

Setting up Mechanical Turk, *boto(3)*, and *mturkutils*

Florian Jaeger with help from Andrew Watts, Linda Liu, Zach Burchill, and Xin Xie

November 12, 2021

Contents

1 Overview

This document walks you through how to set up and use Amazon’s Mechanical Turk to conduct web-based experiments. Mechanical Turk (short MTurk) is a crowdsourcing platform that gives you access to hundreds of thousands of potential participants for your experiments. Amazon calls them *workers* and calls you, the experimenter, the *requester*. Experiments are called *Human Intelligence Tasks (HITs)* and individual instances of these HITs that workers can complete are called *assignments*. MTurk provides many ready-made templates for experiments that don’t require you to run a web server or to do most of the steps in this document. However, if you want to collect information like reaction times, etc. you will have to program your own experiments, put them on a webserver, and download participants’ responses from MTurk. This will require you to interact with MTurk through their python-based command line interface called *boto3*. There are now shellscripts that simplify this interaction that we will call *mturkutils*. This document walks you through all the steps to set up the necessary software environment and online accounts (all of them are free, but you need to pay workers who complete your HIT assignments, and Amazon collects a fee on top of each worker payment).

2 Set up Mechanical Turk accounts

If you are not already familiar with Amazon Mechanical Turk, the steps below will help you get started:

1. Register for an MTurk requester account (<https://www.mturk.com/mturk/welcome>), sandbox requester account (<https://requestersandbox.mturk.com>), sandbox worker account (<https://workersandbox.mturk.com>), and an Amazon Web Services (AWS) account (<https://aws.amazon.com/>).
 - (a) You can use the same email address that you use for your main Amazon.com account to make things easier, but you might want to make a separate account (if you plan to share your password with other members of the group).
 - (b) For the AWS account, you should select Basic access when prompted. Note that you will be asked to enter a credit card (though you will not be charged for this class).
 - (c) **Link your AWS account with your MTurk requester account.** To do that, go to <https://requester.mturk.com/developer>. Your AWS account should already show under Step 1 of that page. Under Step 2 of that page, there is an orange button “Link your AWS account” to link the requester account to AWS. Do that.
 - (d) **It may take up to a few days before your account(s) are confirmed, so be sure to sign up early.**
 - (e) If you have a non-U.S. address, your account may be denied.

2. Sign in to your AWS account and navigate to the console (<https://console.aws.amazon.com/iam/home>). Under *Access management* → *Users*, click the blue button that says *Create New Users*. This will start the user creation process, which consists of several steps.
 - (a) On the first screen, it doesn't matter what you enter for your username, so pick whatever you want.
 - (b) Select *Programmatic access*.
 - (c) On the next screen, select *Attach existing policies directly* and search for "AmazonMechanical-TurkFullAccess". Then select this policy by clicking the check box to its left.
 - (d) You can skip the subsequent screens. You don't need to select any tags. Just complete all steps and press the *Create user* button.
3. Once you create a user, you will be provided with an access key and a secret key. **These are very important. You must save them somewhere since you will need to refer to it later. And, do not share the secret key with others!**

3 Install Python and boto3

1. Begin by install Python (version 3 or higher). **If you had an earlier version of Python installed, the new version might be linked as Python3 and pip might be linked as pip3, and you might have to adjust the commands below accordingly.** pip helps you install Python packages. It should come with any recent version of Python. You can find out what version of Python you have by typing the following into a terminal window:

```
python --version
```

or, if you already have python version 3, try:

```
python3 --version
```

2. Then install boto3, a Python package for interaction with Amazon's MTurk.

```
pip install boto3
```

3. You will also need to install a few Python packages. pyyaml is a YAML parser for Python. YAML stands for "Yet Another Markup Language" or "YAML Ain't Markup Language", and is a convenient way of reading (for humans and computers) data. You will see an example of YAML syntax in a later section. In your terminal:

```
pip install pyyaml
pip install ruamel_yaml
pip install xmltodict
pip install unicodedcsv
pip install pandas
```

4. And if you're using Python 3.9 or higher additionally install the following:

```
pip install ruamel.yaml.clib
```

3.1 If you want to use boto3 through R

Finally, install pyMTurkR, an R library that allow interaction with MTurk through the boto package without requiring you to know Python. Step by step instructions are provided at <https://cran.r-project.org/web/packages/pyMTurkR/readme/README.html>. Make sure to complete all three steps on that page.

4 Link Boto to your AWS account

Now we need to link boto with your account by setting a configuration file with your AWS access key and secret key from Step 1. This can be done in a number of ways. The ultimate goal in all cases is to set three environment variables `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, and `AWS_DEFAULT_REGION` (to: `us-east-1`).

On Windows, directly set all three variables by editing your system environment variables, or you can create a credentials file in `C:\Users\USERNAME\.aws\credentials`. The latter method might be preferably in case the setting of environment variables is not permanent. On Mac or Linux machines, create a credentials file in `~/.aws` (i.e. `~/.aws/credentials`) and edit it to contain the following lines (case matters!):

```
[default]
aws_access_key_id = your_access_key_here
aws_secret_access_key = your_secret_key_here
```

Next, create and edit the config file in `~/.aws` (i.e. `~/.aws/config`):

```
[default]
region = us-east-1
```

Both the shared credentials file and the config file also supports the concept of profiles. Profiles represent logical groups of configuration. For example, for the shared credential file you might want to specify two separate profiles:

```
[default]
aws_access_key_id=foo
aws_secret_access_key=bar

[lab]
aws_access_key_id=foo2
aws_secret_access_key=bar2
```

You can then directly refer to the name of a profile when you work with `*boto3*` or `*mturkutils*`. This is convenient, for example, if you want to use your personal AWS account for debugging your experiment but then use a user linked to the lab's AWS account for the actual experiment. For more information or if you run into problems, see <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/credentials.html>.

If you're stuck on how to create the config file...

You can use your text editor of choice to create this file. If you're stuck or don't have a text editor preference, here are some instructions for how to do it using vi. This is the default text editor for all UNIX-esque systems, so if you have a Mac, you already have it. As a warning, if you haven't used vi before, these steps might seem funky and unintuitive. But vi is pretty much as fast, lightweight, and powerful as text editors come, so I highly recommend it.

At the command line, type

```
vi ~/.aws/credentials
```

This opens vi with a new file called ~/.boto. Now that you are in vi, do the following steps:

- Press i (to go into insert mode)
- Paste your credentials in the proper format
- Press Esc (to go into command mode)
- Type :wq Enter (to quit and save)

4.1 AWS shell

[AWS shell](#) is a tool that simplifies interaction with MTurk through the command line. There is a [nice introduction to AWS shell with guidelines on how to install it](#) and a [tutorial on more advanced topics](#).

4.2 If want to use boto3 through R

Open the .Renviron file. You can do this through RStudio, either through the file menu (make sure to select the option "Show hidden files" or through the terminal within RStudio). On Linux/MacOS, the file will be in your home directory. On Windows, the .Renviron file might be under C:\Users\USERNAME\Documents\. If you do not already have a .Renviron file, you can make one in RStudio (File > New File > Textfile and save it as .Renviron in the directory indicated in the previous sentences). In the .Renviron file add the following lines and save the file (case matters!):

```
AWS_ACCESS_KEY_ID=your_access_key_here  
AWS_SECRET_ACCESS_KEY=your_secret_key_here
```

Then restart R (from the Session menu in RStudio). Your AWS credentials should not be linked to boto3 everytime you use R.

5 Test it

Let's make sure everything is properly connected. To do this, we will try to connect to your MTurk sandbox using boto3. The sandbox is the testing area for HITs; only requesters (and not normal workers) can view sandboxed HITs. Other than that, the sandbox environment is identical to the environment that workers see. We will explore the sandbox in more detail in Section 6 by uploading a HIT to it.

In your command line, start python (or use whatever you typically use to run python). In your Python interpreter, type the following to connect to MTurk sandbox:

```
import boto3
session = boto3.Session()
# Alternatively, specify which profile you would like to use, e.g.:
# session = boto3.Session(profile_name='lab')

mtc = session.client('mturk')

# Alternatively, you can specify a specific region name, overriding your default profile
# endpoint = 'https://mturk-requester-sandbox.us-east-1.amazonaws.com'
# mtc = session.client('mturk', endpoint_url=endpoint, region_name='us-east-1')
print(mtc.get_account_balance())
```

You should get something like the following output (MTurk sandbox always contains \$10k of fake money):

```
{'AvailableBalance': '10000.00',
 'ResponseMetadata': {'RequestId': 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX',
 'HTTPStatusCode': 200,
 'HTTPHeaders': {'x-amzn-requestid': 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX',
 'content-type': 'application/x-amz-json-1.1',
 'content-length': '31',
 'date': 'Tue, 13 Oct 2020 19:02:08 GMT'},
 'RetryAttempts': 0}}
```

Be sure to get this working before continuing, but don't worry if you got a message telling you that your account balance is 0 (rather than 10,000.00!). Apparently, the outcome of this command has changed recently. If things are *not* working, you should see an error message that is very clearly different from the output shown above.

5.1 If you're using boto3 through R

In RStudio:

```
library(pyMTurkR)
getbalance()
```

You should get the following output (MTurk sandbox always contains \$10k of fake money):

```
[$10,000.00]
```

If you instead get an error message like the following:

```
Error in py_call_impl(callable, dots$args, dots$keywords):
RequestError: An error occurred (RequestError) when calling the GetAccountBalance operation:
To use the MTurk API, you will need an Amazon Web Services (AWS) Account.
Your AWS account must be linked to your Amazon Mechanical Turk Account.
Visit https://requestersandbox.mturk.com/developer to get started.
Check your AWS credentials.
Error in py_call_impl(callable, dots$args, dots$keywords) :
RequestError: An error occurred (RequestError) when calling the GetAccountBalance operation:
To use the MTurk API, you will need an Amazon Web Services (AWS) Account.
Your AWS account must be linked to your Amazon Mechanical Turk Account.
Visit https://requestersandbox.mturk.com/developer to get started.
```

This points to some issue with your account setup. PyMTurkR is able to reach your Amazon's IAM

(Identity and Access Management) system with your credentials, but either a) your credentials are wrong, or b) you forgot to create a requester sandbox account under Step 1, and/or c) you forgot to link the MTurk requester account to AWS under Step 1.

6 Interacting with MTurk via mturkutils

There are many ways to interact with MTurk. One convenient way is to use a collection of boto3 script written by former HLP lab manager Andrew Watts. You can get the most recent mturkutils at <https://bitbucket.org/hlplab/mturkutils/src/master/>. Alternatively, you might use the R library pyMTurkR.

This tutorial comes pre-packaged with mturkutils (linked as a [git submodule](#)), and we will use it below to upload HIT to MTurk and obtain results from MTurk. The mturkutils scripts can be found in the mturkutils/ folder. As you can read in the README.md “The original scripts use the now unsupported boto library. Some scripts have been updated to use the current boto3 and botocore libraries. Originals are in the ‘boto’ directory and updated are in ‘boto3’. Updated versions also require Python 3.” We will be using the scripts in the boto3 folder.

To learn more about any of the mturkutils scripts, call the script with the -h tag (“h” for help). In your terminal, go to the mturkutils/boto3/ folder and try, for example:

```
python loadHIT.boto3.py -h
```

Make sure that you are in the mturkutils/boto3/ folder when you run this (or provide the right path). Notice that loadHits.py takes a configuration file (our YAML file), and an optional sandbox tag. **Unless you’re ready for your experiment to go live and pay for it, always make sure to use the sandbox tag! Once a hit is posted to MTurk via boto, it is exceedingly difficult to delete it. And, in any case, you will charged for it.**