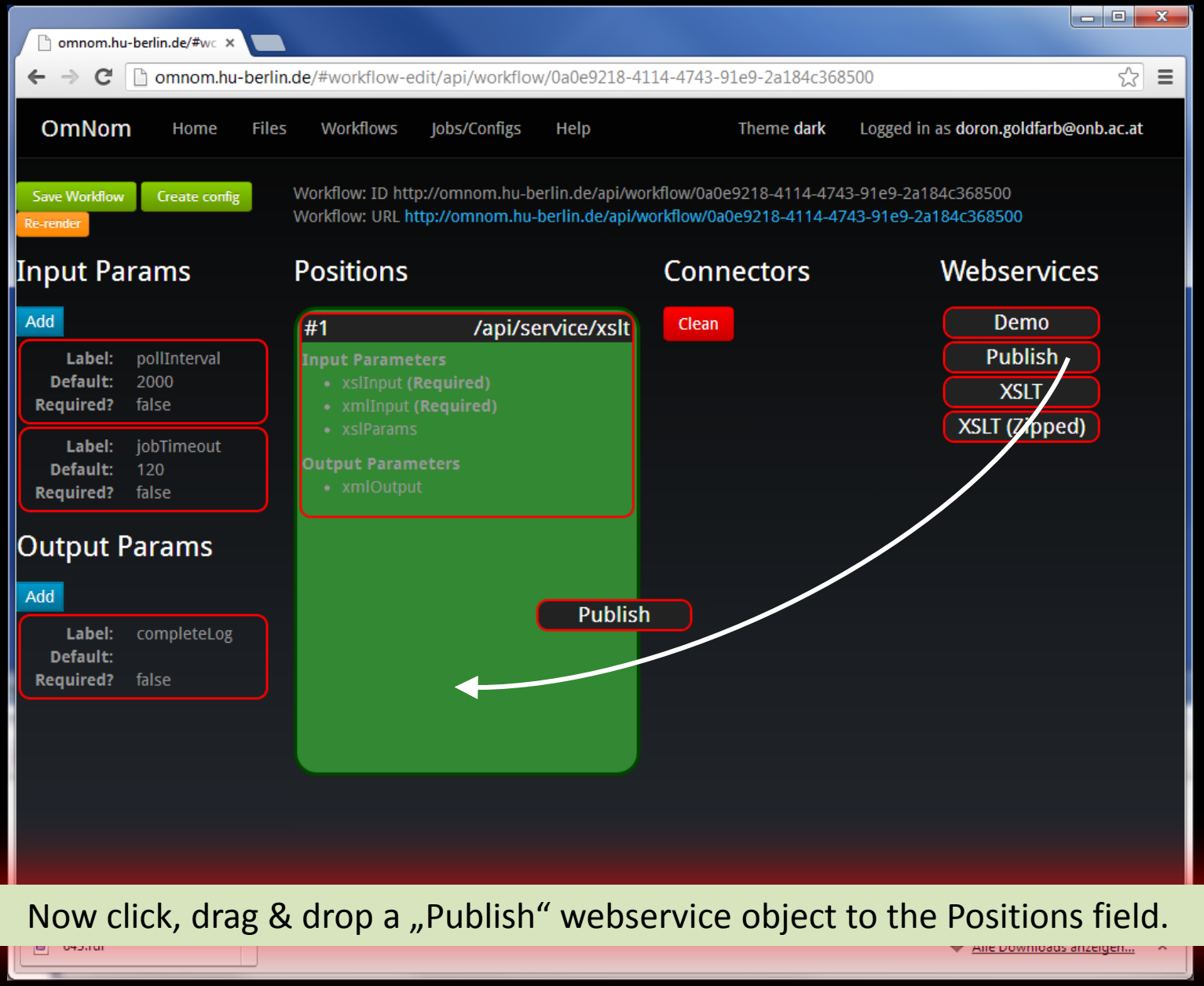
This is the environment to create OmNom workflows. The main elements are the Webservices, Connectors, Positions and Input Parameters. We will now start to configure a workflow.



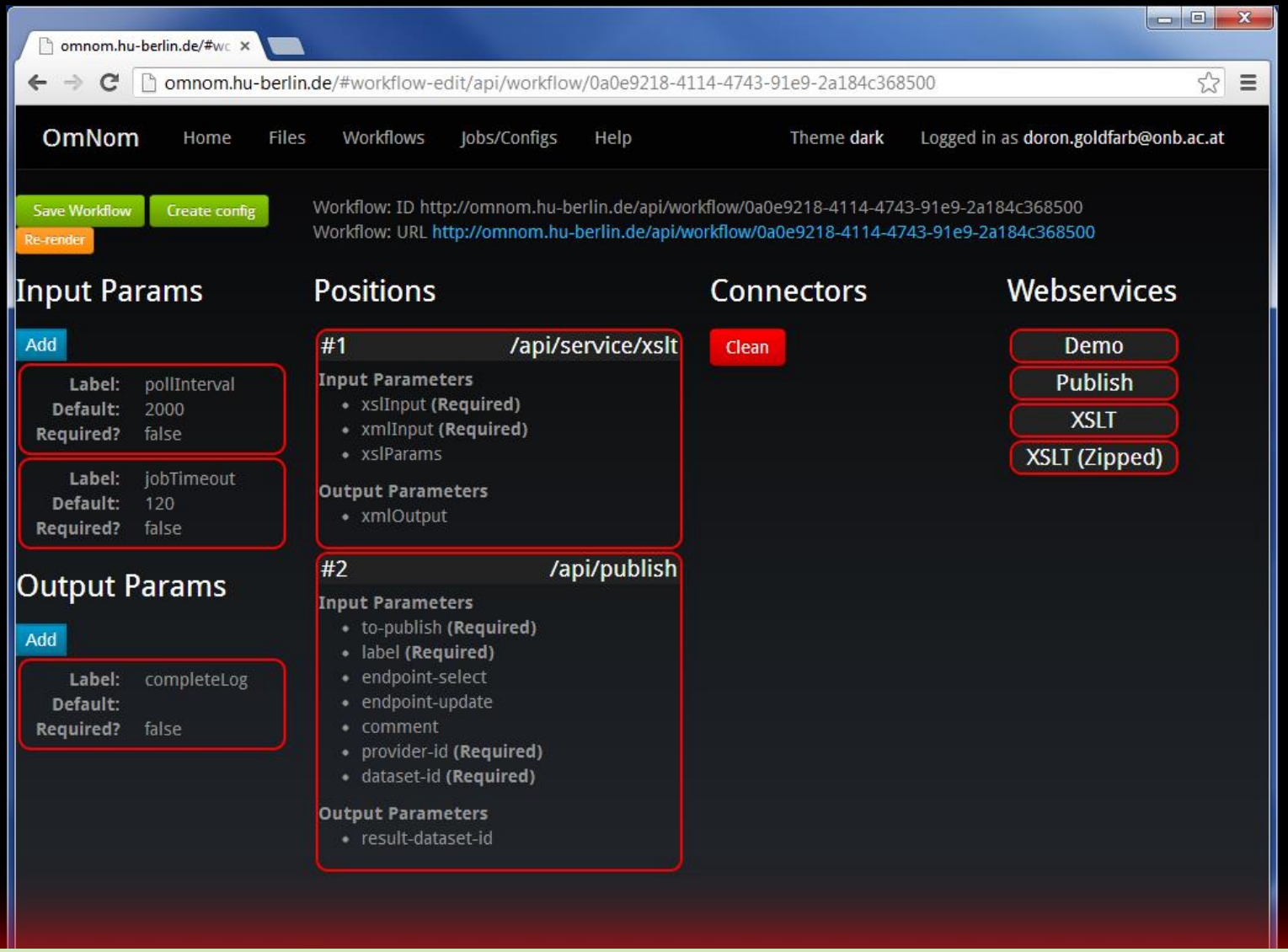
Click on the „XSLT“ webservice box and drag it to the Positions field and drop it there.



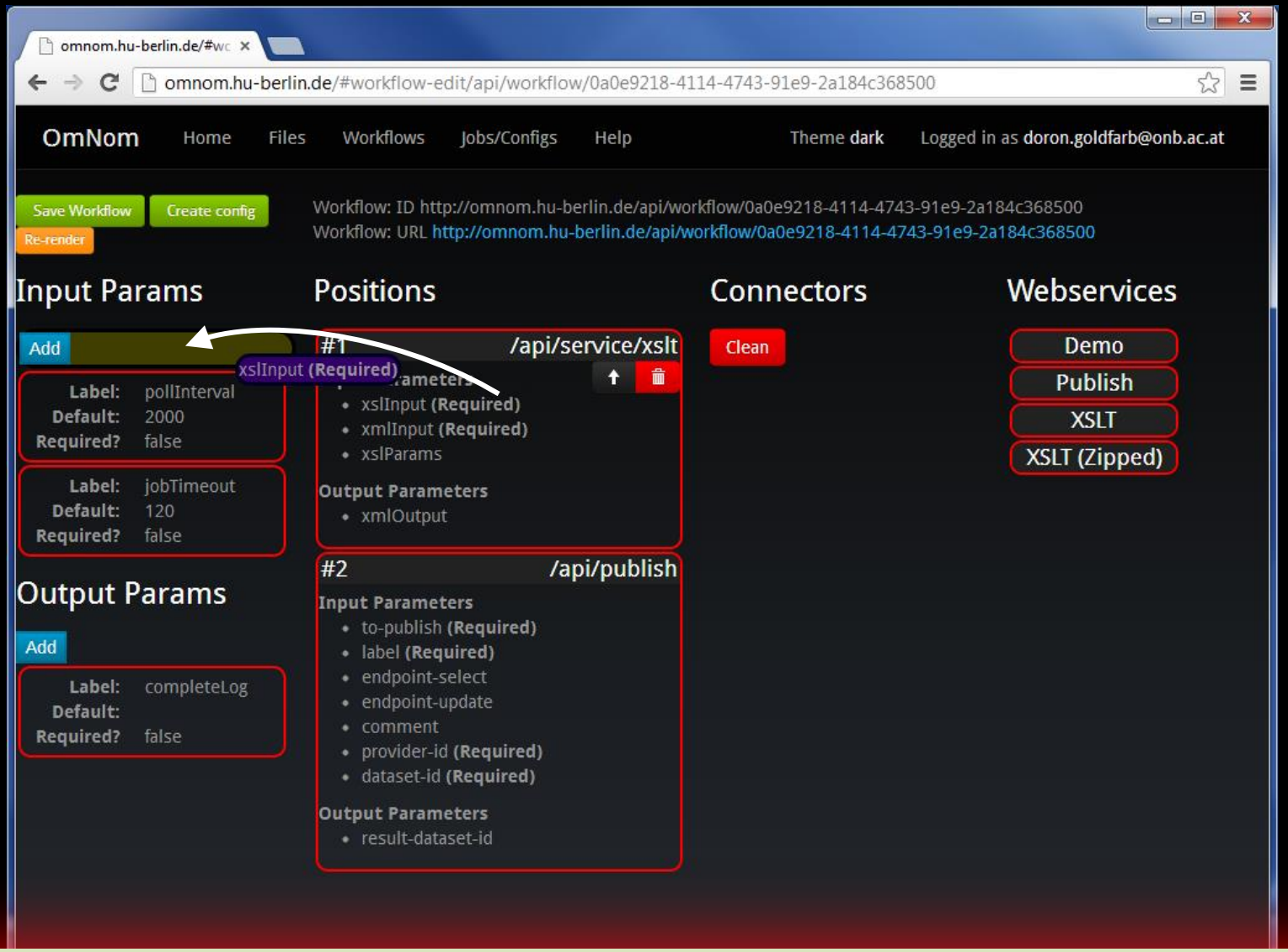
You have filled Position #1 with an XSLT webservice element. You can see a list of its input and output parameters.



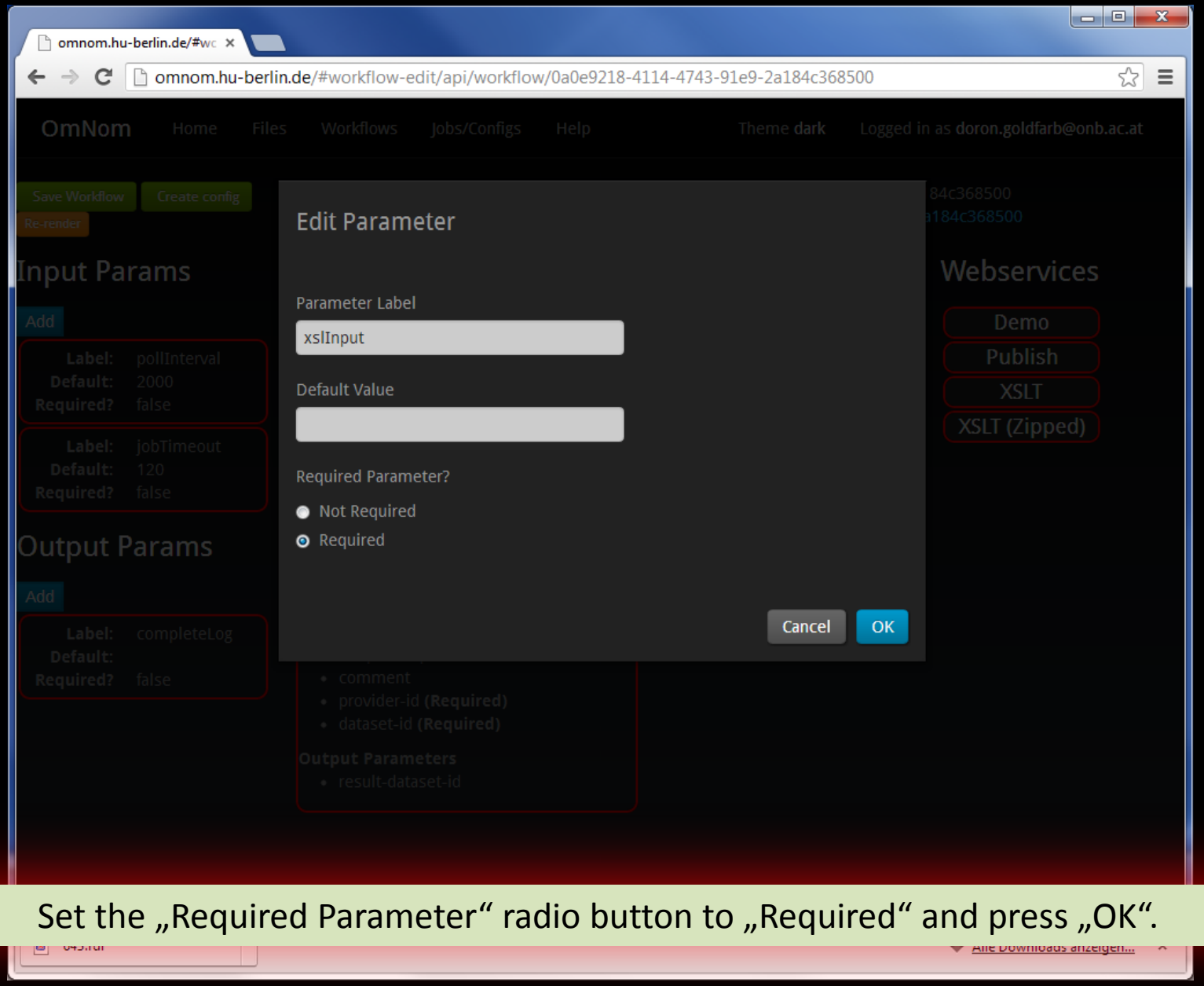
Now click, drag & drop a „Publish“ webservice object to the Positions field.



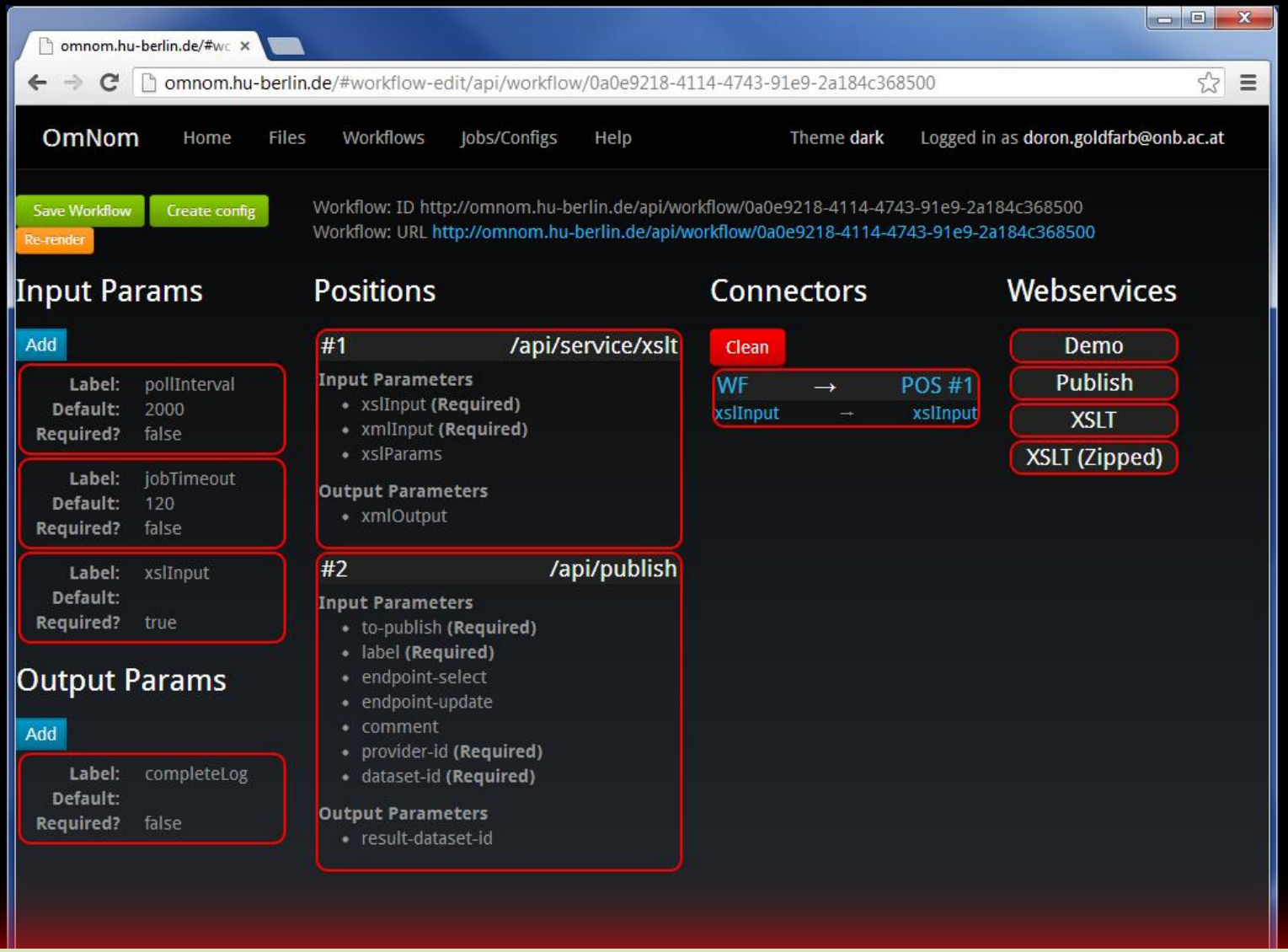
You now have defined two Positions in the workflow, #1 for the XSL transformation, #2 for publishing the results of the transformation to the DM2E Triplestore. The two elements have to be connected with their Input & Output parameters.



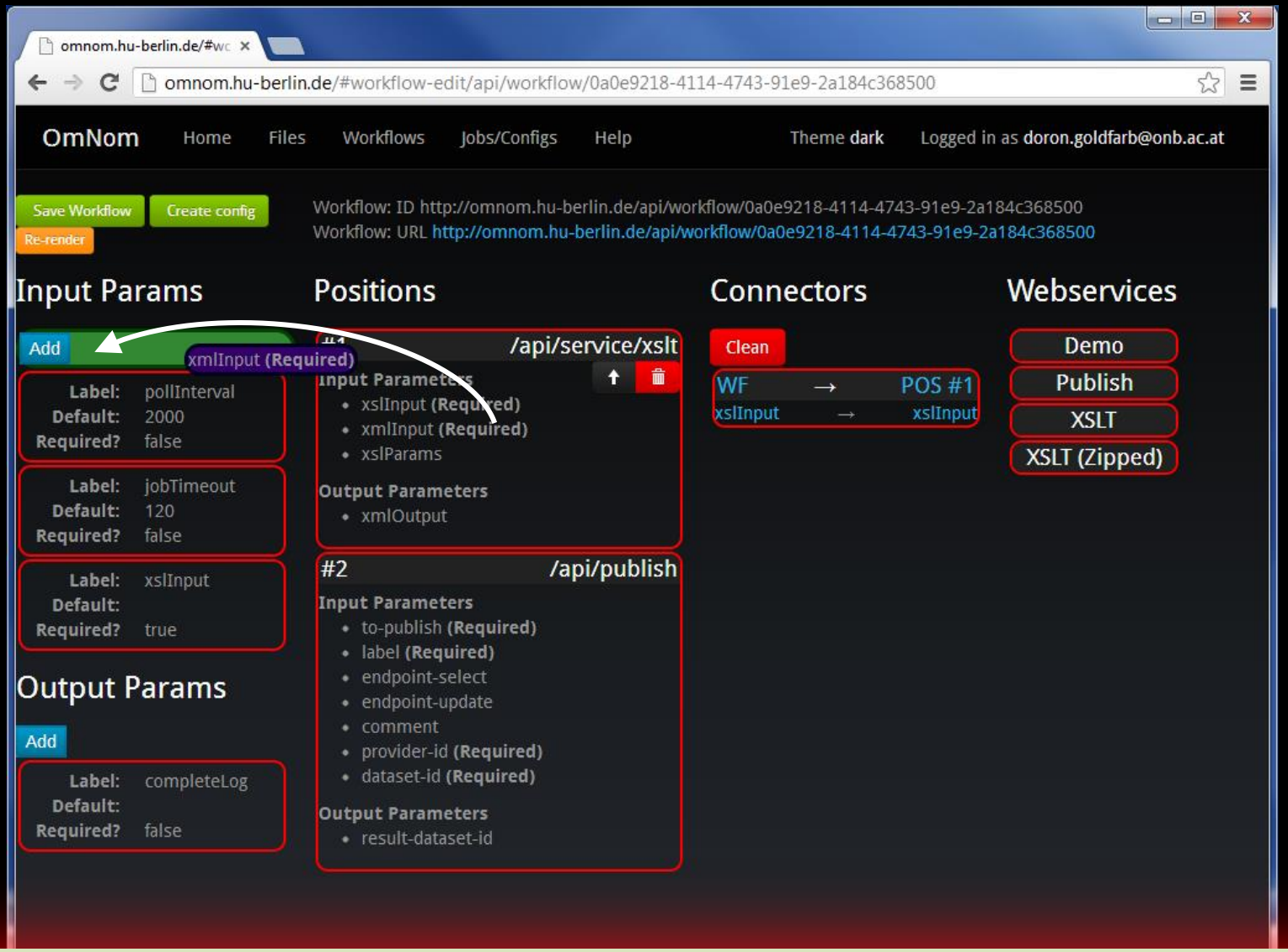
For Position #1, drag and drop the „xslInput“ parameter from its „Input parameters“ list to the „Add“ field of the „Input Params“ section.



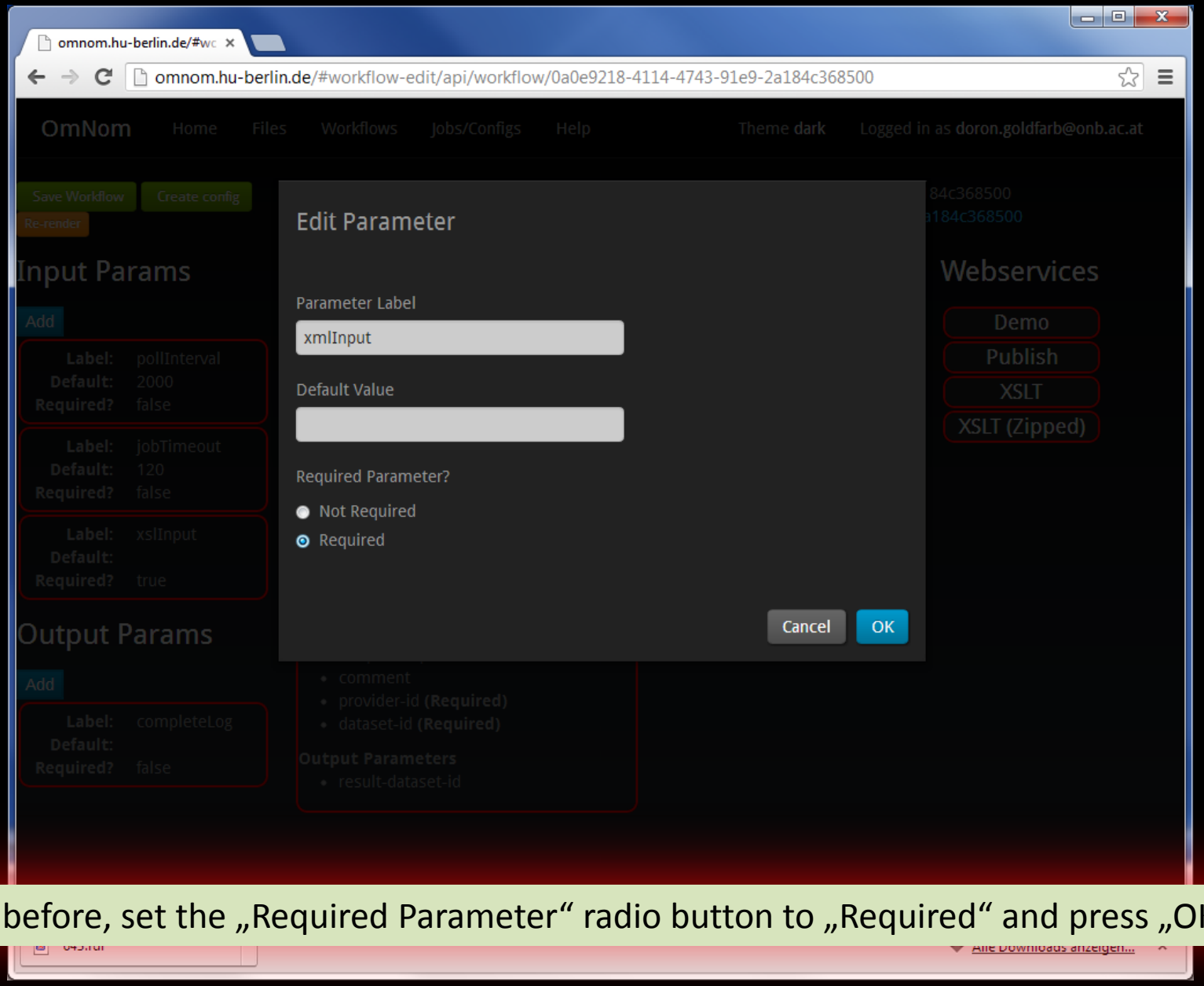
Set the „Required Parameter“ radio button to „Required“ and press „OK“.



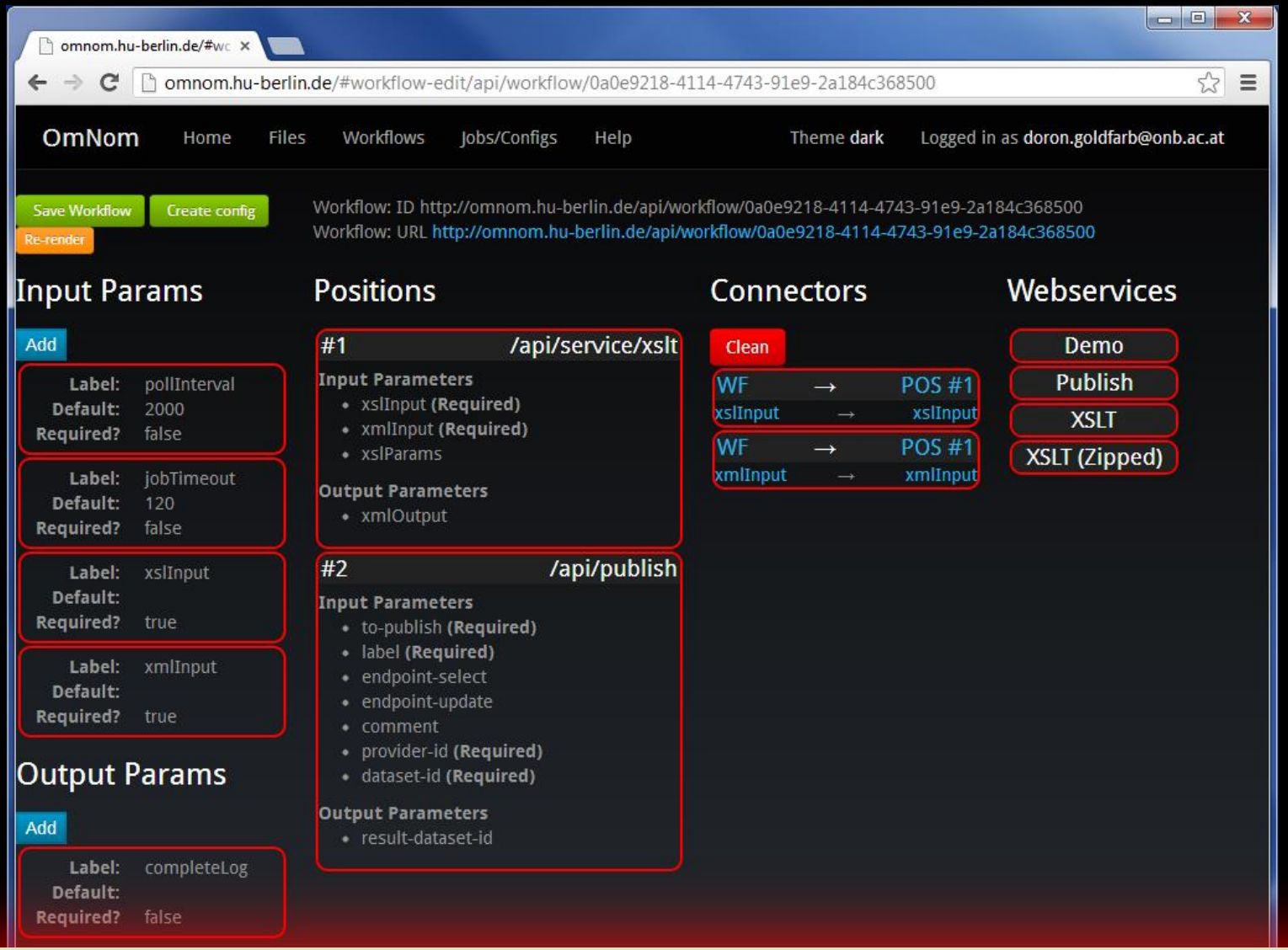
Note the newly added parameter in the „Input Params“ section on the left. Moreover, the „Connectors“ section now shows the connection between the Input parameter and the „XSLT“ webservice on Position #1.



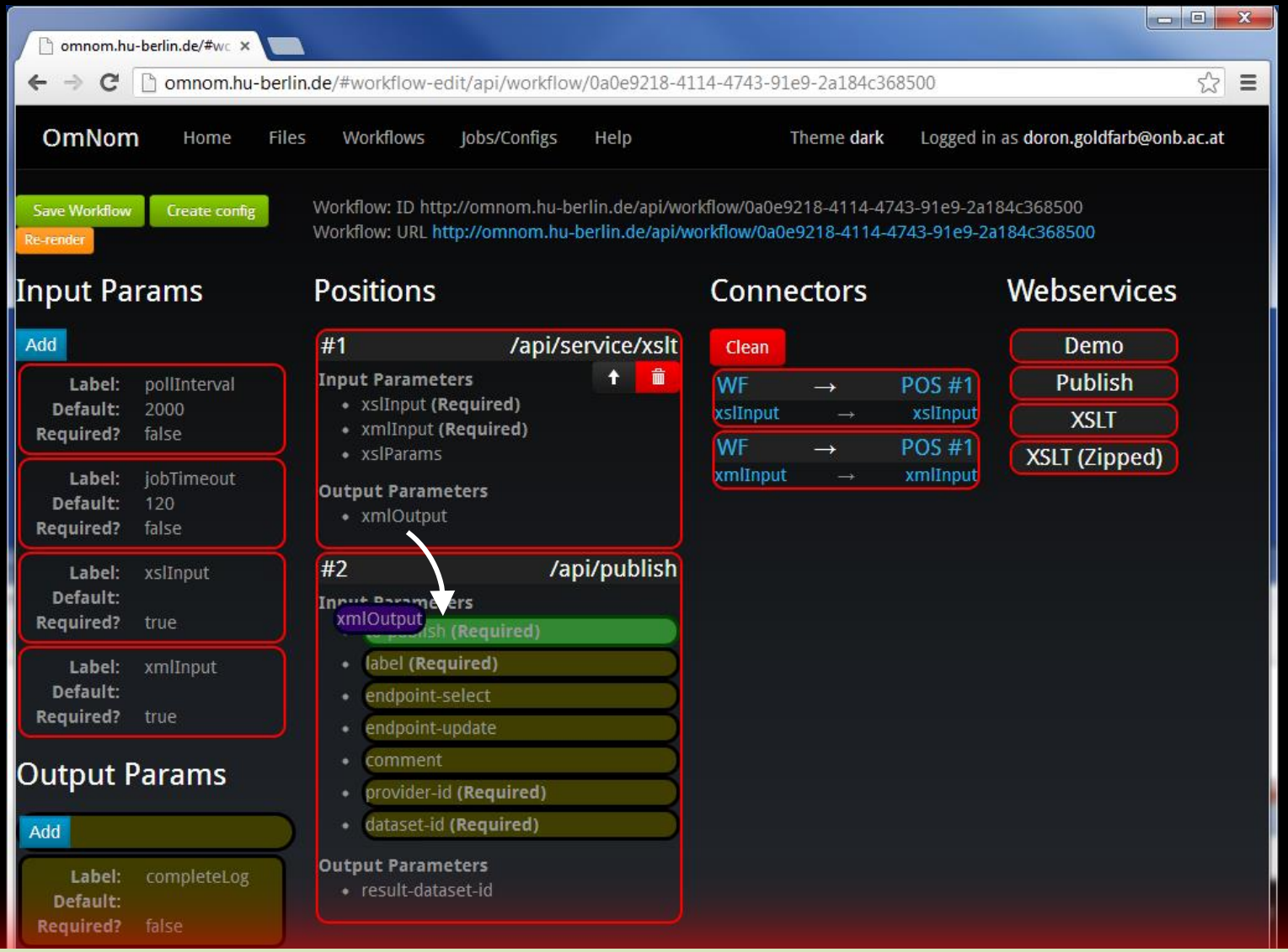
Still for Position #1, now drag and drop the „xmlInput“ parameter from #1‘s „Input parameters“ list to the „Add“ field of the „Input Params“ section.



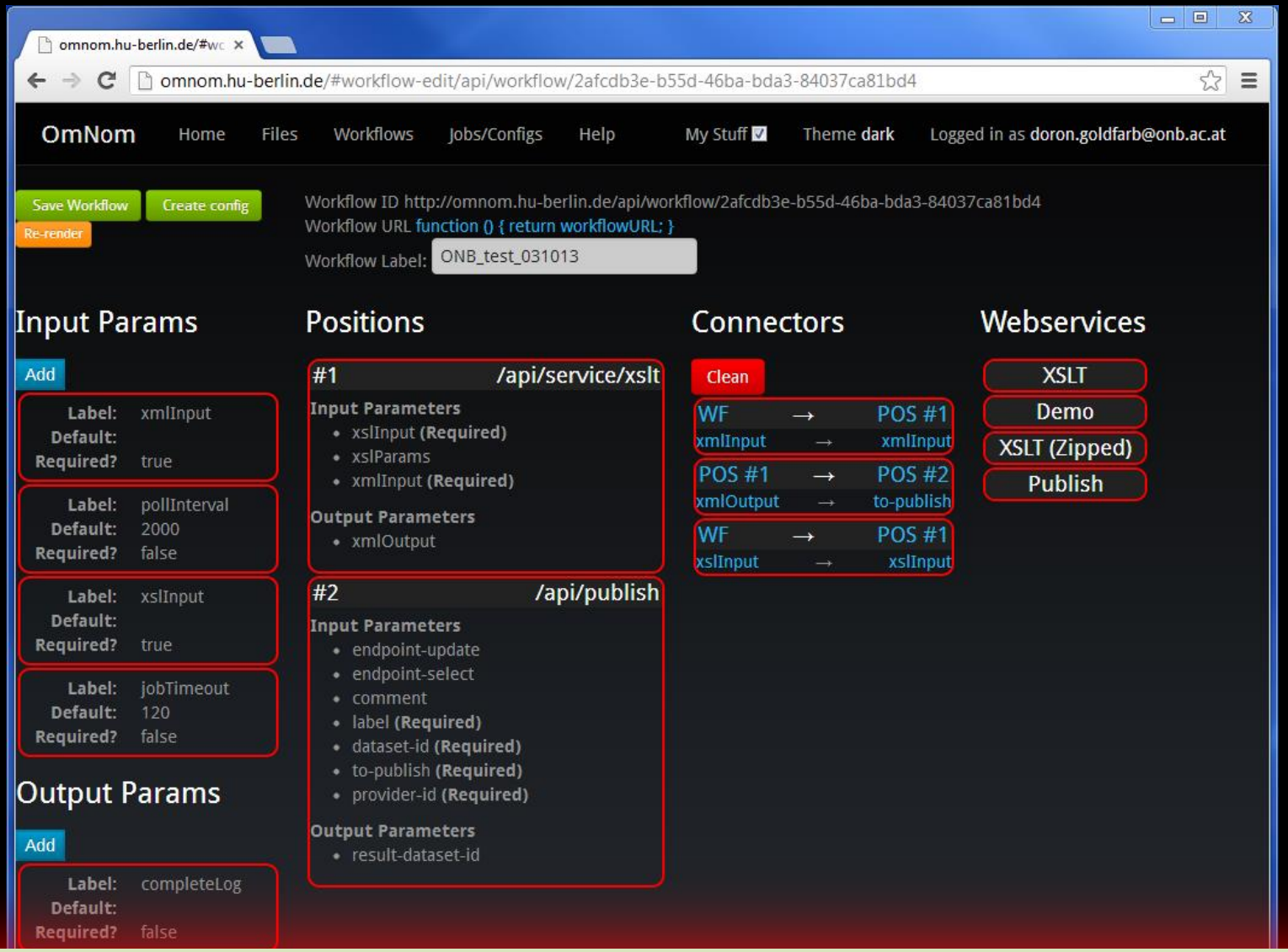
As before, set the „Required Parameter“ radio button to „Required“ and press „OK“.



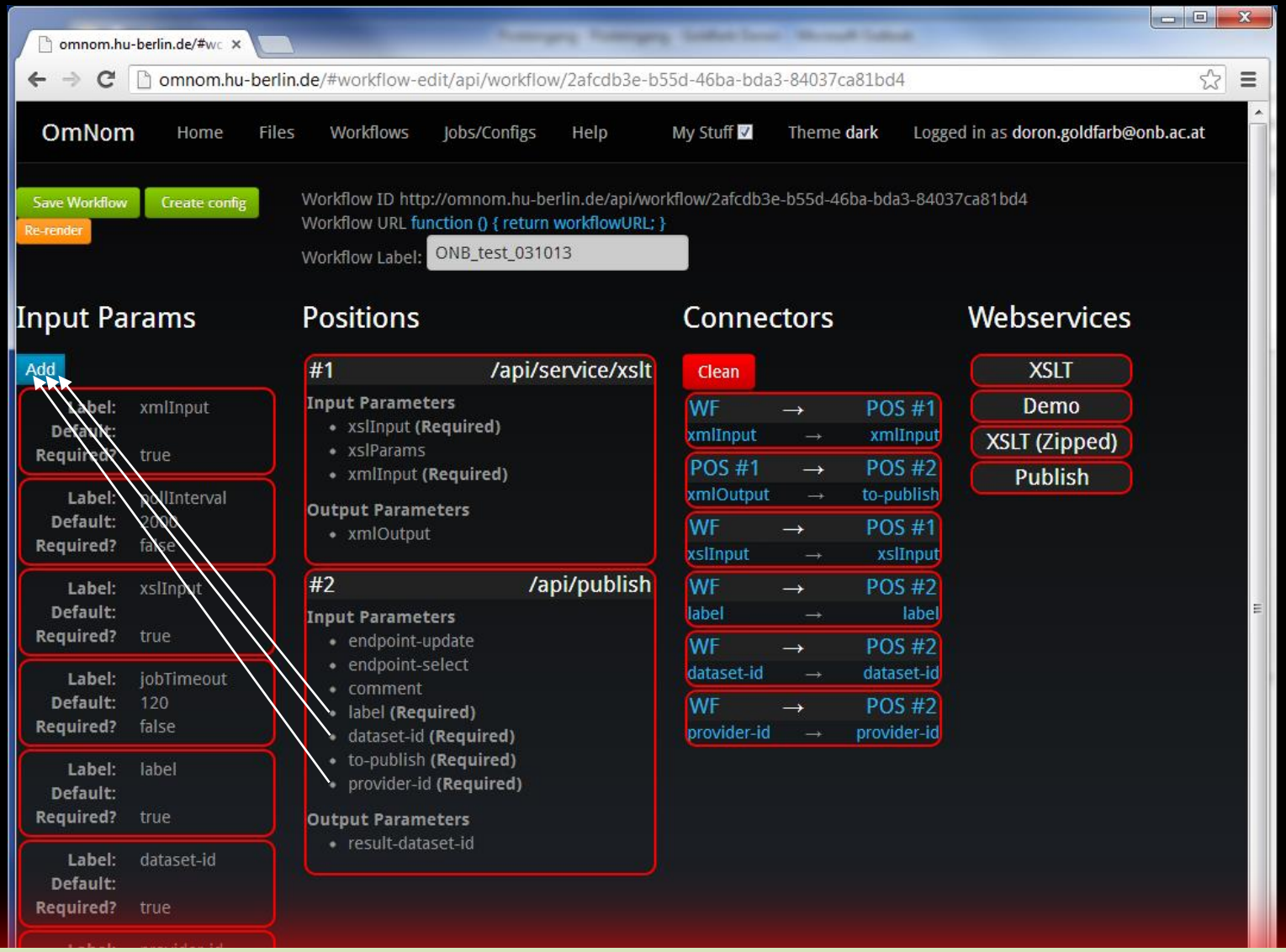
Both „xmlInput“ and „xslInput“ parameters should now be visible in the „Input Params“ section on the left. Accordingly, the „Connectors“ section now shows the both connections between the Input parameters and the „XSLT“ webservice on Position #1.



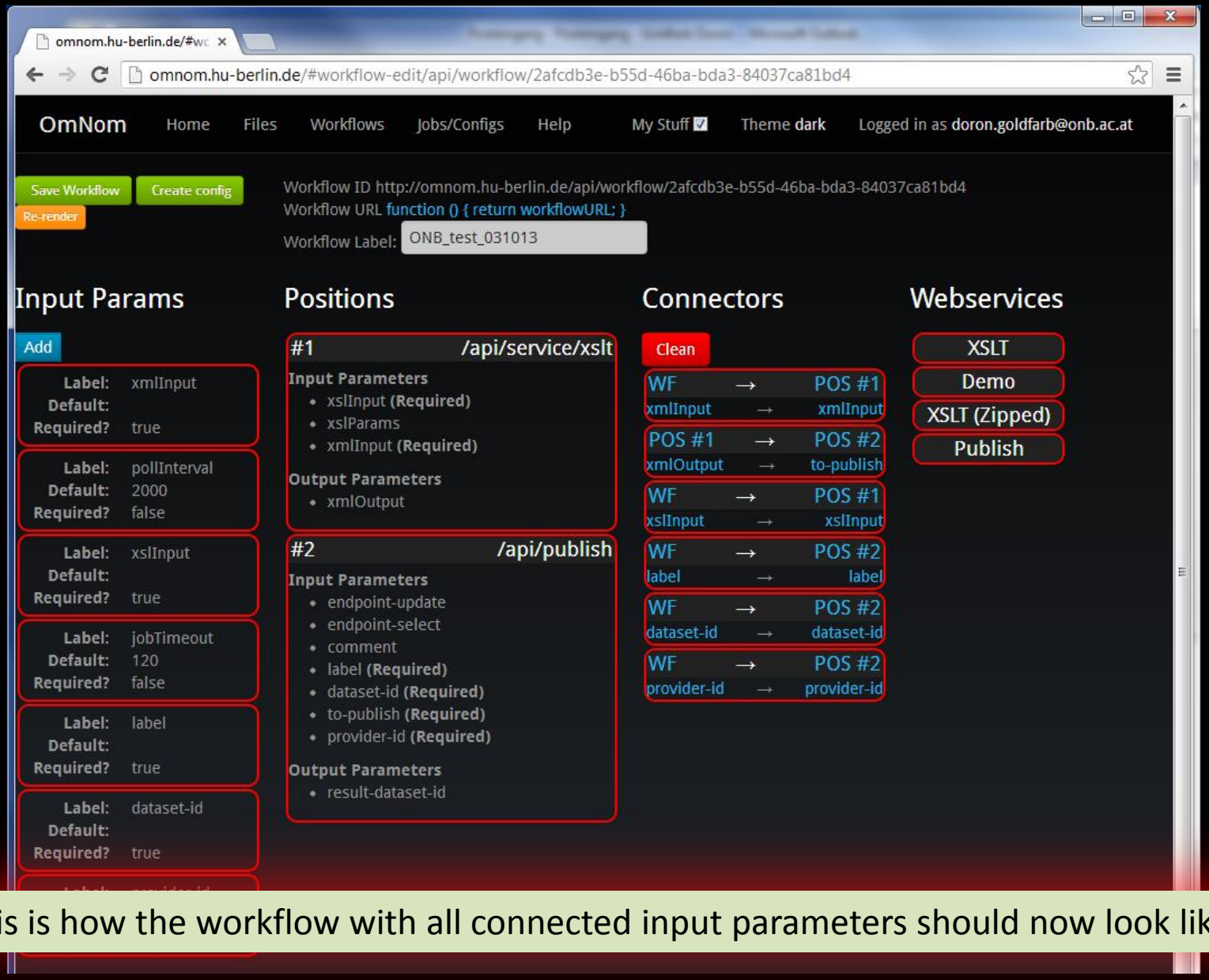
Now we connect Pos. #1 and #2 by dragging & dropping Pos #1‘s output parameter „xmlOutput“ to Pos #2‘s „to-publish“ input parameter.



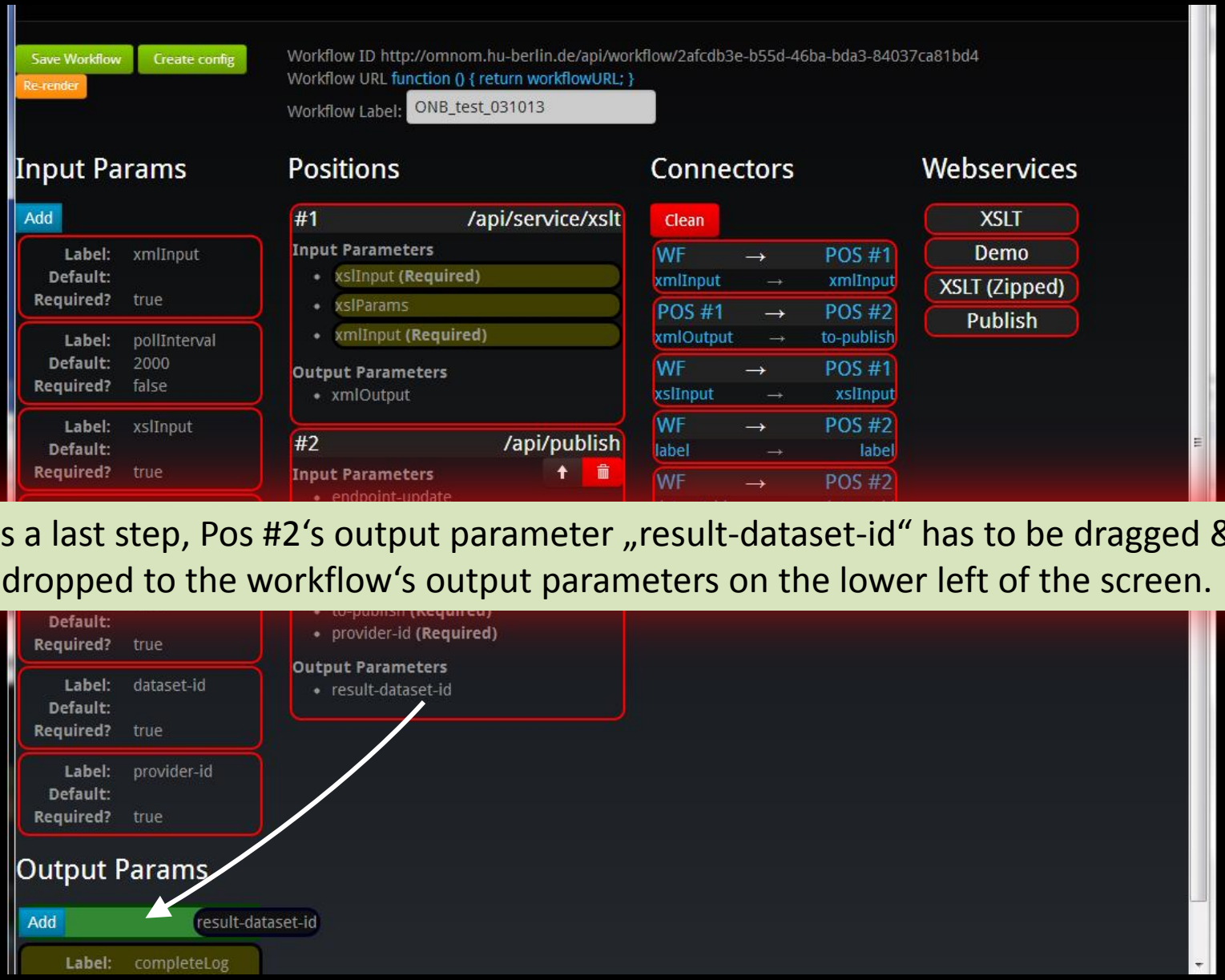
The „Connectors“ section now features a new connection between Pos #1 and Pos #2, also showing by which respective output/input parameter the connection is defined.



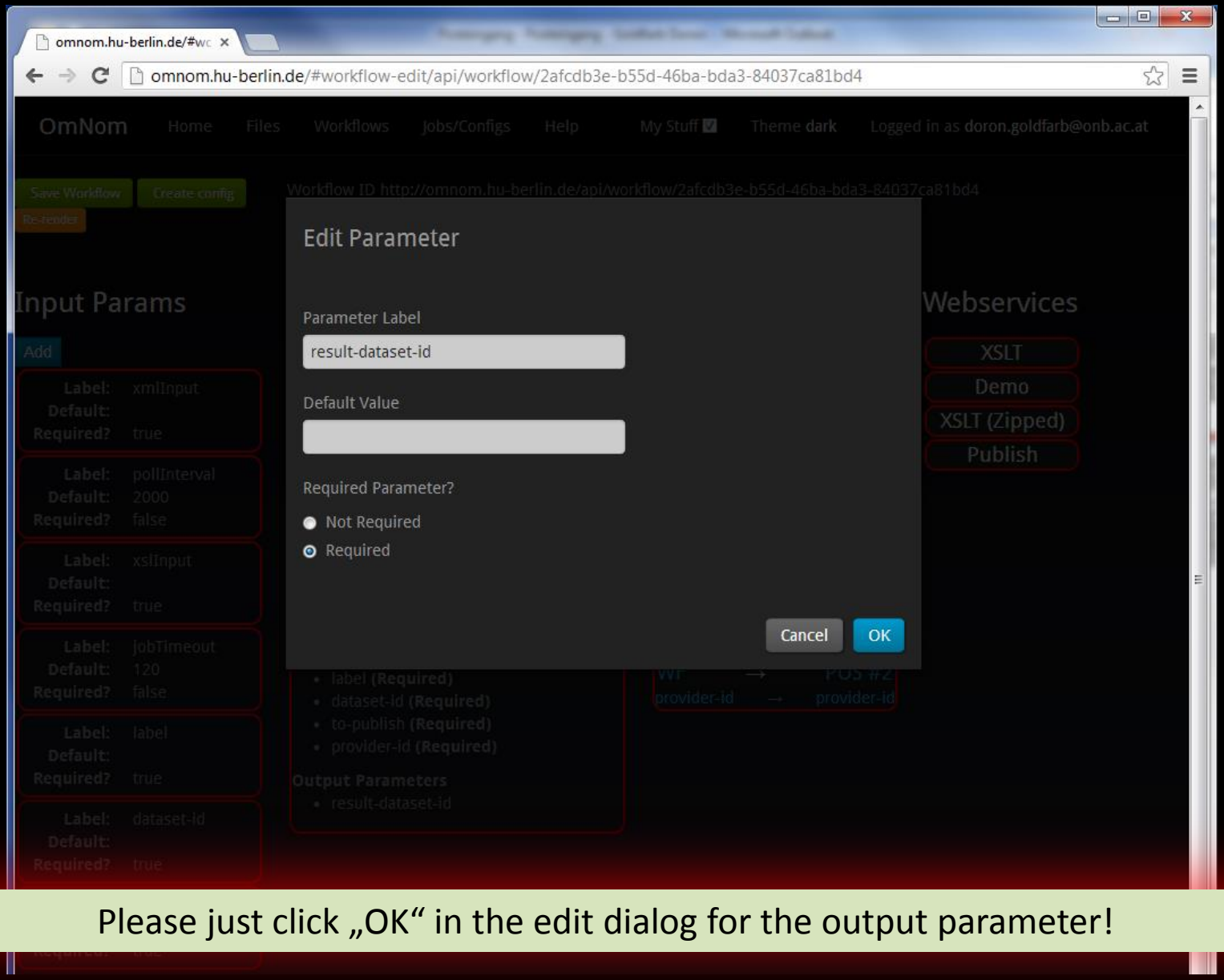
Analog to Pos #1‘s input parameters, Pos #2‘s remaining „required“ input parameters „label“, „dataset-id“ and „provider-id“ have to be added to the input parameter list on the left. Remember to set them to „Required“ in the „Edit Parameter“ dialog!



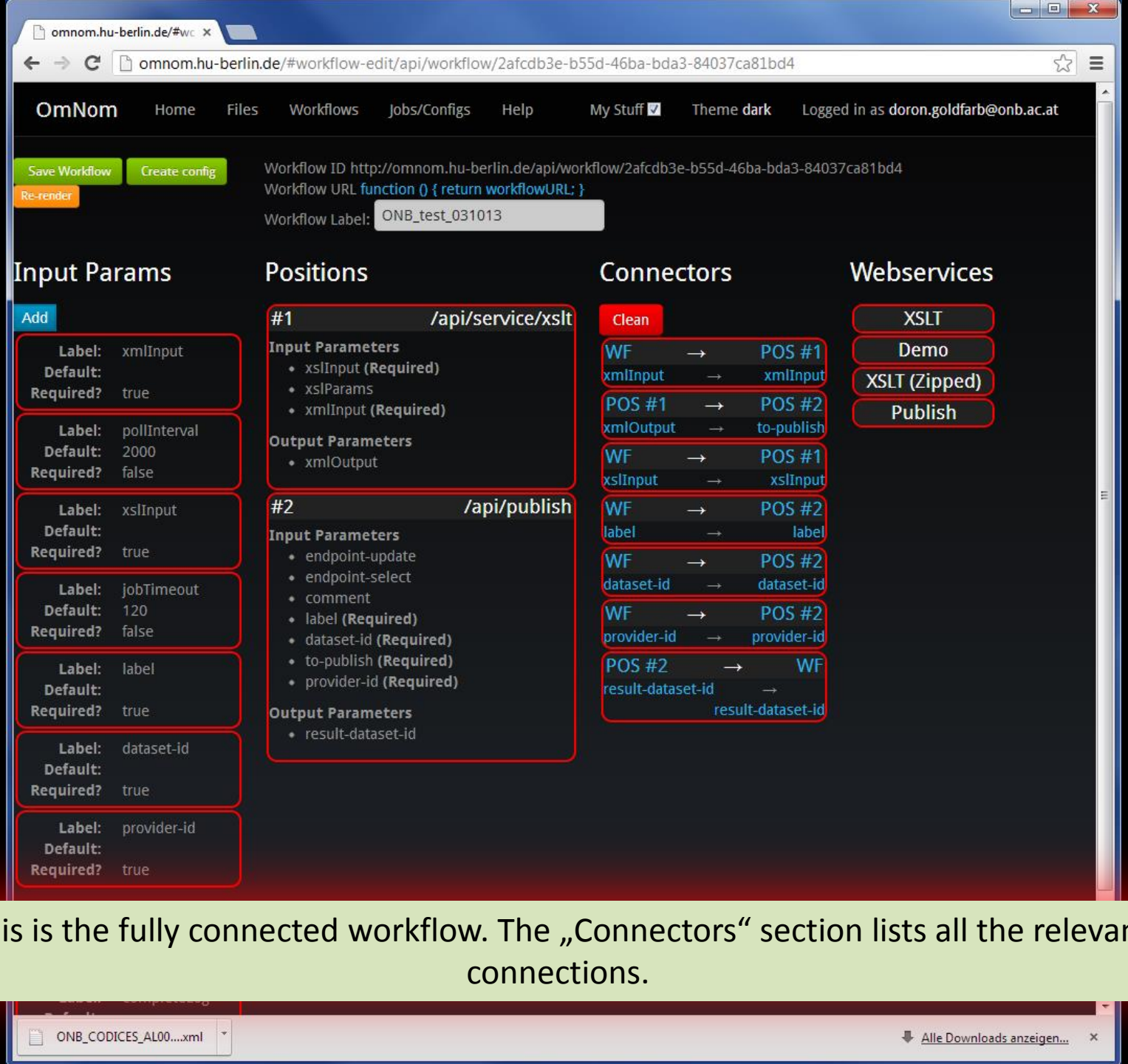
This is how the workflow with all connected input parameters should now look like.



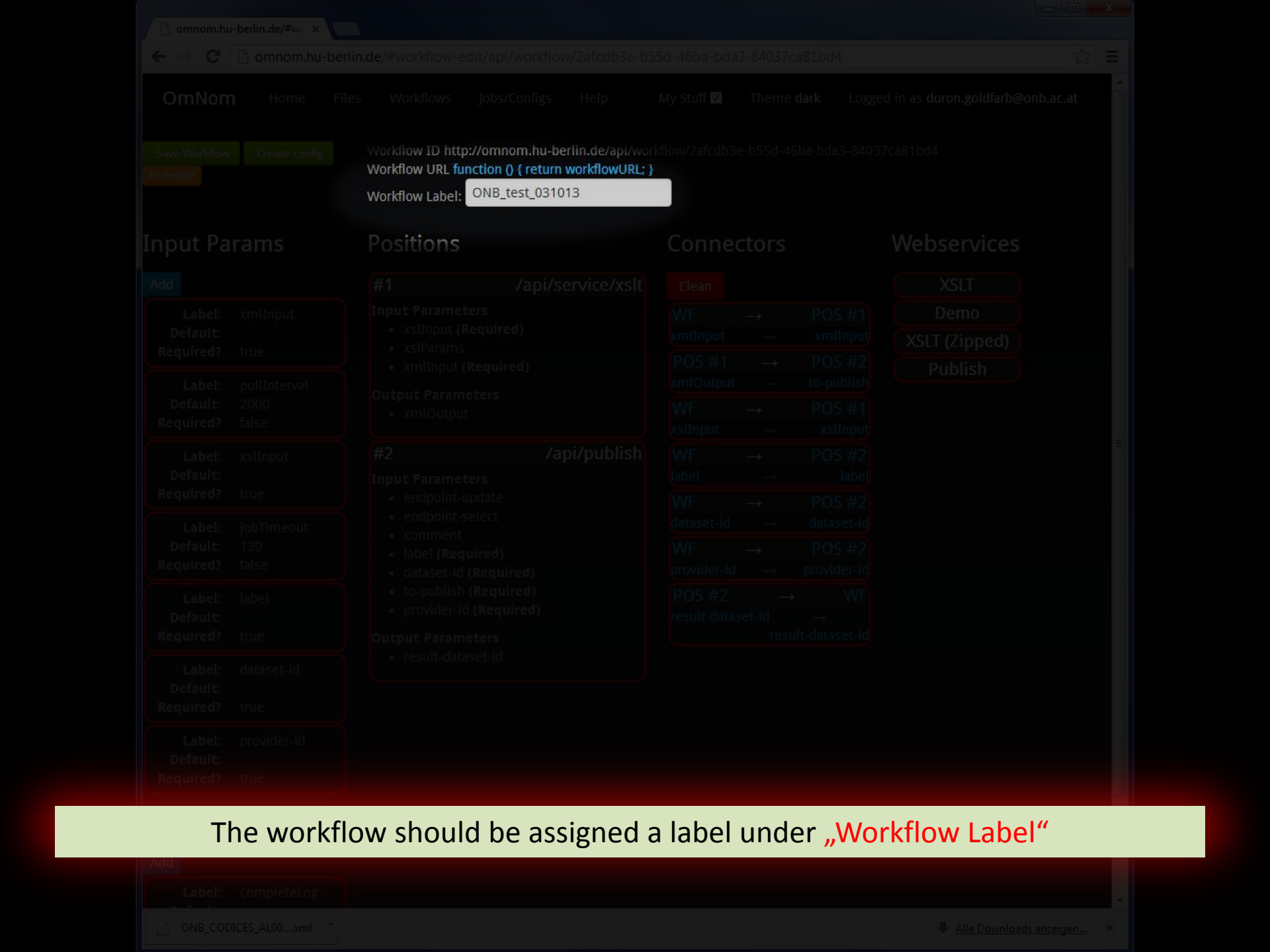
As a last step, Pos #2‘s output parameter „result-dataset-id“ has to be dragged & dropped to the workflow‘s output parameters on the lower left of the screen.



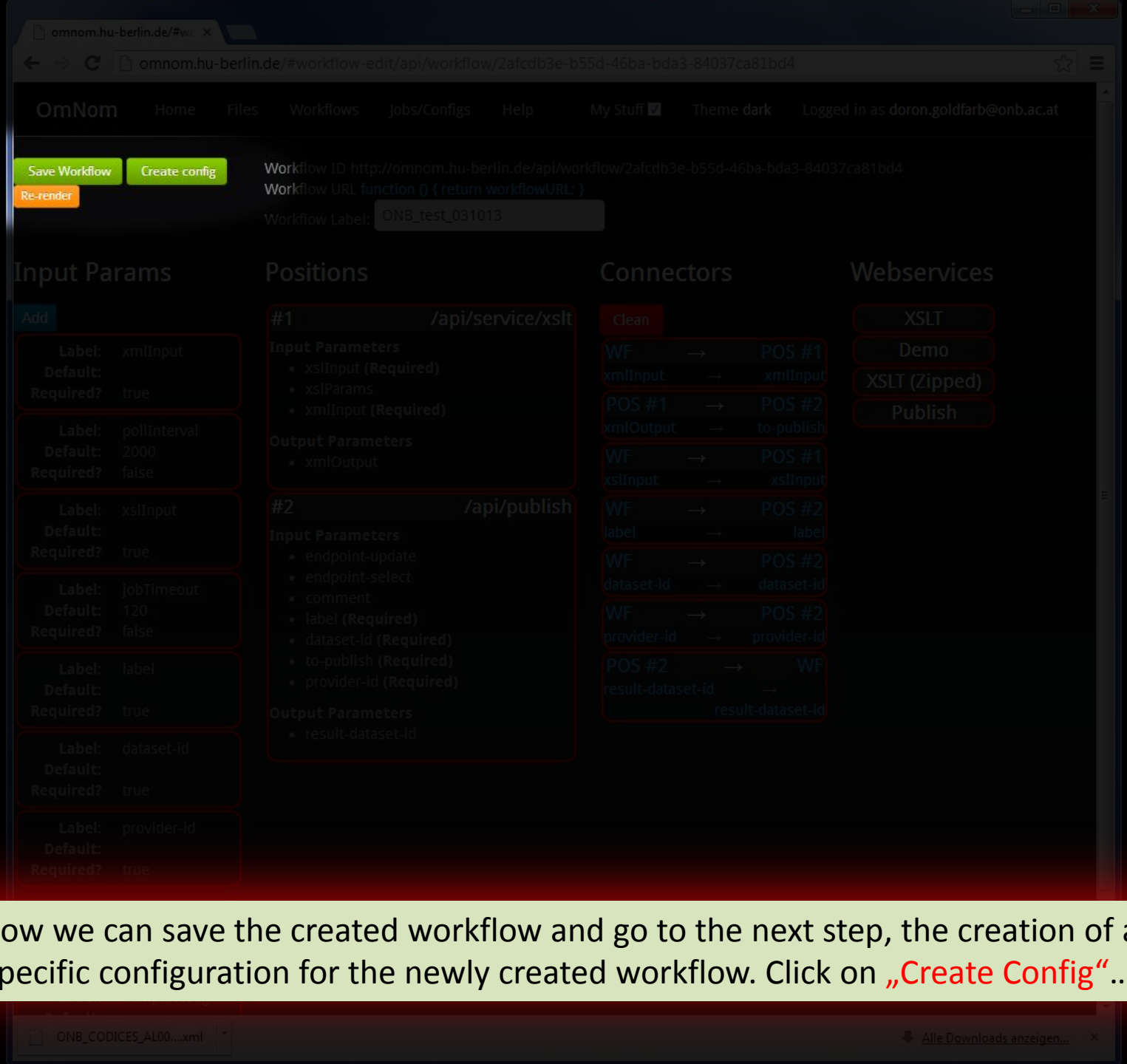
Please just click „OK“ in the edit dialog for the output parameter!



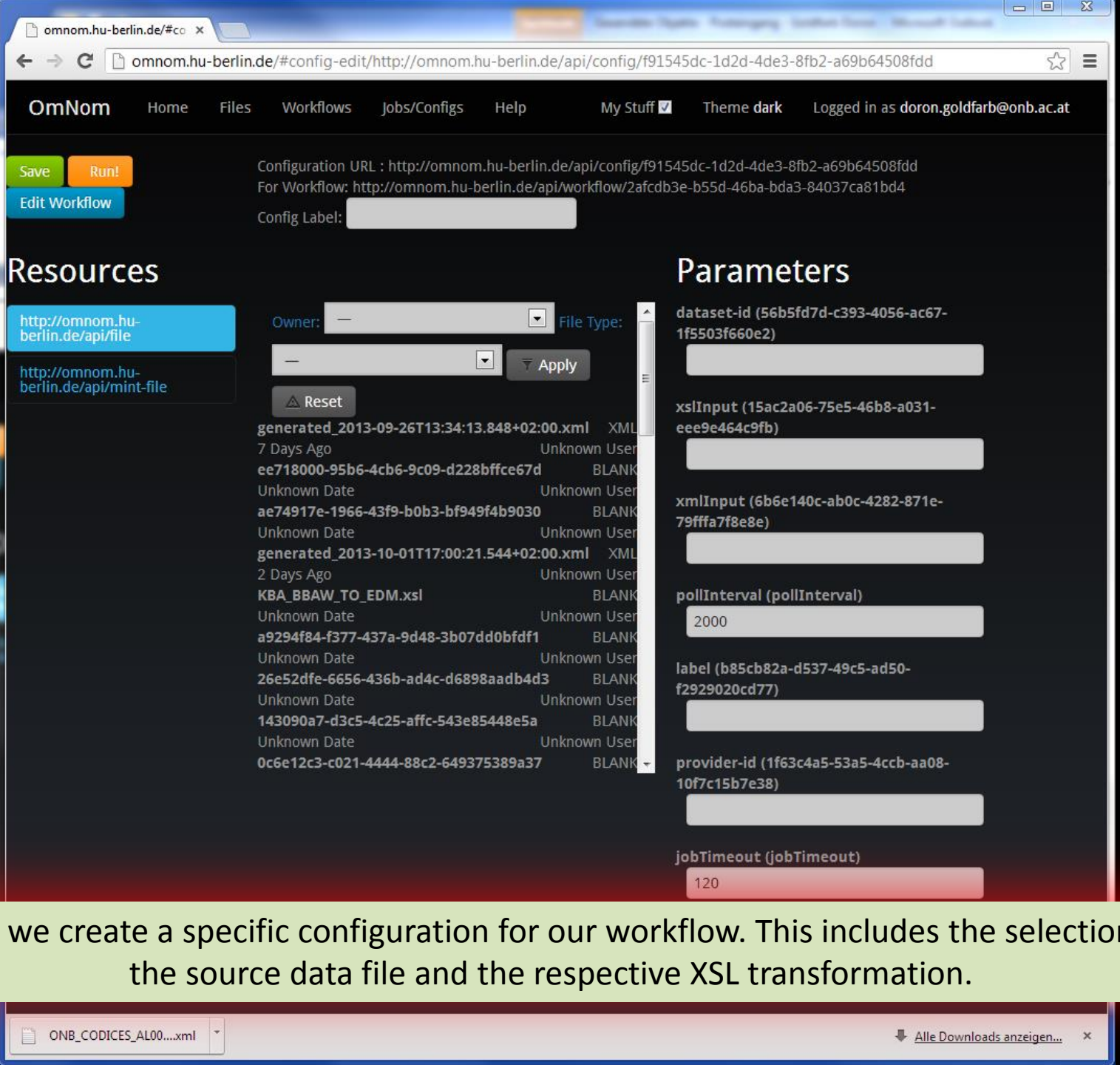
This is the fully connected workflow. The „Connectors“ section lists all the relevant connections.



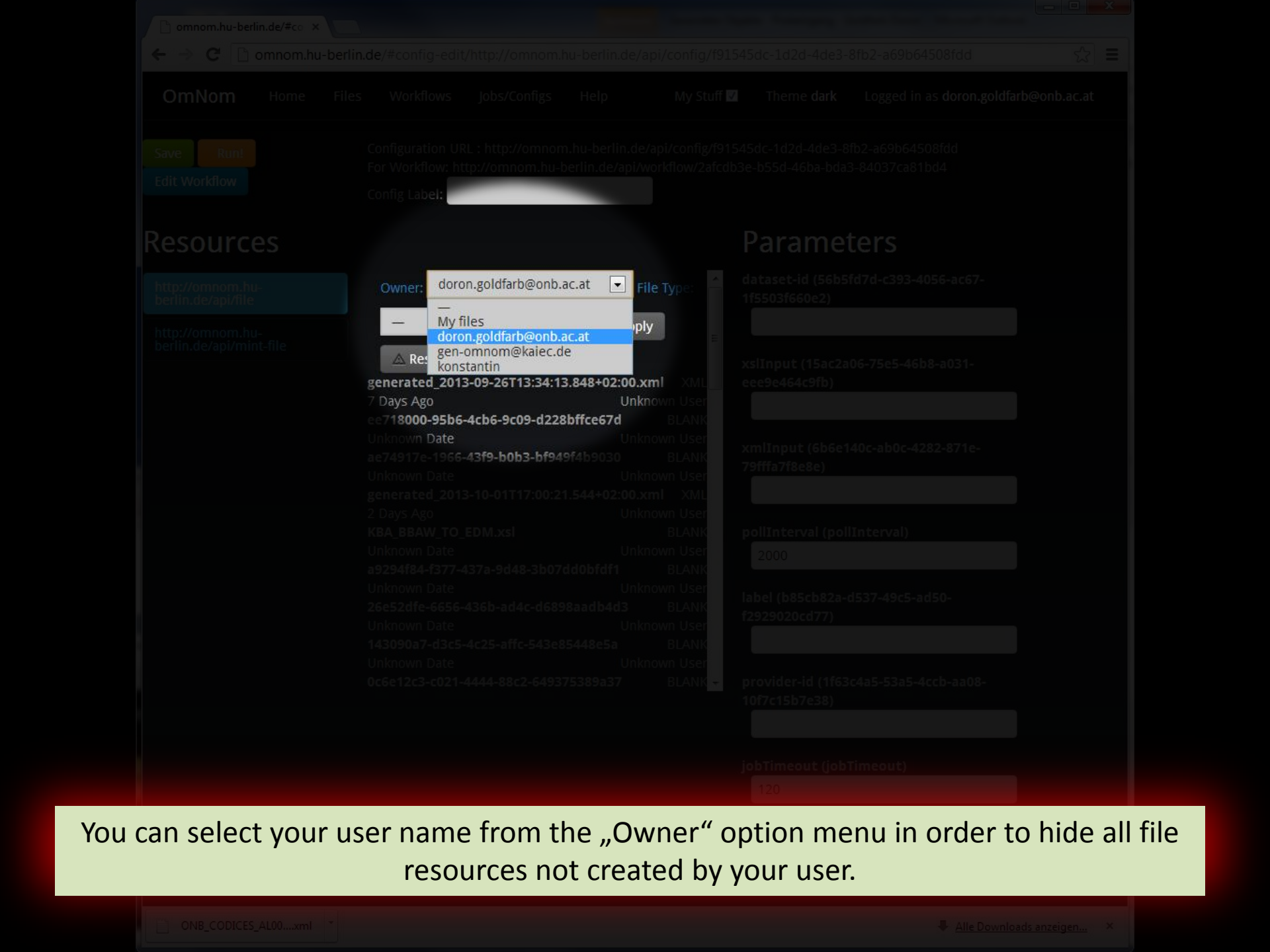
The workflow should be assigned a label under „Workflow Label“



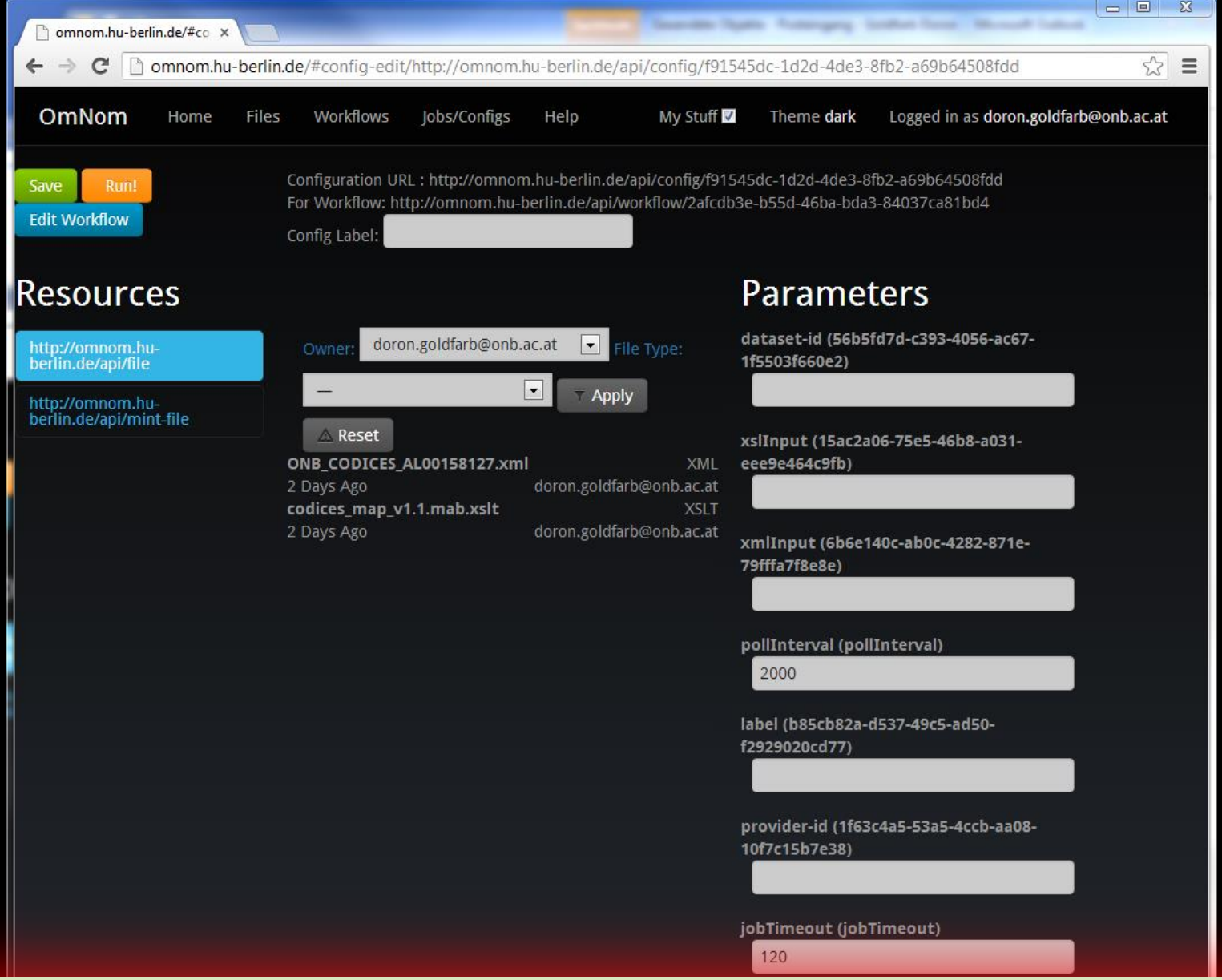
Now we can save the created workflow and go to the next step, the creation of a specific configuration for the newly created workflow. Click on „Create Config“…



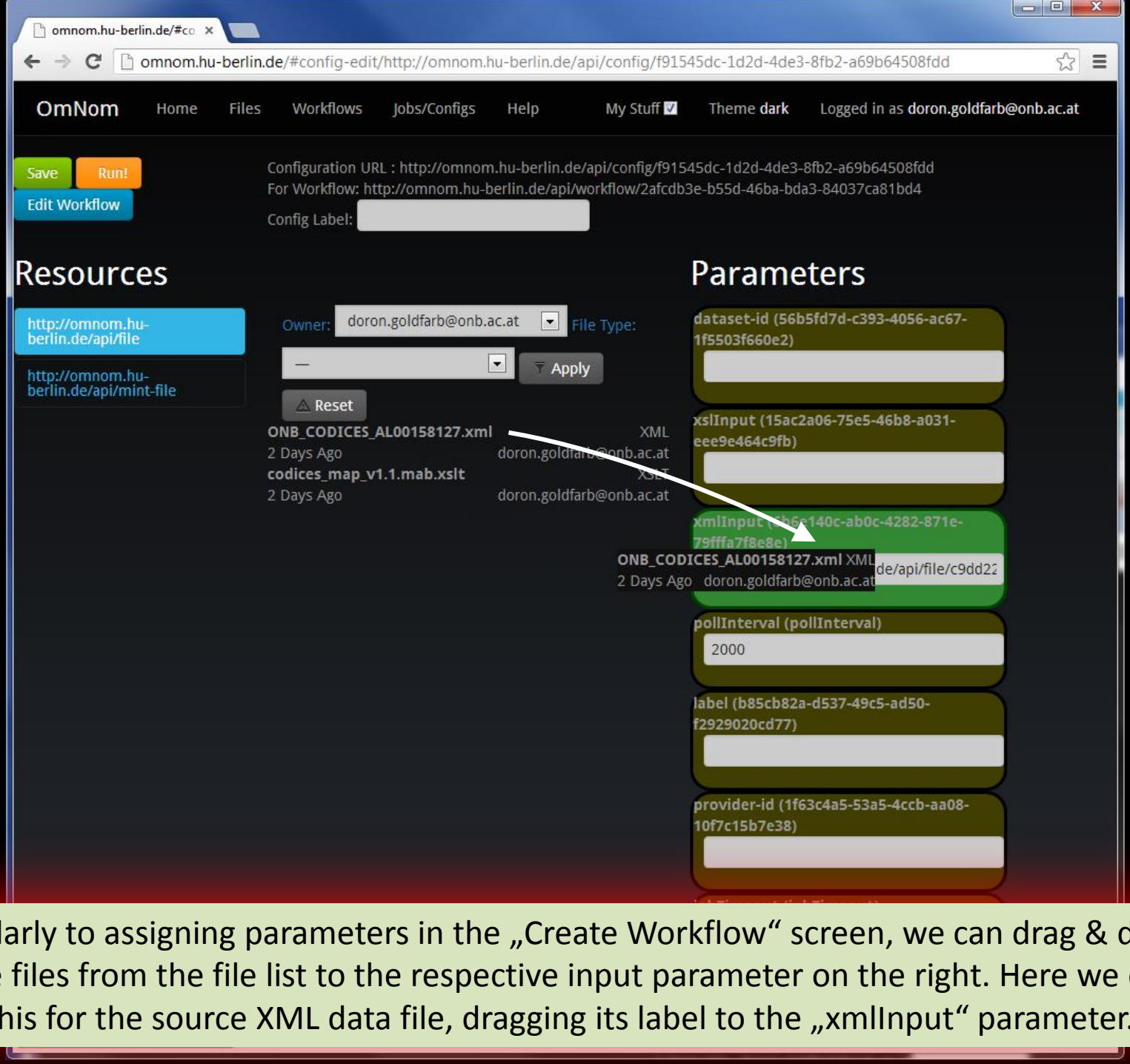
Here we create a specific configuration for our workflow. This includes the selection of the source data file and the respective XSL transformation.



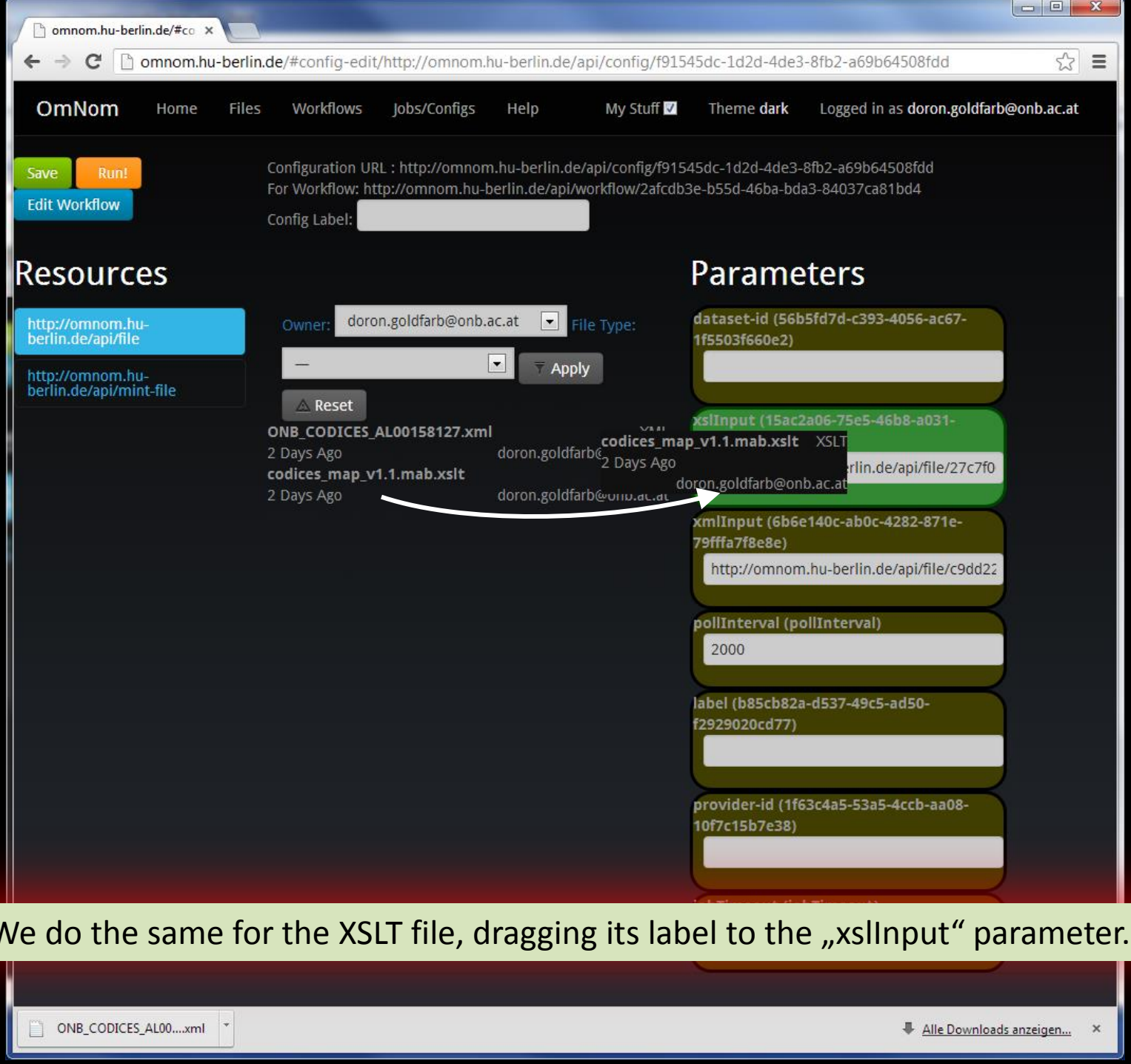
You can select your user name from the „Owner“ option menu in order to hide all file resources not created by your user.



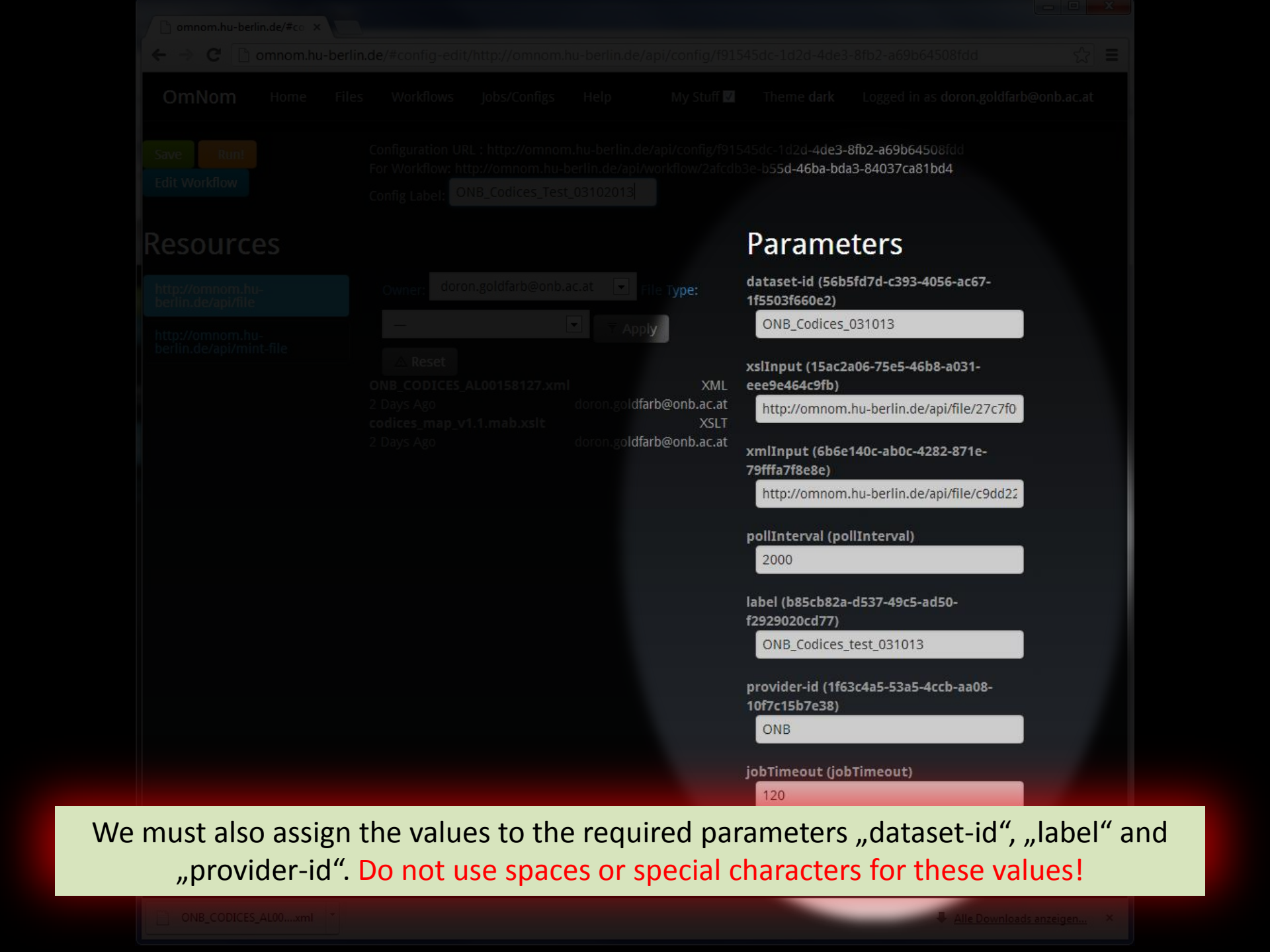
The filtered list contains all resources uploaded by our user name. For this configuration, we need the source XML data file and the appropriate XSL transformation.



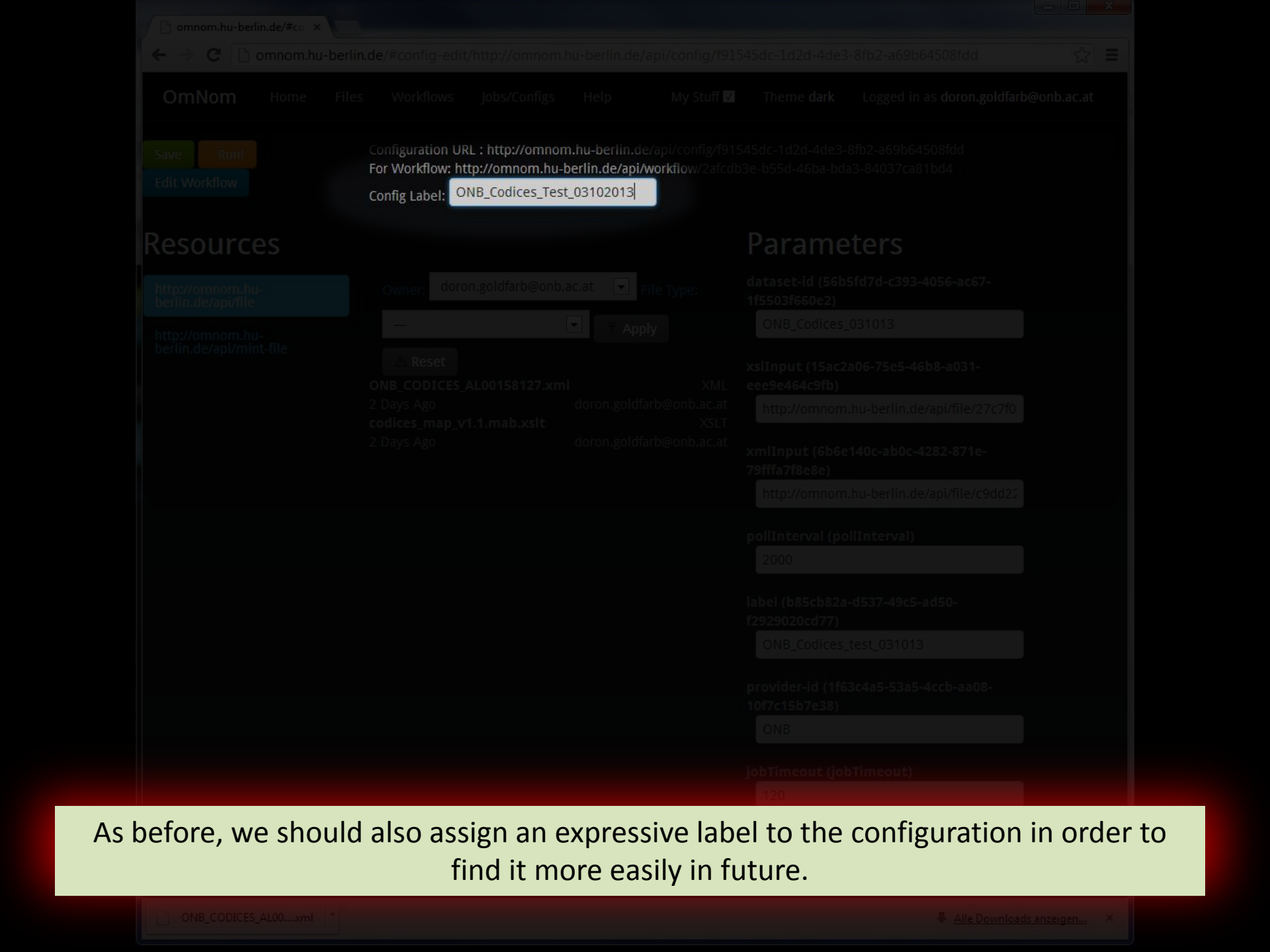
Similarly to assigning parameters in the „Create Workflow“ screen, we can drag & drop the files from the file list to the respective input parameter on the right. Here we do this for the source XML data file, dragging its label to the „xmlInput“ parameter.



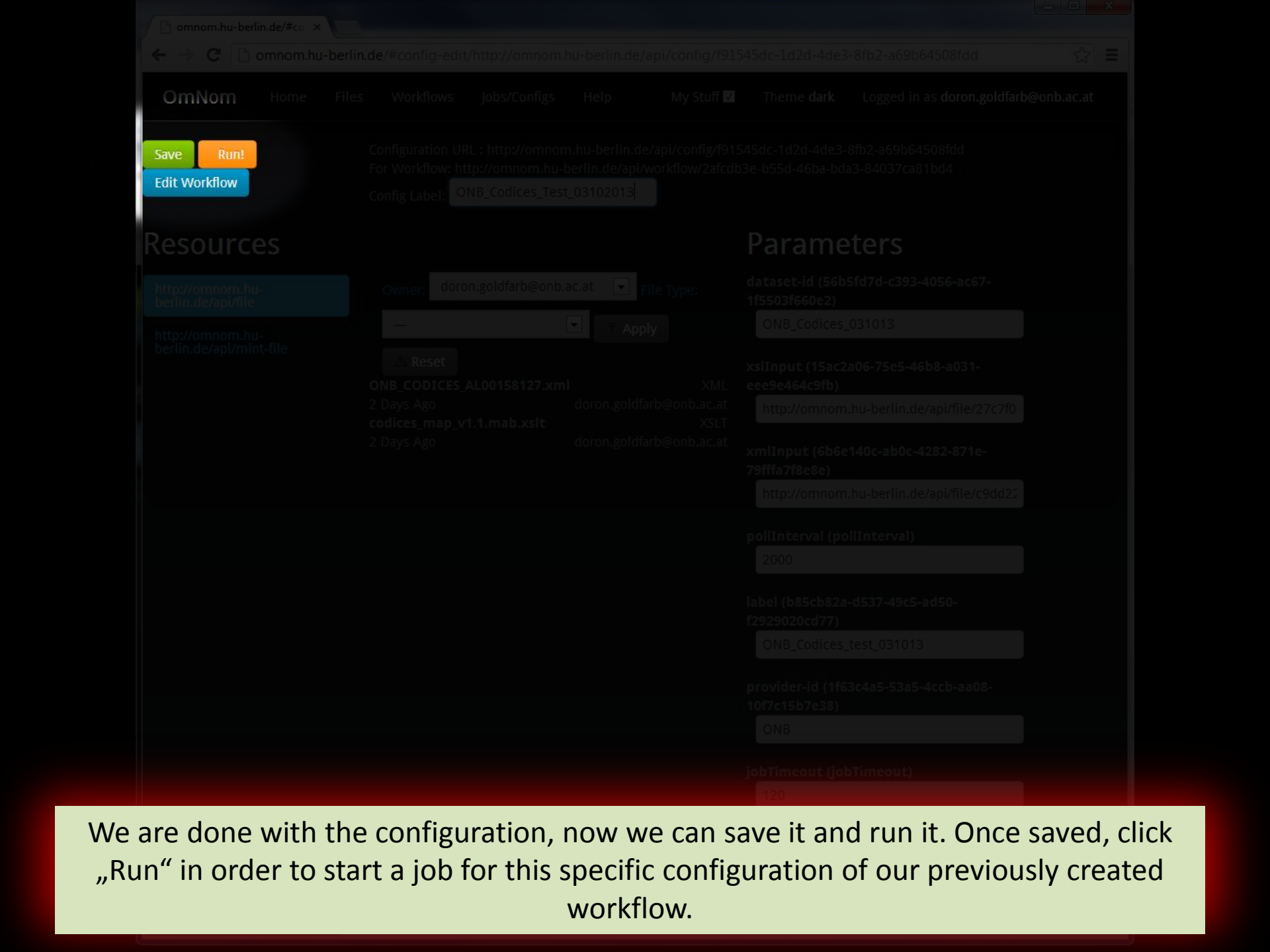
We do the same for the XSLT file, dragging its label to the „xslInput“ parameter.



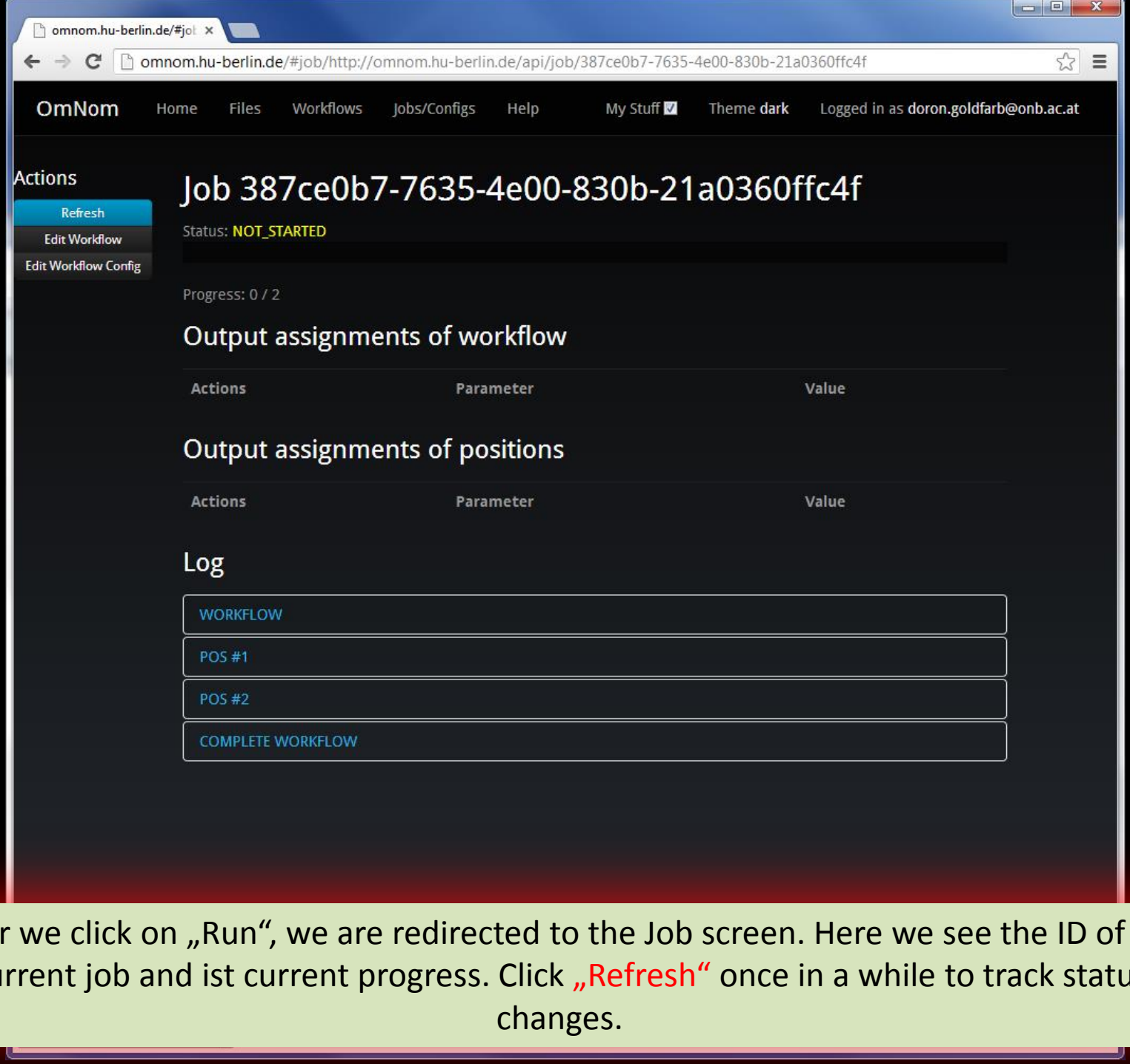
We must also assign the values to the required parameters „dataset-id“, „label“ and „provider-id“. Do not use spaces or special characters for these values!



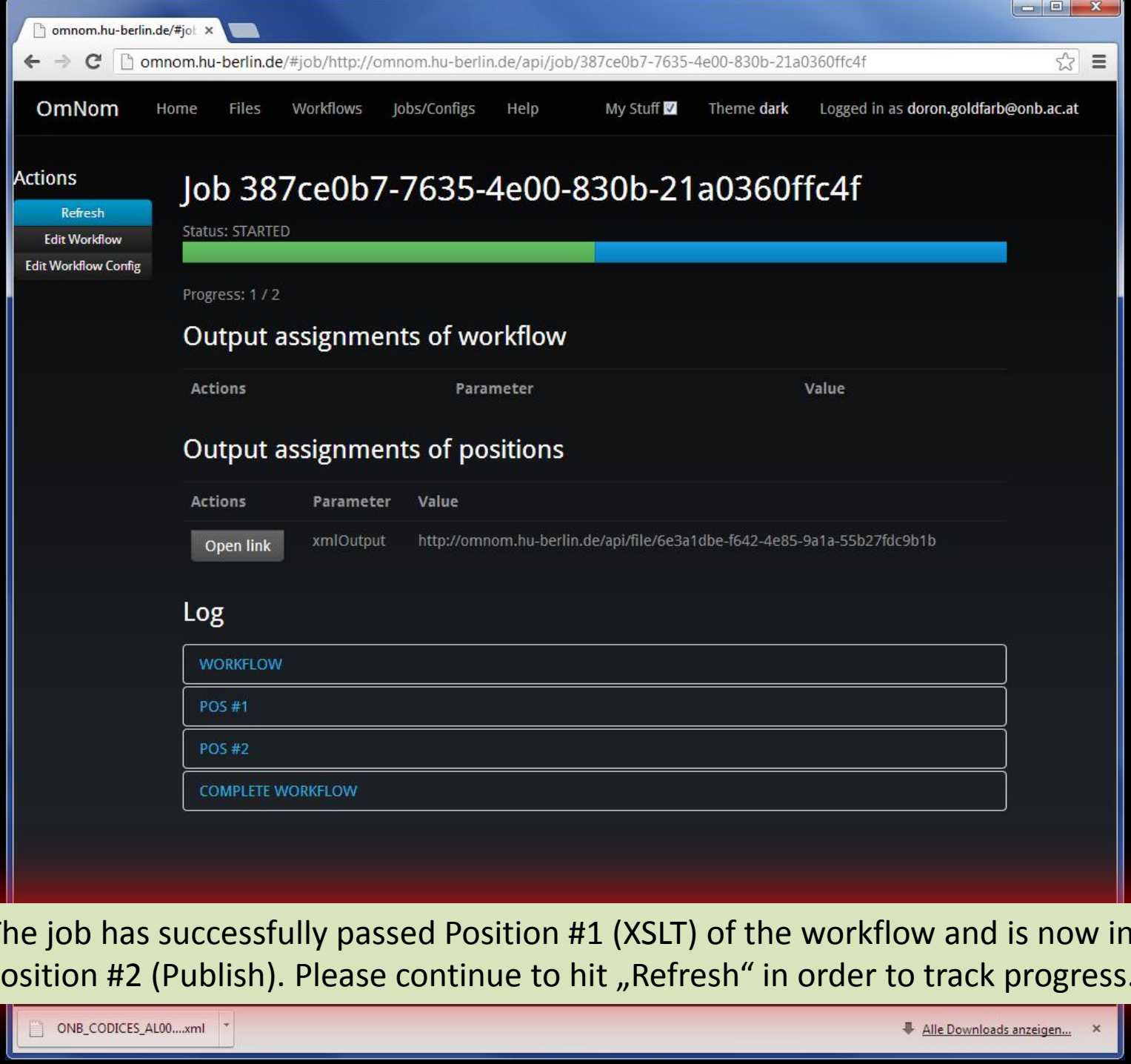
As before, we should also assign an expressive label to the configuration in order to find it more easily in future.



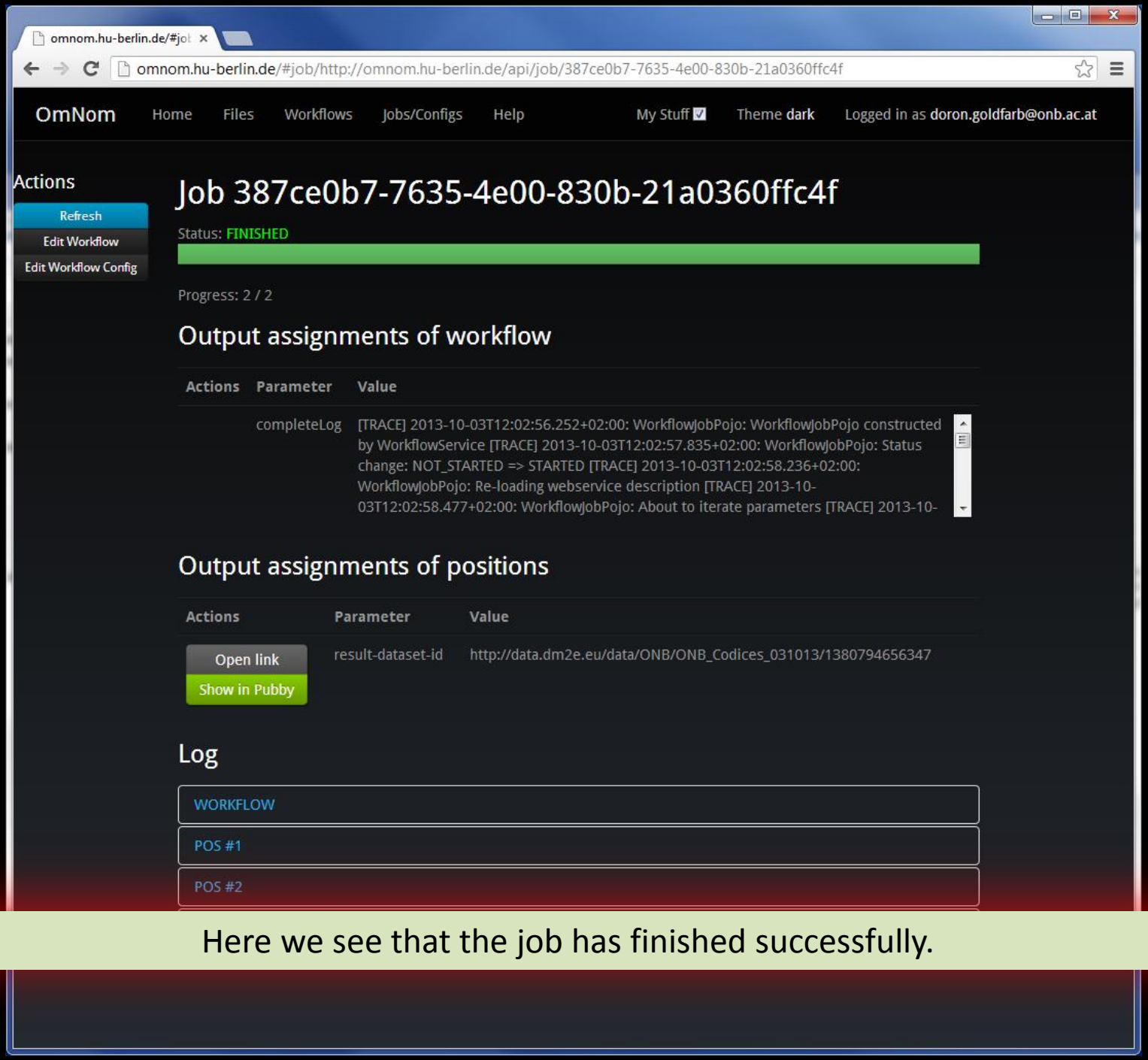
We are done with the configuration, now we can save it and run it. Once saved, click „Run“ in order to start a job for this specific configuration of our previously created workflow.



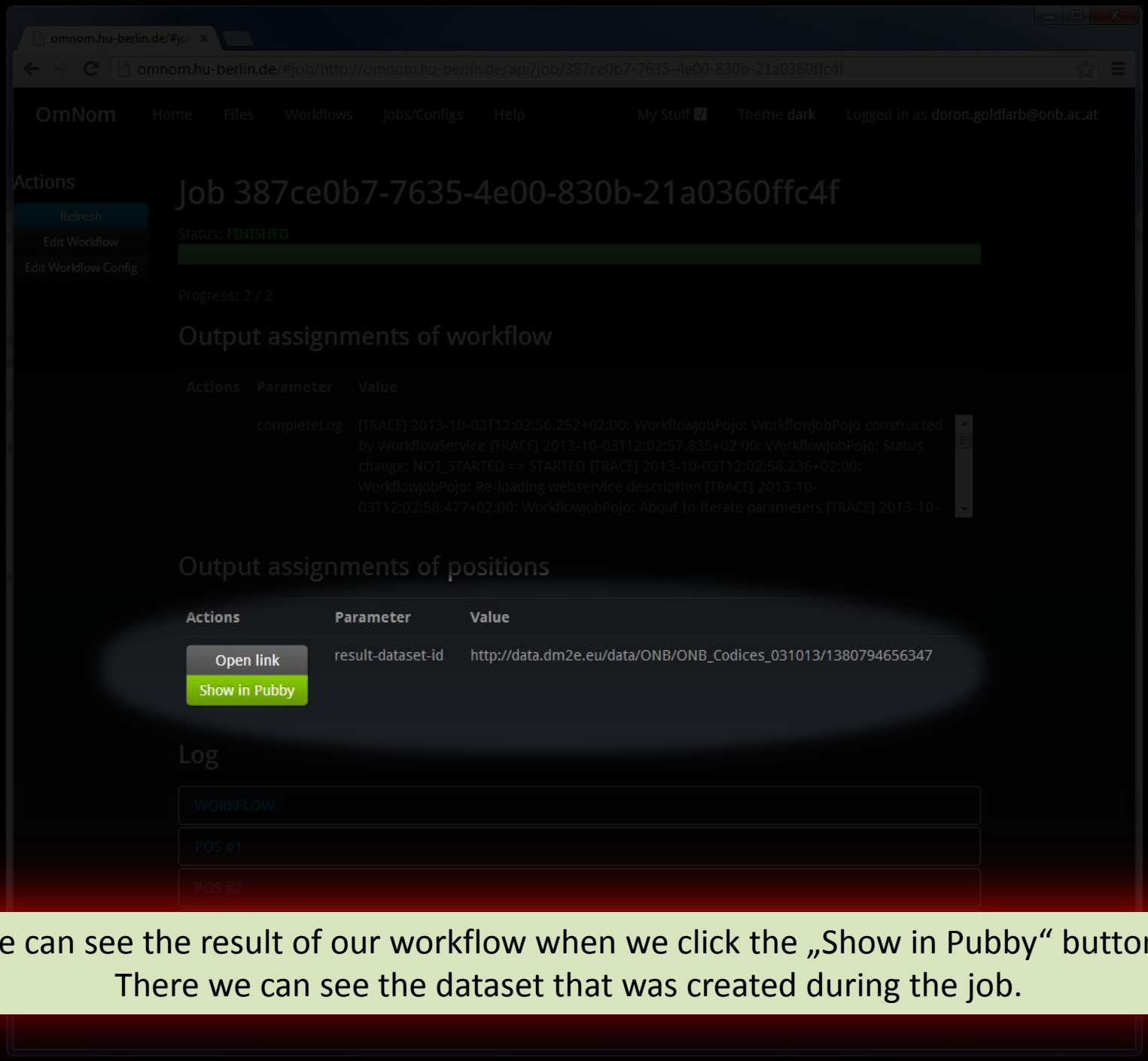
After we click on „Run“, we are redirected to the Job screen. Here we see the ID of the current job and ist current progress. Click „Refresh“ once in a while to track status changes.



The job has successfully passed Position #1 (XSLT) of the workflow and is now in Position #2 (Publish). Please continue to hit „Refresh“ in order to track progress.

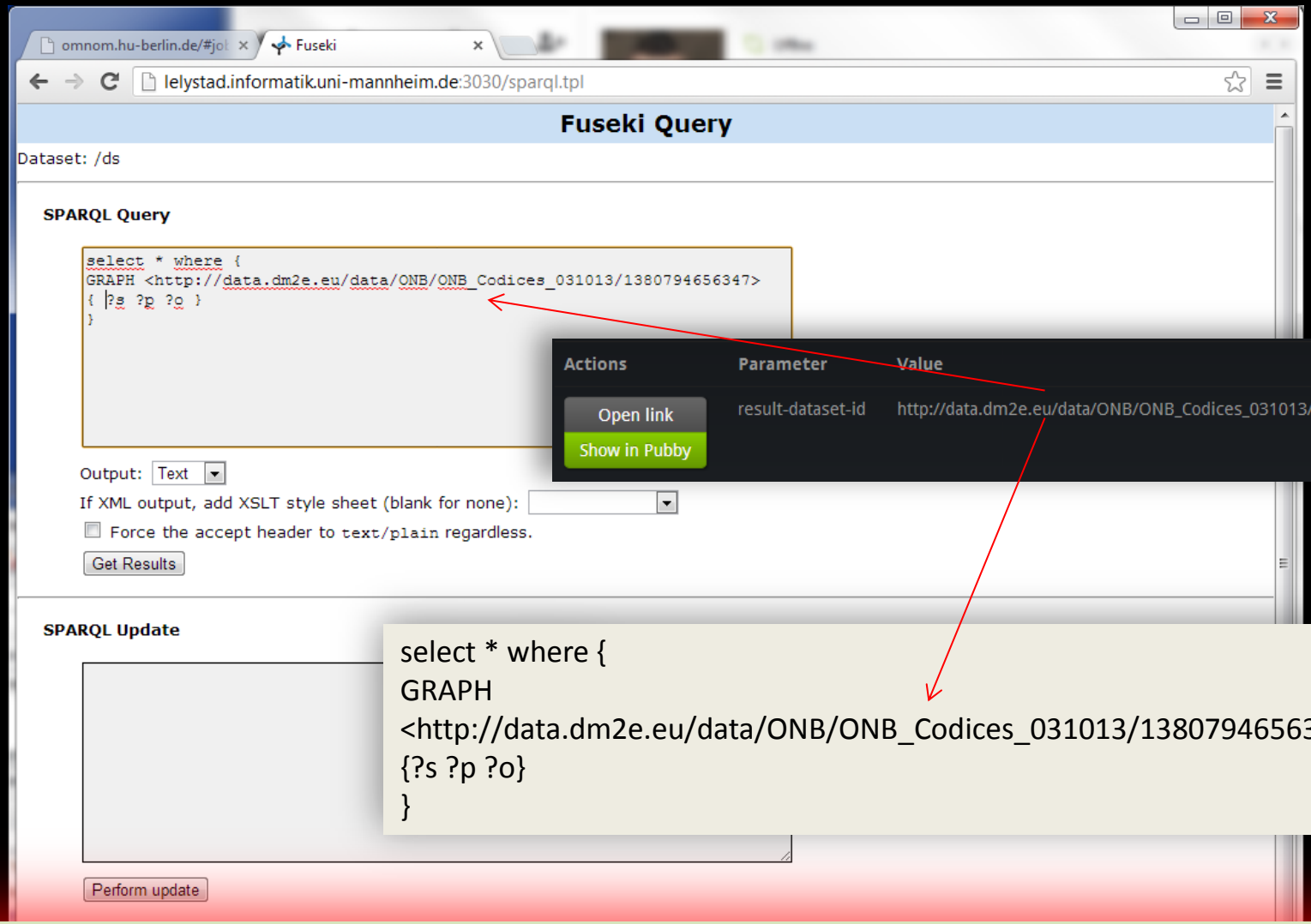


Here we see that the job has finished successfully.



We can see the result of our workflow when we click the „Show in Pubby“ button. There we can see the dataset that was created during the job.

This is the Pubby view of the dataset. Soon it should also be possible to browse through the ingested items from here.



select * where {
GRAPH
<http://data.dm2e.eu/data/ONB/ONB_Codices_031013/1380794656347>
{?s ?p ?o}
}

If you want to see the ingested metadata directly, you can use the SPARQL endpoint at http://lelystad.informatik.uni-mannheim.de:3030/control-panel.tpl . Click select and enter your query in the text box. Use the provided sample query, but remember to fill in the correct „result-dataset-id“ URI from your job result screen !!!

Please post bugs, suggestions for enhancement, questions to

https://github.com/DM2E/dm2e-gui/issues

Thank You !!!