CBRAIN APIs (RPI-22)

canarie

McGill

cbrain-support.mni@mcgill.ca

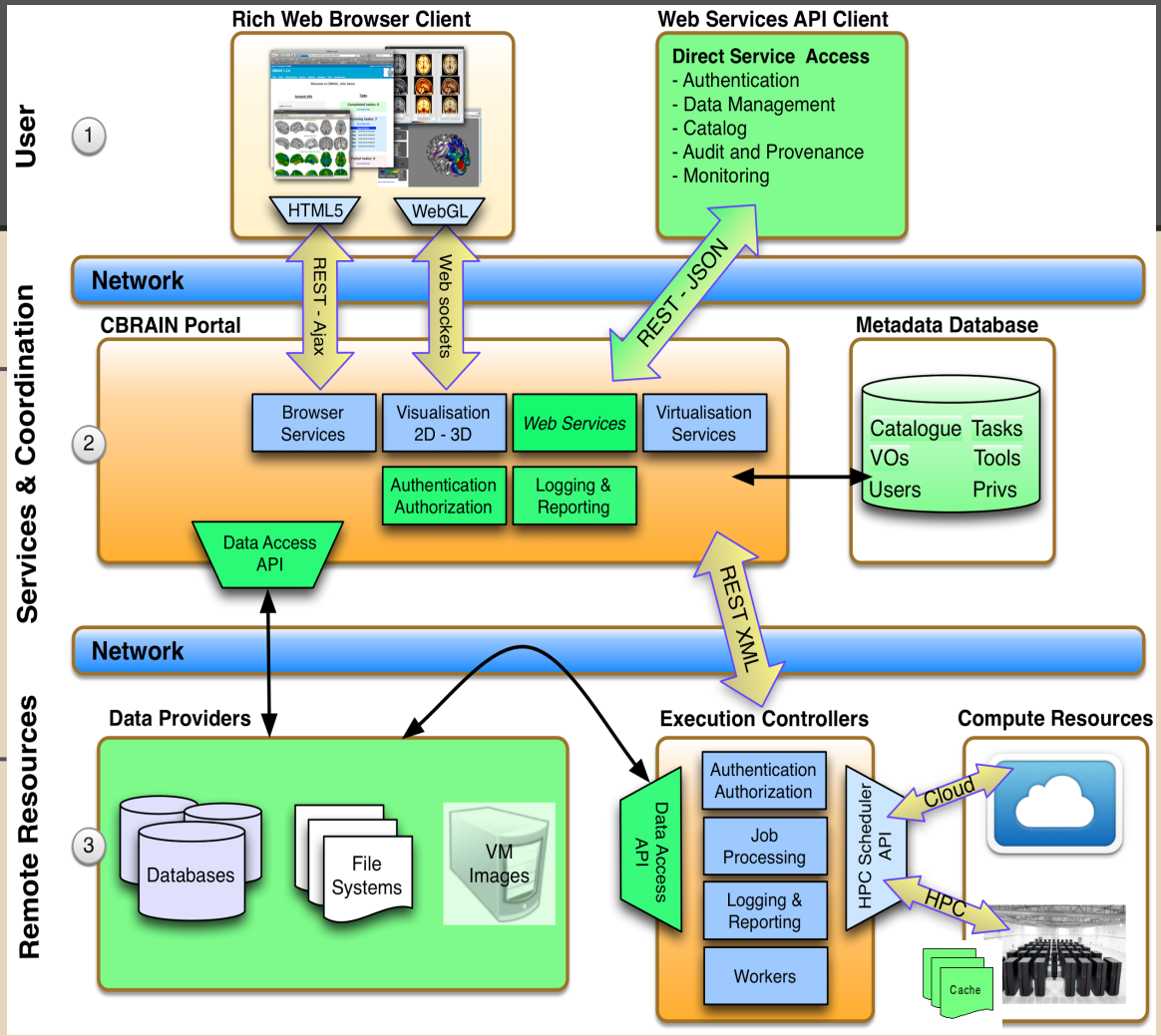
Overview

- CBRAIN
- API features
- Use case
- Documentation

CBRAIN

- Middleware infrastructure in place
- No software installation required

- Efficient file caching on multiple sites, with synchronization status monitoring
- Users can access to HPC



https://portal.cbrain.mcgill.ca/

API Features: Data Management

- **Data providers:**
  - List, get information
  - Create/delete/edit
- **Files catalogue:**
  - Un/register, list registered files
- **Data transfer:**
  - Move/copy files between data providers
  - File information/synchronization status
  - Download/upload/delete files

API Features: Compute Resources

- **Task:**
  - Create a task/launch job on compute resources
  - List/show tasks

API Features: Monitoring

- **Data providers:**
  - Online/alive status
  - Access permissions
  - Disk usage report
  - Etc...
- **Computing site:**
  - Online status
  - Uptime
  - Last measured queuing time
  - Etc...

API Internal Development Roadmap

Authentication and licenses (Test codes 0xx and 9xx)								
Rails Controller	**Rails Action**	**Perl API**	**Ruby API**	**JavaScript API**	**Perl Test**	**Ruby Test**	**JavaScript Test**	**Description**
session	new + create	login()	login()	login()	000,001	000,001		Authenticate
Userfiles Actions (Test codes 3xx)								
userfiles	index	index_userfiles()	index_userfiles()	index()	300	300		Get list of userfiles

```ruby
agent = CbrainRubyAPI.new("https://portal.cbrain.mcgill.ca")
agent.login("user_jdoe","FakePassw0rd!")
index = agent.index_userfiles({ :data_provider_id => 68})
```

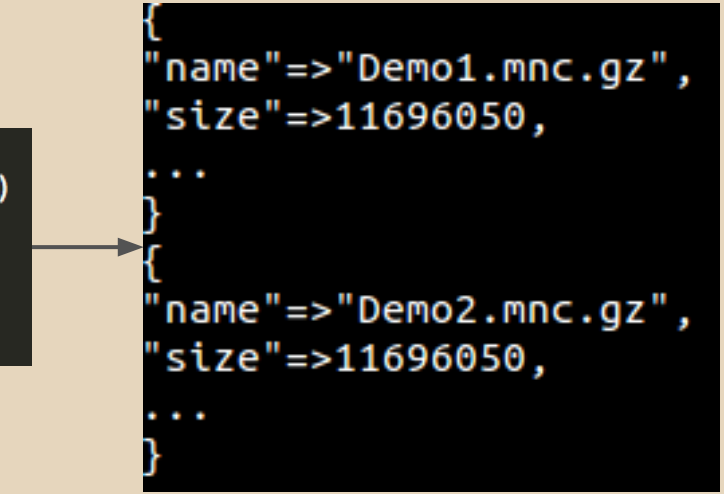
```perl
my $agent = CbrainAPI->new(cbrain_server_url => "https://portal.cbrain.mcgill.ca");
$agent->login("user_jdoe","FakePassw0rd!");
my $index = $agent->index_userfiles({"data_provider_id" => 68});
```

Use Case: Overview

- Explore files on a special data provider meant for registration
- Register some files with CBRAIN (add in catalog)
- Launch a task to process the files on a HPC
- Monitor task progress

Example: Query a Data Provider

```ruby
# Creation of a new agent and login
agent = CbrainRubyAPI.new("https://portal.cbrain.mcgill.ca")
agent.login("user_jdoe","FakePassw0rd!")

# Find all files on data provider
files_on_dp = agent.browse_data_provider(68)
```

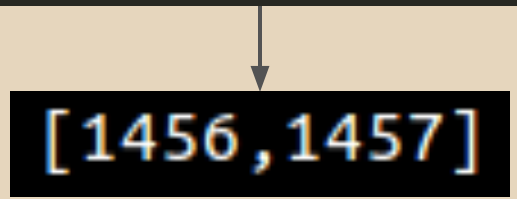


Example: Register New Files

```
# Register all the Demo*.mnc.gz file in CBRAIN
# These files are on DP but not yet in catalog
files_to_register = [ "Demo1.mnc.gz", "Demo2.mnc.gz" ]
userfile_infos    = agent.register_userfiles(files_to_register, "MincFile", 68)
```

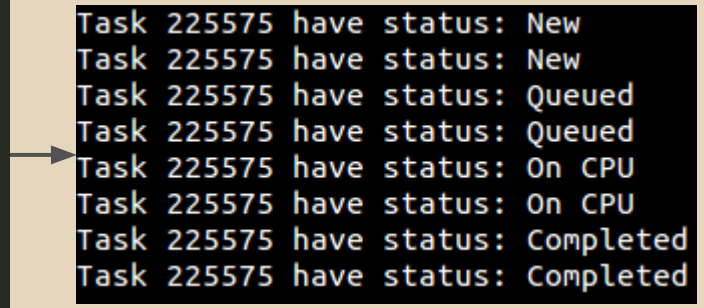
Example: Create/Launch Tasks

```
# Launch a task on these files
input_file_ids = [ 8363, 3436 ]
task_ids       = agent.create_task( input_file_ids , 210)
```



Example: Monitoring

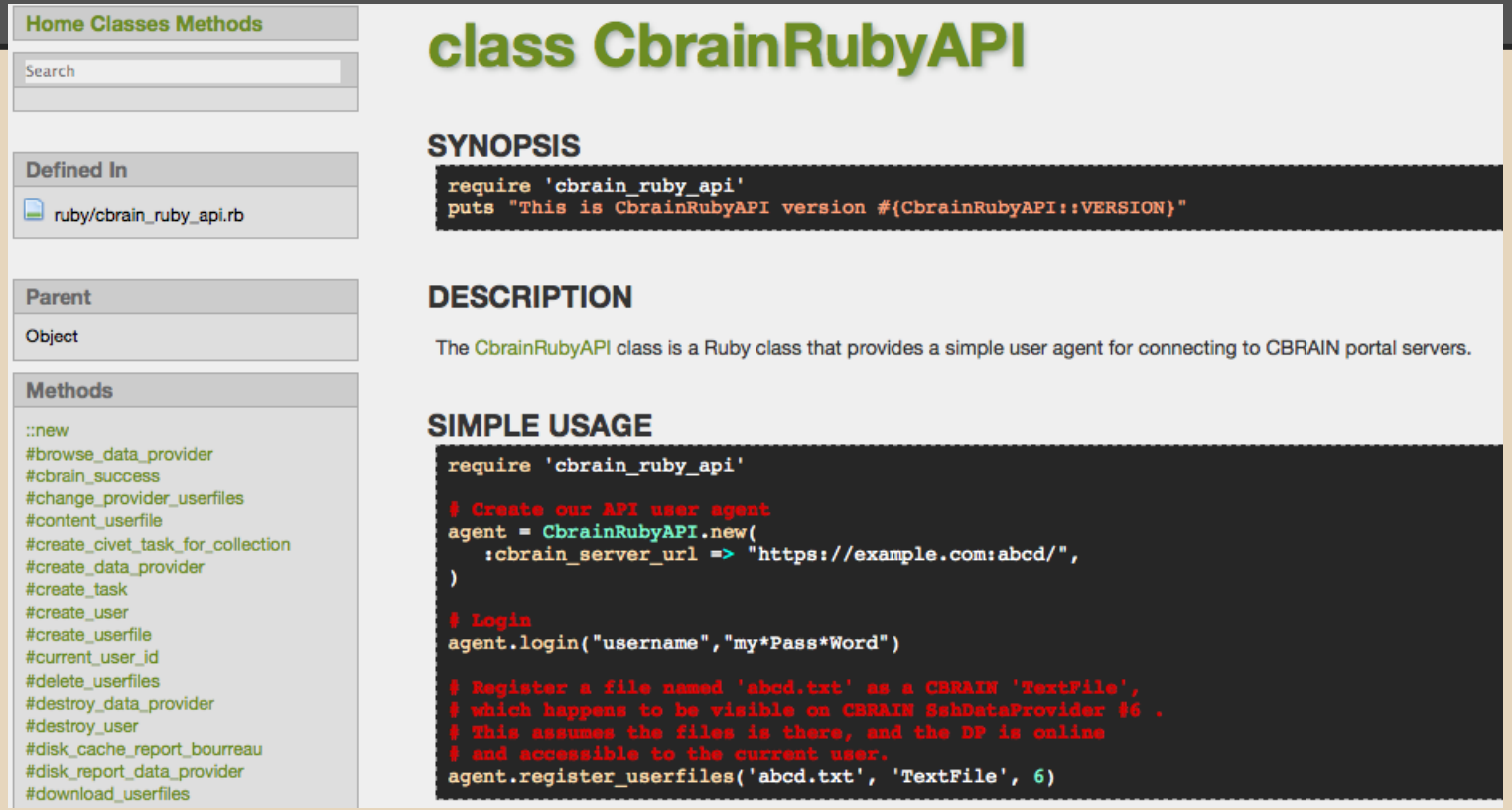
```ruby
# Monitoring of tasks status
task_ids_completed = []
until (task_ids.size == 0)
  task_ids.each do |id|
    show_task   = agent.show_task(id)
    task_status = show_task["status"]
    puts "Task #{id} have status: #{task_status}"
    task_ids_completed << id unless task_status =~ /New|Queued|On CPU/
  end
  task_ids -= task_ids_completed
  sleep(20)
end
```



Example:

```ruby
# Load the Ruby CBRAIN API
require './cbrain_ruby_api.rb'

# Creation of a new agent and login
agent = CbrainRubyAPI.new("https://portal.cbrain.mcgill.ca")
agent.login("user_jdoe","FakePassw0rd!")

# Find all files on data provider
files_on_dp = agent.browse_data_provider(68)

# Register all the Demo*.mnc.gz file in CBRAIN
# These files are on DP but not yet in catalog
unregistered_demo_filenames=[]
files_on_dp.each { |file| unregistered_demo_filenames << file[:name] if file[:name] =~ /Demo*.mnc.gz/ }
userfile_infos    = agent.register_userfiles(unregistered_demo_filenames, "MincFile", 68)

# Extract all ids of newly registered files
new_files_ids = []
userfile_infos[:newly_registered_userfiles].each { |file| new_files_ids << file[:id] }

# Launch a task on these files
task_ids = agent.create_task( new_files_ids , 210, {}, {'conv_direction' => 'minc2'} )

# Monitoring of tasks status
task_ids_completed = []
until (task_ids.size == 0)
  task_ids.each do |id|
    show_task   = agent.show_task(id)
    task_status = show_task["status"]
    puts "Task #{id} have status: #{task_status}"
    task_ids_completed << id unless task_status =~ /New|Queued|On CPU/
  end
  task_ids -= task_ids_completed
  sleep(20)
end
```

Documentation



Home Classes Methods

Search

Defined In

ruby/cbrain_ruby_api.rb

Parent

Object

Methods

::new
#browse_data_provider
#cbrain_success
#change_provider_userfiles
#content_userfile
#create_civet_task_for_collection
#create_data_provider
#create_task
#create_user
#create_userfile
#current_user_id
#delete_userfiles
#destroy_data_provider
#destroy_user
#disk_cache_report_bourreau
#disk_report_data_provider
#download_userfiles

class CbrainRubyAPI

SYNOPSIS

```
require 'cbrain_ruby_api'
puts "This is CbrainRubyAPI version #{CbrainRubyAPI::VERSION}"
```

DESCRIPTION

The CbrainRubyAPI class is a Ruby class that provides a simple user agent for connecting to CBRAIN portal servers.

SIMPLE USAGE

```
require 'cbrain_ruby_api'

# Create our API user agent
agent = CbrainRubyAPI.new(
   :cbrain_server_url => "https://example.com:abcd/",
)

# Login
agent.login("username","my*Pass*Word")

# Register a file named 'abcd.txt' as a CBRAIN 'TextFile',
# which happens to be visible on CBRAIN SshDataProvider #6 .
# This assumes the files is there, and the DP is online
# and accessible to the current user.
agent.register_userfiles('abcd.txt', 'TextFile', 6)
```

Some Projects Using the APIs

- Internal projects
  - Automation of administrative tasks for CBRAIN
  - Data processing from the LORIS database
- External projects
  - Data transfers for the genetics and genomic analysis platform
  - Data processing from European “Neugrid for Users”

Acknowledgements

- Project funding: Canarie (RPI-22)
- Principal Investigator: Alan C. Evans
- Programs Manager: Reza Adalat
- CBRAIN Team: Marc Rousseau, Pierre Rioux, Tarek Sherif, Alden Woodward
- Visiting scholar: Tristan Glatard