Stoa User Guide v0.8

Peter R Hague

1 Introduction

Stoa is a workflow management system that keeps your code and data on a server and allows you and your collaborators to control them remotely through a web interface. It organises your data into worktables, into which your code is embedded through CWL (Common Workflow Language). Worktables are linked together into a higher level execution graph.

`STOA` is able to take the output of any worktable and present it online as a service to others. At present, the services supported are `fits` format download, and VO cone search.

2 Installation

Please download `STOA` from

`https://www.github.com/petehague/stoa`

`STOA` requires Python 3 and some Python libraries, all available through `pip` - we recommend that you use a virtual environment when trying this out. When you have one set up, install `numpy`, `astropy`, `cwltool`, `grpcio-tools` and `tornado` and you should be good to go. In order to prepare STOA to run, type

```
./ready.sh
```

this only need be done once per STOA install. Then type

```
./start.sh example 9000
```

to run the demo. Go to your browser and visit `localhost:9000` to try it out

2.1 First Use

Log in as `guest` (no password is required) and try to create a worktable to implement the 'find' command. Once this table is present, go into it and add a new row with 'product'

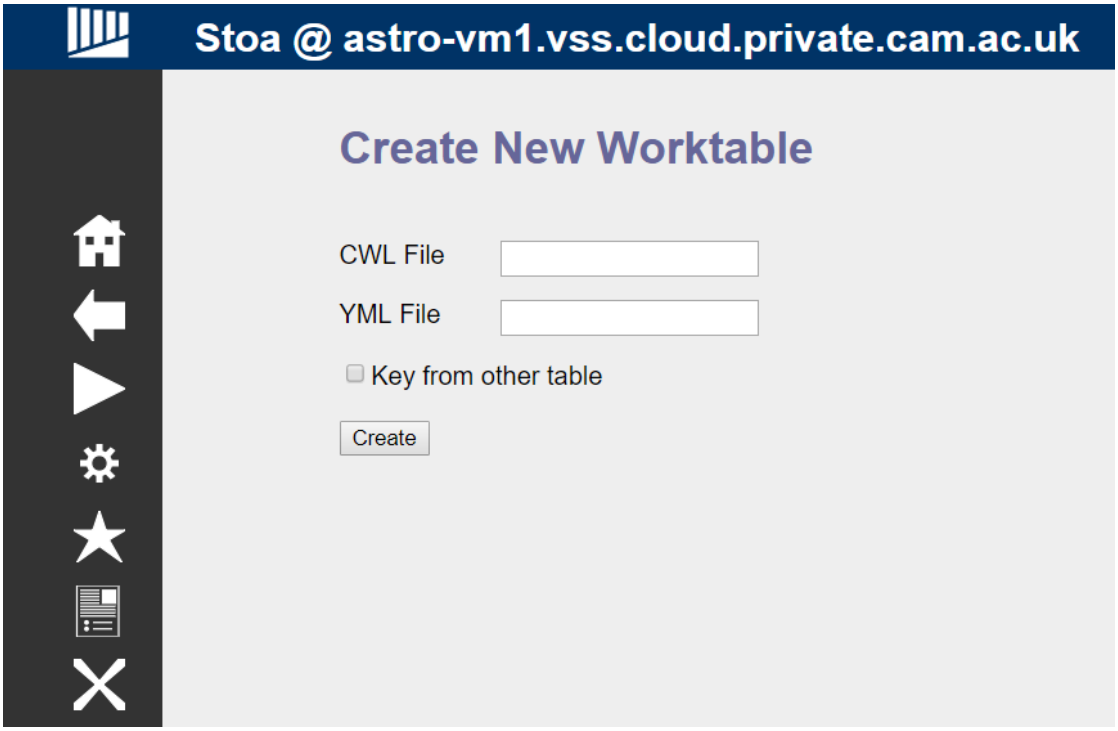

Figure 1: Dialogue box to create a new worktable

as its input, and then run that row.

3 Getting Started

In order to use `STOA`, it is first necessary to create CWL wrappers for all the code you need. This is not typically difficult, and some wrappers for simple functions are included already for your convenience. There is a user guide for CWL at

`https://www.commonwl.org/user_guide/`

which will teach you the basics of the language and quickly get you writing wrappers for your own scripts. In brief, CWL describes the tools you use in terms of their inputs and outputs, and then lets you combine them into workflows with linked inputs and outputs. Any command line tool whose operation is driven by its command line parameters, and can be modified to store all its output in a file named `cwl.output.json`, will have a very simple wrapper. Future versions of `STOA` will include a way to automatically generate such wrappers.

From the home screen, clicking on `Create New Table` gives the screen shown in Figure 1. This gives the option to select the CWL file you wish to use as a basis for

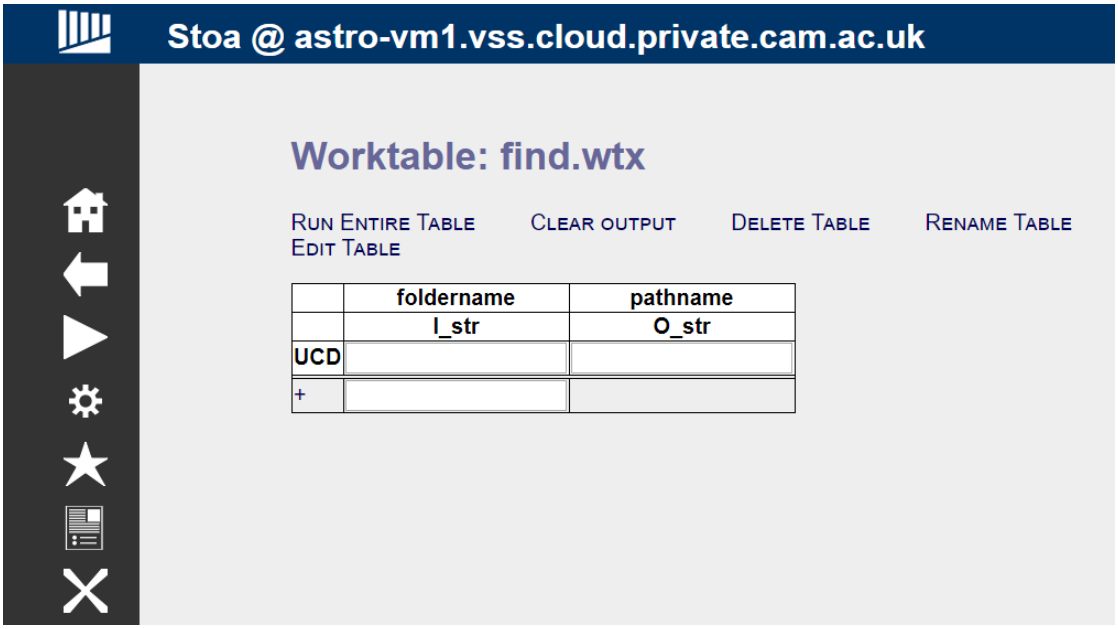

Figure 2: View of an empty worktable for the `find.cwl` action

the table, and also the `.yml` file that contains the defaults used to populate empty rows. The `STOA` install provides `find.cwl` and `find.yml` for testing purposes, so you should be able to immediately try this out.

Once the worktable is created, the screen should look like Figure 2. In order to run the worktable, it needs input for the workflow. Recall that each execution instance of a workflow is a row in the table, so in order to provide input for a workflow, we need to add a row. The dialogue box next to the '+' sign can be used to type in a value for this workflow's solitary input. Then click on the '+' to add a row.

4 Architecture

`STOA` runs as a cluster of services, one of which is a Tornado web application visible to the user. There is a service for managing user data, and the action server which manages the execution of workflows in the background. Ports 6999 and 7000 are required to be available on the host machine, alongside whichever port the web application service has been told to use.

4.1 Worktables

A worktable is a combination of a workflow and a table. The workflow defines the inputs and outputs of a process - be it a single command, or a set of connected ones - and these inputs and outputs form the colulmns of the worktable. Each row of the worktable is an input/output pair representing one execution of the workflow.

Worktables are stored on the disk as zip files with a .wtx extension. In addition to containing a CWL workflow and table data, they can also contain any files which are required by the workflow during its execution - the entire worktable file will be extracted into the execution directory. This is **not** intended as a way to store data - only code.

The data which a worktable operates on must be located in the project folder which the particular instance of `STOA` is operating on.