

Goal:

We run a sports betting organization that automates all our bets from the collection of stats, to the placement of bets on various online sports books. As we've grown, our needs have outpaced our ability to build the scripts and we need some help! Of course, none of the sports books have accessible API's so we have to use selenium and bs4 to screen scrape all of our data needs.

For this particular project we are looking to improve our bet reconciliation. Each book provides a history of bets, which we need to scrape and compare to our ledger. We are looking for someone(s) to build a script for each book that will allow us to update the data everyday.

URLs:

bookmaker.eu

williamhill.com

sportsbook.fanduel.com

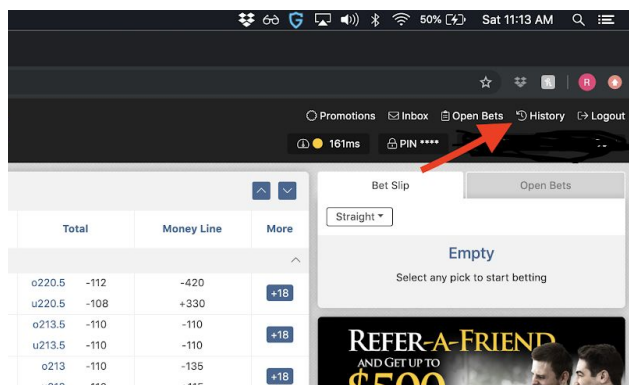
draftkings.com/sportsbook

playsugarhouse.com

Where is the Data:

Each book has the information in different places.

For example - bookmaker.eu:



Output:

The script should print and return a pandas Dataframe with the following columns:

- "id" (str) - a unique identifier provided by the book, either explicitly or within the html tags
- "book" (str) - the sports book the bet is pulled from
- "sport" (str) - for the purposes of this project, it will be nba or mlb

- “period” (str) - the game period for which the bet was made. For nba this is ‘q1’, ‘q2’, ‘q3’, or ‘q4’ and for mlb it is either ‘f5’(first five innings) or full game
- “bnet_type” (str) - this will be either ‘total’ or ‘spread’
- “position” (str) - if the bet_type is ‘spread’, this is a team name. If the bet_type is ‘total’ this is over - ‘o’, or under - ‘u’
- “line” (float) - this is always a positive or negative number that represents the spread or total
- “odds” (int) - this is a positive or negative number (int) that represents the odds of the bet
- “team1” (str) - the first team in the matchup (order doesn’t matter)
- “team2” (str) - the second team in the matchup
- “at_risk” (float) - this was the amount wagered
- “to_win” (float) - this is the amount won or lost in the wager
- “date_placed” (str) - this is a date in format “2020-02-08” (year-month-day) representing the date the bet was placed
- “result” (str) - results can either be won, lost, or pushed (tied), represented by ‘w’, ‘l’ or ‘p’

Tools We Use:

Here are the tools we use for these types of scripts - familiarity with them is nice. If you think other frameworks / tools would be better we’re open to hearing about them.

- Python3
- Selenium
- BS4
- Pandas

Code Sample:

The function used for logging into bookmaker.eu:

```
def login(self, auth):

    driver = self.driver

    url = 'https://www.bookmaker.eu/'
    name = os.environ['bm_username']
    pw = os.environ['bm_password']

    driver.get(url)

    if auth == False:
        return "done"
```

```

        try:
            print(self.time_check()+' filling namefield')
            name_field =
WebDriverWait(driver,20).until(EC.element_to_be_clickable((By.ID,'account'))))
            time.sleep(5)
            name_field.send_keys(name)
        except TimeoutException:
            print('couldnt load username field')

        try:
            print(self.time_check()+' filling pwfield')
            pw_field =
WebDriverWait(driver,20).until(EC.element_to_be_clickable((By.ID,'password'))))
            pw_field.send_keys(pw)
        except TimeoutException:
            print('couldnt load password field')

        try:
            print(self.time_check()+' clicking login button')
            login_btn = WebDriverWait(driver,
10).until(EC.element_to_be_clickable((By.CSS_SELECTOR,
'input.btnCta_light.gradLightGray'))))
            login_btn.click()
        except TimeoutException:
            print("Loading took too much time!")

```

What we'll Need You to Do:

1. Create a test account for each sportsbook
2. Fanduel, Sugarhouse, and Draftkings each have geo-restriction. We operate in New Jersey, US for those sites so you'll need to make sure you have access to create bets, and that the behavior is the same as in New Jersey for those sites.
3. Place some test bets - use the minimum bet amount - \$1 for example:
 - a. nba 3rd quarter spread
 - b. nba 1st quarter total under
 - c. nba 2nd quarter total over
4. Write a script that will grab individual bet data and output a dataframe with all bets- it should take a start date and end date as parameters and iterate through each date between.