

# Taverna Server Ruby Gem

Robert Haines

School of Computer Science, University of Manchester,  
Manchester, United Kingdom

IMPACT Workshop, 14/11/2011

# What is it?

The t2-server ruby gem provides:

- A ruby API for programmatic access to the Taverna Server REST interface
- Commandline scripts for common tasks:
  - Server administration
  - Running workflows
  - Collecting workflow outputs
  - Deleting workflow runs

# How to get it

Just install directly from RubyGems:

```
$ [sudo] gem install t2-server
```

Documentation:

```
http://rubydoc.info/gems/t2-server/0.6.1/frames
```

Source available on GitHub:

```
https://github.com/myGrid/t2-server-gem
```

# Commandline scripts

# Commandline scripts

- Written in ruby
  - Useful tools
  - Exemplars of API usage
- The `--help` switch gives usage instructions for each script
  - `t2-delete-runs`
  - `t2-run-workflow`
  - `t2-server-admin`
  - `t2-get-output`
  - `t2-server-info`

Get general server information and your run list.

```
$ t2-server-info https://example.com/taverna
```

## Output

```
Server: https://example.com/taverna
```

```
Version: 2.3
```

```
Run limit: 100
```

```
No. of runs: 1
```

Run ID	Status	Expiry time (local)
0257e5f3 ...	Finished	15:57:35 11/11/2011

Run a workflow.

```
$ t2-run-workflow https://example.com/taverna <  
hellow.t2flow
```

## Output

```
Created run with uuid:  
    0257e5f3-0684-4c88-94c3-48633f6283f0  
Created at Thu Nov 10 15:57:34 UTC 2011  
Started at Thu Nov 10 15:57:39 UTC 2011  
Running.....  
Finished at Thu Nov 10 15:57:47 UTC 2011  
Outputs:  
    OUT (0) -> "Hello, World!"
```

Get output(s) from a workflow.

```
$ t2-get-output --total-size  
https://example.com/taverna  
0257e5f3-0684-4c88-94c3-48633f6283f0
```

## Output

OUT

Values: "Hello, World!"

Total data size: 13



# Ruby API

There are two main entry points:

## `T2Server::Run`

- Represents a single “run” of a workflow
- Provides full control over the run
- Use this for interacting with individual (or few) runs

## `T2Server::Server`

- Represents a Taverna Server instance
- Provides access to all of the runs on the server that the (authenticated) user has access to
- Use this for interacting with multiple runs or for finer grained control of the server itself

# Initialization

Load the library:

```
require 'rubygems'  
require 't2-server'
```

Server address and user credentials:

```
uri = "https://example.com/taverna"  
creds = T2Server::HttpBasic.new(username, password)
```

## Server information

```
T2Server::Server.new(uri) do |server|
  puts "    Server: #{server.uri}"
  puts "    Version: #{server.version}"
  puts "    Run limit: #{server.run_limit(creds)}"
  runs = server.runs(creds)
  puts "No. of runs: #{runs.length}"
end
```

# Initialize a workflow run

Via the Run interface:

```
T2Server::Run.create(uri, workflow, creds) do |run|  
  # ...  
end
```

Via the Server interface:

```
server.create_run(workflow, creds) do |run|  
  # ...  
end
```

# Set up the run

Set the expiry time

```
run.expiry = "18:00"
```

Set the inputs

```
ports = run.input_ports
```

```
# Set port named 'greeting' to a string
```

```
ports["greeting"].value = "Hello!"
```

```
# Set port named 'emails' to use a file
```

```
ports["emails"].file = "/opt/taverna/addresses.txt"
```

# Run it!

```
run.start  
puts "Started at #{run.start_time}"
```

Wait for the run to finish...

```
until run.finished?  
  sleep(1)  
end  
  
puts "Finished at #{run.finish_time}"
```

# Get the outputs

Just print them out:

```
run.output_ports.each_value do |port|  
  p port.values  
end
```

Or download all of them in a single zip file:

```
File.open("outputs.zip", "w") do |file|  
  file.syswrite(run.zip_output)  
end
```



# Delete the run

Via the Run interface:

```
run.delete
```

Via the Server interface:

```
server.delete_run(run, creds)
```

```
# Or delete all runs!
```

```
server.delete_all_runs(creds)
```

You may only delete a run if you have permission to do so!

The ruby gem can also:

- Upload inputs in Baclava format
- Retrieve outputs in Baclava format
- Get references to outputs rather than the actual data
- Download parts of outputs rather than the whole thing
- Grant/revoke permissions to allow other users to manipulate your runs
- Configure a running server (needs administrator access)