BCCP Web Scraping Course

Preparations for the Course

Julian Harke and Kevin Tran[1]

June 24, 2019, 09:30 - 12:30
June 25, 2019, 09:30 - 12:30
June 26, 2019, 14:00 - 17:00

We would like to make the course as interactive as possible. So if you would like, please bring your computers and try out the prepared scripts on your own. In order to minimize troubleshooting time during the course, here are some things that you could prepare before the first session.

1 Install a Python 3 distribution and the needed packages

As we are using Python during the course, you will need a working installation of Python (ideally Python 3.7). Installing Python can cause various complications on bad days, so doing this before the course is a good idea.

Two good Python distributions are Anaconda (Link) and Miniconda (Link). Anaconda has the advantage that it has many pre-installed packages. Miniconda has the advantage that it only comes with the most necessary packages and therefore requires less disk space. You can then easily install the packages you need manually. If you decide on one of these distributions, you can follow these steps:

1. Download the installer Download the Python 3.7 installer for your operation system for Anaconda (Link) or Miniconda (Link). On MacOS, best choose the .pkg installer of Miniconda.

2. Run the installer Run the downloaded file and follow the instructions.

(3. Create a Python environment) This step is not absolutely necessary but we do recommend that you create a Python environment for the course. A Python environment is like a virtual separate Python installation with only the packages you specify. You could also just install all needed packages in your root environment but this may lead to some conflicts sometimes.

[1] Julian is a Research Fellow at the WZB and a PhD student at Vrije Universiteit Amsterdam. Kevin is a PhD student at the DIW Graduate Center and Technische Universität Berlin. Both are members of the Berlin Centre for Consumer Policies (BCCP).

In order to create a Python environment proceed as follows:

1. Open the Terminal/Command Prompt.[2] Alternatively, you can also use the Anaconda Prompt.
2. Create a Python 3.7 environment "webscraping_course" (you can call it however you like):

   ```
   conda create -n webscraping_course python=3.7
   ```

3. Activate the new environment:

   ```
   conda activate webscraping_course
   ```

   If you see a `(webscraping_course)` at the start of the line, everything worked out fine.

4. Install the needed packages Once you are in the environment you would like to use for the course, you need to install the packages that we will be using. You can either do this manually or using the requirements.txt file that we provided. The requirements.txt is a text file that contains a list of the packages that we need for the course.

- If you would like to use the requirements.txt, proceed as follows:
  1. Open the Terminal/Command Prompt/Anaconda Prompt.
  2. Change the current directory to the directory where you saved the requirements.txt. For example:

     ```
     cd C:\Users\kevin\Downloads
     ```

     if you saved the file in `C:\Users\kevin\Downloads`
  3. Install the packages using

     ```
     pip install -r requirements.txt
     ```

- If you would like to install the packages manually, proceed as follows:
  1. Open the Terminal/Command Prompt/Anaconda Prompt.
  2. Install the packages using

     ```
     pip install requests beautifulsoup4 selenium pandas lxml jupyter tweepy
     world_bank_data pandas-datareader
     ```

2 Installing a browser and downloading the corresponding driver

For the part on browser automation, you will need to have an internet browser installed and downloaded a corresponding browser driver. In principle, you can use any browser you want given that a driver exists for it.

In the example script, I will use Google's Chrome browser. Doing the same will reduce the likelihood of browser-specific errors. For the course, please prepare the following:

1. Install and locate a browser Decide on a browser and install it, if you do not already have it on your computer. I will use Google Chrome. Further, find out where the program file of your browser is located on your computer. You will need this to run the script on browser automation.

[2] Windows+R, then type "cmd", and press ENTER on Windows; cmd+space, then type "terminal", and press ENTER on MacOS.

2. Download a fitting browser driver Download a driver that fits your browser. Links to the drivers for Chrome, Edge, Firefox, and Safari can be found here: `https://selenium-python.readthedocs.io/installation.html#drivers`. Also make sure that the driver version fits your installed browser version. The easiest way to do so is to install/update to the most current version of the browser and download the most current version of the driver. Save the driver somewhere where you will find it later, you will need the location of the file to run the script on browser automation.

3 Twitter API authentification

If you wish to work with twitter data you will need the consumer_key, consumer_secret, access_token, and access_token_secret. Please get the keys beforehand. Otherwise you will not be able to run the twitter code yourself.

1. Visit following Tweepy tutorial
2. Follow the three steps in the section “Creating Twitter API Authentication Credentials” to obtain the keys.
3. Save them somewhere, so you can use them in the course.

4 Feel free to ask

If you try any of these steps and get an error, feel free to contact any of us (julian.harke@wzb.eu or ktran@diw.de) and we will try to help as well as possible.