This is transmissionrpc. This module helps using Python to connect to a Transmission JSON-RPC service. transmissionrpc is compatible with Transmission 1.3 and later.

transmissionrpc is licensed under the MIT license.
Transmission is available at Python Package Index. To install the transmissionrpc python module use easy_install or pip.

```
$ easy_install transmissionrpc
```

**Note:** You might need administrator privileges to install python modules.

You may also download the tarball from Python Package Index. Untar and install.

```
$ tar -xzf transmissionrpc-0.7.tar.gz
$ cd transmissionrpc-0.7
$ python setup.py install
```

## 2.1 Dependencies

transmissionrpc has the following dependencies.

- Python >= 2.5.
- simplejson >= 1.7.1 or Python >= 2.6.

If Python 2.6 or later is detected the standard library json implementation will be used.

## 2.2 Report a problem

Problems with transmissionrpc should be reported through the issue tracker at bitbucket. Please look through the existing issues before opening a new issue.
3.1 The source code

Transmission is hosted at bitbucket using mercurial. To get a working copy, run

$ hg clone http://www.bitbucket.org/blueluna/transmissionrpc/

The source code will be fetched and stored the directory transmissionrpc.

Then install the module using

$ python setup.py install

Or if you wish to further develop transmissionrpc itself use

$ python setup.py develop

This will link this directory to the library as transmissionrpc.

3.2 Poking around

Now that transmissionrpc has been installed, run python and start to poke around. Following will create a RPC client and list all torrents.

```python
>>> import transmissionrpc
>>> tc = transmissionrpc.Client('localhost', port=9091)
>>> tc.list()
```

List will return a dictionary of Torrent object indexed by their id. You might not have any torrents yet. This can be remedied by adding an torrent.

```python
{1: <Torrent 1 "ubuntu-8.10-desktop-i386.iso">}
>>> tc.info(1)
{1: <Torrent 1 "ubuntu-8.10-desktop-i386.iso">}
```

As you saw, the add_url and info calls also returns a dictionary with `{<id>: <Torrent>, ...}`. More information about a torrent transfer can be found in the Torrent object.

```python
>>> torrent = tc.info(1)[1]
>>> torrent.name
'ubuntu-8.10-desktop-i386.iso'
>>> torrent.hashString
```
'33820db6dd5e5928d23bc811bbac2f4ae94cb882'
>>> torrent.status
'downloading'
>>> torrent.eta
datetime.timedelta(0, 750)
>>> for key, value in torrent.fields.iteritems():
...    print(key, value)
...
('comment', 'Ubuntu CD releases.ubuntu.com')

The last call will list all known data provided by Transmission.

Well, we weren’t that interested in Ubuntu so lets stop the transfer and the remove it.

>>> tc.stop(1)
>>> tc.remove('33820db6dd5e5928d23bc811bbac2f4ae94cb882')

See what we did there? most methods in transmissionrpc can take both torrent id and torrent hash when referring to a torrent. lists and sequences are also supported.

>>> tc.info([2, 'caff87b88f50f46bc22da3a2712a6a4e9a98d91e'])
{2: <Torrent 2 "ubuntu-8.10-server-amd64.iso">, 3: <Torrent 3 "ubuntu-8.10-alternate-amd64.iso">}
>>> tc.info('1:3')
{2: <Torrent 2 "ubuntu-8.10-server-amd64.iso">, 3: <Torrent 3 "ubuntu-8.10-alternate-amd64.iso">}

Continue to explore and have fun! For more in depth information read the module reference.

### 3.3 A note about debugging information

If you ever need to see what’s going on inside transmissionrpc, you can change the logging level of transmissionrpc. This is done with these easy steps

```python
>>> import logging

>>> logging.getLogger('transmissionrpc').setLevel(logging.DEBUG)
```

Note that this will produce a whole lot of output! Other levels are (listed by severity)

- `logging.ERROR`
- `logging.WARNING`
- `logging.INFO`
- `logging.DEBUG`

The default logging level of transmissionrpc is `logging.ERROR`. 
CHAPTER
FOUR

MODULE REFERENCE

4.1 transmissionrpc — Module reference

This documentation will not describe all RPC fields in detail. Please refer to the RPC specification for more information on RPC data.

Contents

- transmissionrpc — Module reference
  - Exceptions
  - Torrent object
  - Session object
  - Client object
    - Torrent ids
    - Timeouts

4.1.1 Exceptions

class transmissionrpc.TransmissionError (message='', original=None)

This exception is raised when there has occurred an error related to communication with Transmission. It is a subclass of Exception.

original

The original exception.

class transmissionrpc.HTTPHandlerError (httpurl=None, httpcode=None, httpmsg=None, httpheaders=None, httpdata=None)

This exception is raised when there has occurred an error related to the HTTP handler. It is a subclass of Exception.

url

The requested url.

code

HTTP error code.

message

HTTP error message.

headers

HTTP headers.
4.1.2 Torrent object

Torrent is a class holding the information received from Transmission regarding a bittorrent transfer. All fetched torrent fields are accessible through this class using attributes. The attributes use underscore instead of hyphen in the names though. This class has a few convenience attributes using the torrent information.

Example:

```python
>>> import transmissionrpc
>>> t = transmissionrpc.Torrent({'id': 1, 'comment': 'My torrent', 'addedDate': 1232281019})
>>> t.comment
'My torrent'
>>> t.date_added
datetime.datetime(2009, 1, 18, 13, 16, 59)
```

class transmissionrpc.Torrent (client, fields)

Torrent is a class holding the data received from Transmission regarding a bittorrent transfer. All fetched torrent fields are accessible through this class using attributes. This class has a few convenience properties using the torrent data.

- **date_active**
  Get the attribute “activityDate” as datetime.datetime.

- **date_added**
  Get the attribute “addedDate” as datetime.datetime.

- **date_done**
  Get the attribute “doneDate” as datetime.datetime.

- **date_started**
  Get the attribute “startDate” as datetime.datetime.

- **eta**
  Get the “eta” as datetime.timedelta.

- **files ()**
  Get list of files for this torrent.

  This function returns a dictionary with file information for each file. The file information is has following fields:

  ```python
  {
      <file id>: {
          'name': <file name>,
          'size': <file size in bytes>,
          'completed': <bytes completed>,
          'priority': <priority ('high'|'normal'|'low')>,
          'selected': <selected for download>
      }
  ...
  }
  ```

- **format_eta ()**
  Returns the attribute *eta* formatted as a string.
• If eta is -1 the result is ‘not available’
• If eta is -2 the result is ‘unknown’
• Otherwise eta is formatted as <days>:<hours>:<minutes>:<seconds>.

**priority**
*Get the priority as string. Can be one of ‘low’, ‘normal’, ‘high’.*

**progress**
*Get the download progress in percent.*

**ratio**
*Get the upload/download ratio.*

**status**
*Returns the torrent status. Is either one of ‘check pending’, ‘checking’, ‘downloading’, ‘seeding’ or ‘stopped’. The first two is related to verification.*

**update**(other)
*Update the torrent data from a Transmission JSON-RPC arguments dictionary*

### 4.1.3 Session object

Session is a class holding the session data for a Transmission session.

Access the session field can be done through attributes. The attributes available are the same as the session arguments in the Transmission RPC specification, but with underscore instead of hyphen. `download-dir` -> `download_dir`.

```python
class transmissionrpc.Session(fields = {})
    fields should be a dictionary build from session information from an Transmission JSON-RPC result.

Session.update(other)
    Updates the Session object with data from `other`.
    `other` should be a Session object or session information from an Transmission JSON-RPC result.
```

### 4.1.4 Client object

This is it. This class implements the JSON-RPC protocol to communicate with Transmission.

**Torrent ids**

Many functions in Client takes torrent id. A torrent id can either be id or hashString. When suppling multiple id’s it is possible to use a list mixed with both id and hashString.

**Timeouts**

In Python 2.6 it is possible to supply a timeout to a HTTP request. This is accessible through transmissionrpc by either changing the timeout property of a Client object or supply the named argument `timeout` in most methods of Client. The default timeout is 30 seconds.

```python
class transmissionrpc.Client(address='localhost', port=9091, user=None, password=None,
                           http_handler=None, timeout=None)
    Client is the class handling the Transmission JSON-RPC client protocol.
```
add *(data, timeout=None, **kwargs)*

Add torrent to transfers list. Takes a base64 encoded .torrent file in data. Additional arguments are:

<table>
<thead>
<tr>
<th>Argument</th>
<th>RPC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bandwidthPriority</td>
<td>8</td>
<td>Priority for this transfer.</td>
</tr>
<tr>
<td>download_dir</td>
<td>1</td>
<td>The directory where the downloaded contents will be saved in.</td>
</tr>
<tr>
<td>filename</td>
<td>1</td>
<td>A filepath or URL to a torrent file or a magnet link.</td>
</tr>
<tr>
<td>files_unwanted</td>
<td>1</td>
<td>A list of file id’s that shouldn’t be downloaded.</td>
</tr>
<tr>
<td>files_wanted</td>
<td>1</td>
<td>A list of file id’s that should be downloaded.</td>
</tr>
<tr>
<td>metainfo</td>
<td>1</td>
<td>The content of a torrent file, base64 encoded.</td>
</tr>
<tr>
<td>paused</td>
<td>1</td>
<td>If True, does not start the transfer when added.</td>
</tr>
<tr>
<td>peer_limit</td>
<td>1</td>
<td>Maximum number of peers allowed.</td>
</tr>
<tr>
<td>priority_high</td>
<td>1</td>
<td>A list of file id’s that should have high priority.</td>
</tr>
<tr>
<td>priority_low</td>
<td>1</td>
<td>A list of file id’s that should have low priority.</td>
</tr>
<tr>
<td>priority_normal</td>
<td>1</td>
<td>A list of file id’s that should have normal priority.</td>
</tr>
</tbody>
</table>

add_uri *(uri, **kwargs)*

Add torrent to transfers list. Takes a uri to a torrent, supporting all uri’s supported by Transmissions torrent-add ‘filename’ argument. Additional arguments are:

<table>
<thead>
<tr>
<th>Argument</th>
<th>RPC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bandwidthPriority</td>
<td>8</td>
<td>Priority for this transfer.</td>
</tr>
<tr>
<td>download_dir</td>
<td>1</td>
<td>The directory where the downloaded contents will be saved in.</td>
</tr>
<tr>
<td>files_unwanted</td>
<td>1</td>
<td>A list of file id’s that shouldn’t be downloaded.</td>
</tr>
<tr>
<td>files_wanted</td>
<td>1</td>
<td>A list of file id’s that should be downloaded.</td>
</tr>
<tr>
<td>paused</td>
<td>1</td>
<td>If True, does not start the transfer when added.</td>
</tr>
<tr>
<td>peer_limit</td>
<td>1</td>
<td>Maximum number of peers allowed.</td>
</tr>
<tr>
<td>priority_high</td>
<td>1</td>
<td>A list of file id’s that should have high priority.</td>
</tr>
<tr>
<td>priority_low</td>
<td>1</td>
<td>A list of file id’s that should have low priority.</td>
</tr>
<tr>
<td>priority_normal</td>
<td>1</td>
<td>A list of file id’s that should have normal priority.</td>
</tr>
</tbody>
</table>

blocklist_update *(timeout=None)*

Update block list. Returns the size of the block list.

change *(ids, timeout=None, **kwargs)*

Change torrent parameters for the torrent(s) with the supplied id’s. The parameters are:
<table>
<thead>
<tr>
<th>Argument</th>
<th>RPC</th>
<th>Replaced by</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bandwidthPriority</td>
<td>5</td>
<td></td>
<td>Priority for this transfer.</td>
</tr>
<tr>
<td>downloadLimit</td>
<td>5</td>
<td></td>
<td>Set the speed limit for download in Kib/s.</td>
</tr>
<tr>
<td>downloadLimited</td>
<td>5</td>
<td></td>
<td>Enable download speed limiter.</td>
</tr>
<tr>
<td>files_unwanted</td>
<td>1</td>
<td></td>
<td>A list of file id’s that shouldn’t be downloaded.</td>
</tr>
<tr>
<td>files_wanted</td>
<td>1</td>
<td></td>
<td>A list of file id’s that should be downloaded.</td>
</tr>
<tr>
<td>honorsSessionLimits</td>
<td>5</td>
<td></td>
<td>Enables or disables the transfer to honour the upload limit set in the session.</td>
</tr>
<tr>
<td>ids</td>
<td>1</td>
<td></td>
<td>Local download location.</td>
</tr>
<tr>
<td>peer_limit</td>
<td>1</td>
<td></td>
<td>The peer limit for the torrents.</td>
</tr>
<tr>
<td>priority_high</td>
<td>1</td>
<td></td>
<td>A list of file id’s that should have high priority.</td>
</tr>
<tr>
<td>priority_low</td>
<td>1</td>
<td></td>
<td>A list of file id’s that should have normal priority.</td>
</tr>
<tr>
<td>priority_normal</td>
<td>1</td>
<td></td>
<td>A list of file id’s that should have low priority.</td>
</tr>
<tr>
<td>seedIdleLimit</td>
<td>10</td>
<td></td>
<td>Seed inactivity limit in minutes.</td>
</tr>
<tr>
<td>seedIdleMode</td>
<td>10</td>
<td></td>
<td>Seed inactivity mode. 0 = Use session limit, 1 = Use transfer limit, 2 = Disable limit.</td>
</tr>
<tr>
<td>seedRatioLimit</td>
<td>5</td>
<td></td>
<td>Seeding ratio.</td>
</tr>
<tr>
<td>seedRatioMode</td>
<td>5</td>
<td></td>
<td>Which ratio to use. 0 = Use session limit, 1 = Use transfer limit, 2 = Disable limit.</td>
</tr>
<tr>
<td>speed_limit_down</td>
<td>1</td>
<td>download-Limit</td>
<td>Set the speed limit for download in Kib/s.</td>
</tr>
<tr>
<td>speed_limit_down_enabled</td>
<td>5</td>
<td>download-Limited</td>
<td>Enable download speed limiter.</td>
</tr>
<tr>
<td>speed_limit_up</td>
<td>1</td>
<td>upload-Limit</td>
<td>Set the speed limit for upload in Kib/s.</td>
</tr>
<tr>
<td>speed_limit_up_enabled</td>
<td>5</td>
<td>upload-Limit</td>
<td>Enable upload speed limiter.</td>
</tr>
<tr>
<td>trackerAdd</td>
<td>10</td>
<td></td>
<td>Array of string with announce URLs to add.</td>
</tr>
<tr>
<td>trackerRemove</td>
<td>10</td>
<td></td>
<td>Array of ids of trackers to remove.</td>
</tr>
<tr>
<td>trackerReplace</td>
<td>10</td>
<td></td>
<td>Array of (id, url) tuples where the announce URL should be replaced.</td>
</tr>
<tr>
<td>uploadLimit</td>
<td>5</td>
<td></td>
<td>Set the speed limit for upload in Kib/s.</td>
</tr>
<tr>
<td>uploadLimited</td>
<td>5</td>
<td></td>
<td>Enable upload speed limiter.</td>
</tr>
</tbody>
</table>

**Note:** transmissionrpc will try to automatically fix argument errors.

**get_files**(ids=None, timeout=None)

Get list of files for provided torrent id(s). If ids is empty, information for all torrents are fetched. This function returns a dictionary for each requested torrent id holding the information about the files.

```python
{
    <torrent id>: {
        <file id>: {
            'name': <file name>,
            'size': <file size in bytes>,
            'completed': <bytes completed>,
            'priority': <priority ('high'|'normal'|'low')>,
            'selected': <selected for download (True|False)>
        }
    }
    ...
}
```
 Transmission Documentation, Release 0.7

...
Table 4.1 – continued from previous page

<table>
<thead>
<tr>
<th>Argument</th>
<th>RPC</th>
<th>Replaced by</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alt_speed_up</td>
<td>5</td>
<td></td>
<td>Alternate session upload speed limit (in Kib/s).</td>
</tr>
<tr>
<td>blocklist_enabled</td>
<td>5</td>
<td></td>
<td>Enables the block list</td>
</tr>
<tr>
<td>cache_size_mb</td>
<td>10</td>
<td></td>
<td>The maximum size of the disk cache in MB</td>
</tr>
<tr>
<td>dht_enabled</td>
<td>6</td>
<td></td>
<td>Enables DHT.</td>
</tr>
<tr>
<td>download_dir</td>
<td>1</td>
<td></td>
<td>Set the session download directory.</td>
</tr>
<tr>
<td>encryption</td>
<td>1</td>
<td></td>
<td>Set the session encryption mode, one of required, preferred or tolerated.</td>
</tr>
<tr>
<td>idle_seeding_limit</td>
<td>10</td>
<td></td>
<td>The default seed inactivity limit in minutes.</td>
</tr>
<tr>
<td>idle_seeding_limit_enabled</td>
<td>10</td>
<td></td>
<td>Enables the default seed inactivity limit</td>
</tr>
<tr>
<td>incomplete_dir</td>
<td>7</td>
<td></td>
<td>The path to the directory of incomplete transfer data.</td>
</tr>
<tr>
<td>incomplete_dir_enabled</td>
<td>7</td>
<td></td>
<td>Enables the incomplete transfer data directory. Otherwise data is stored in download directory.</td>
</tr>
<tr>
<td>lpd_enabled</td>
<td>9</td>
<td></td>
<td>Enables local peer discovery for public torrents.</td>
</tr>
<tr>
<td>peer_limit</td>
<td>1 - 5</td>
<td>peer-limit-global</td>
<td>Maximum number of peers</td>
</tr>
<tr>
<td>peer_limit_global</td>
<td>5</td>
<td></td>
<td>Maximum number of peers</td>
</tr>
<tr>
<td>peer_limit_per_torrent</td>
<td>5</td>
<td></td>
<td>Maximum number of peers per transfer</td>
</tr>
<tr>
<td>peer_port</td>
<td>5</td>
<td></td>
<td>Peer port.</td>
</tr>
<tr>
<td>peer_port_random_on_start</td>
<td>5</td>
<td></td>
<td>Enables randomized peer port on start of Transmission.</td>
</tr>
<tr>
<td>pex_allowed</td>
<td>1 - 5</td>
<td>pex-enabled</td>
<td>Allowing PEX in public torrents.</td>
</tr>
<tr>
<td>pex_enabled</td>
<td>5</td>
<td></td>
<td>Allowing PEX in public torrents.</td>
</tr>
<tr>
<td>port</td>
<td>1 - 5</td>
<td>peer-port</td>
<td>Peer port.</td>
</tr>
<tr>
<td>port_forwarding_enabled</td>
<td>1</td>
<td></td>
<td>Enables port forwarding.</td>
</tr>
<tr>
<td>rename_partial_files</td>
<td>8</td>
<td></td>
<td>Appends &quot;.part&quot; to incomplete files.</td>
</tr>
<tr>
<td>seedRatioLimit</td>
<td>5</td>
<td></td>
<td>Seed ratio limit. 1.0 means 1:1 download and upload ratio.</td>
</tr>
<tr>
<td>seedRatioLimited</td>
<td>5</td>
<td></td>
<td>Enables seed ration limit.</td>
</tr>
<tr>
<td>speed_limit_down</td>
<td>1</td>
<td></td>
<td>Download speed limit (in Kib/s).</td>
</tr>
<tr>
<td>speed_limit_down_enabled</td>
<td>1</td>
<td></td>
<td>Enables download speed limiting.</td>
</tr>
<tr>
<td>speed_limit_up</td>
<td>1</td>
<td></td>
<td>Upload speed limit (in Kib/s).</td>
</tr>
<tr>
<td>speed_limit_up_enabled</td>
<td>1</td>
<td></td>
<td>Enables upload speed limiting.</td>
</tr>
<tr>
<td>start_added_torrents</td>
<td>9</td>
<td></td>
<td>Added torrents will be started right away.</td>
</tr>
<tr>
<td>trash_original_torrent_files</td>
<td>9</td>
<td></td>
<td>The .torrent file of added torrents will be deleted.</td>
</tr>
</tbody>
</table>

**Note:** transmissionrpc will try to automatically fix argument errors.

**start** *(ids, timeout=None)*
start torrent(s) with provided id(s)

**stop** *(ids, timeout=None)*
stop torrent(s) with provided id(s)

**timeout**
HTTP query timeout.

**verify** *(ids, timeout=None)*
verify torrent(s) with provided id(s)
INDICES AND TABLES

- genindex
- modindex
- search
t
transmissionrpc.??