TX Dallas Parser Documentation

Release 0.2

Jay C.

Jan 27, 2020

This is a Python package for parsing HTML pages retrieved from the Texas Dallas County Felony and Misdemeanor Courts Case Information. Since the county website strictly monitors server activity, this package is meant to be used only after the user has collected HTML files for their use. Put differently, I will not be sharing any code that can be used to collect these raw HTML files.

CHAPTER

ONE

OUTPUT

The *dallasparser* package will generate a maximum of 15 *.xlsx* files for each data table found from the county website:

Table Name	Output
Appeals	*appeals.xlsx*
Bonds	*bonds.xlsx*
Bond COMMENTS	*bond_comments.xlsx*
Charges	*charges.xlsx*
Competency Data	*competency_data.xlsx*
Dispositions	*dispositions.xlsx*
General Comments	*general_comments.xlsx*
General Comments WS Date	*general_comments_ws_date.xlsx*
Judicial Information	*judicial_information.xlsx*
Motions	*motions.xlsx*
Names	*names.xlsx*
Payments	*payments.xlsx*
Probation Revocation	*probation_revocation.xlsx*
Reduced/Enhanced Charges	*reduced_enhanced_charges.xlsx*
Sets and Passes	*sets_and_passes.xlsx*

CHAPTER

TWO

INSTALLATION

Source:

```
$ git clone https://github.com/jaycatsby/tx_dallas_court_parser.git
$ cd tx_dallas_court_parser
$ python setup.py install
```

PyPI:

```
$ pip install dallasparser
```

CHAPTER

THREE

USAGE

A. CLI:

```
$ dallasparser [-h] [-i INPUT] [-o OUTPUT]

optional arguments:
        -h, --help            show this help message and exit
        -i INPUT, --input INPUT
                                                                                    ␣
↪              absolute path of HTML folder
        -o OUTPUT, --output OUTPUT
                                                                                    ␣
↪              absolute path of XLSX output files
```

B. Module:

```
from dallasparser.parser import TXDallasParser
parser = TXDallasParser(html_path, xlsx_path)
parser.run()
```

CHAPTER

FOUR

SCRIPTS

4.1 cli.py

4.1.1 CLI

Command-line script for running TXDallasParser.

cli.**main**()

Main function to run when parsing using CLI.

cli.**parse_arg**()

Argument parser for CLI usage.

4.2 parser.py

4.2.1 Parser

Parser for extracting relevant information from Texas Dallas County Felony and Misdemeanor Courts using regular expressions.

class dallasparser.parser.**TXDallasParser**(*input_path=None*, *output_path=None*)

Bases: object

Main parser class.

Parameters

- **input_path** (*str*) – Absolute path of HTML folder.
- **output_path** (*str*) – Absolute folder path of XLSX output files.

COLUMN_ORDER = {'appeals': ['da_case_id', 'jd_case_id', 'appeal_id', 'ct_disp_no',

Column order for final exported XLSX files for each table. To modify, see *utils.py*.

extract_tables(*trs*)

Method for separating all of the HTML elements to appropriate list for a given table.

Parameters **trs** (*list*) – List of <tr> objects excluding table header elements

Return type tuple

get_appeals(*appeal_trs*, *da_case_id*, *jd_case_id*)

Extract *Appeals* Table.

Parameters

- **appeals_trs** (*list*) – List of <tr> elements in *Appeals* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_bond_comments (*bond_comment_trs*, *da_case_id*, *jd_case_id*)
Extract *Bond Comments* Table.

Parameters

- **bond_comment_trs** (*list*) – List of <tr> elements in *Bond Comments* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_bonds (*bonds_trs*, *da_case_id*, *jd_case_id*)
Extract *Bonds* Table.

Parameters

- **bonds_trs** (*list*) – List of <tr> elements in *Bonds* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_charges (*charge_trs*, *da_case_id*, *jd_case_id*)
Extract *Charges* Table.

Parameters

- **charges_trs** (*list*) – List of <tr> elements in *Charges* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_competency_data (*comp_trs*, *da_case_id*, *jd_case_id*)
Extract *Competency Data* Table.

Parameters

- **comp_trs** (*list*) – List of <tr> elements in *Competency Data* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_dispositions (*disp_trs*, *da_case_id*, *jd_case_id*)
Extract *Dispositions* Table.

Parameters

- **disp_trs** (*list*) – List of <tr> elements in *Dispositions* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_general_comments (*comment_trs*, *da_case_id*, *jd_case_id*)
Extract *General Comments* Table.

Parameters

- **comment_trs** (*list*) – List of <tr> elements in *General Comments* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.

- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_general_comments_ws (*comment_ws_trs*, *da_case_id*, *jd_case_id*)
Extract *General Comments WS Dates* Table.

Parameters

- **comment_ws_trs** (*list*) – List of <tr> elements in *General Comments WS Date* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_judicial_information (*judicial_trs*, *da_case_id*, *jd_case_id*)
Extract *Judicial Information* Table.

Parameters

- **judicial_trs** – List of <tr> elements in *Judicial Information* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type dict

get_motions (*motion_trs*, *da_case_id*, *jd_case_id*)
Extract *Motions* Table.

Parameters

- **motions_trs** (*list*) – List of <tr> elements in *Motions* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_names (*names_trs*, *da_case_id*, *jd_case_id*)
Extract *Names* Table.

Parameters

- **names_trs** (*list*) – List of <tr> elements in *Names* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_payments (*payment_trs*, *da_case_id*, *jd_case_id*)
Extract *Payments* Table.

Parameters

- **payment_trs** (*list*) – List of <tr> elements in *Payments* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_probation_revocation (*prov_revoc_trs*, *da_case_id*, *jd_case_id*)
Extract *Probation Revocation* Table.

Parameters

- **prob_revoc_trs** (*list*) – List of <tr> elements in *Probation Revocation* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_reduced_enhanced (*red_enh_trs*, *da_case_id*, *jd_case_id*)

Extract *Reduced/Enhanced Charges* Table.

Parameters

- **red_enh_trs** (*list*) – List of <tr> elements in *Reduced/Enhanced Charges* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

get_sets_and_passes (*sets_trs*, *da_case_id*, *jd_case_id*)

Extract *Sets and Passes* Table.

Parameters

- **sets_trs** (*list*) – List of <tr> elements in *Sets and Passes* section.
- **da_case_id** (*str*) – DA Case ID used for linkage.
- **jd_case_id** (*str*) – Judicial Case ID used for linkage.

Return type list

parse (*html_fn*)

Main method to run the parser. This method takes in an HTML file name and loads it as a *BeautifulSoup* object. Afterwards, it calls *extract_tables* method to find all of the relevant HTML elements before checking for each table.

Parameters **html_fn** (*str*) – Filename of the HTML page to parse.

Return type tuple

run ()

Main method to call for parsing HTML files. Iterates *self.input_path* to find all HTML files and for each file found, calls the *parse* method.

4.3 regex.py

This module offers regular expressions for each information table.

4.4 utils.py

4.4.1 Utils

This module offers general convenience and utility functions for dealing with parsed data.

`utils.APPEALS_HEADERS = ['da_case_id', 'jd_case_id', 'appeal_id', 'ct_disp_no', 'date_ap`

Column order of *appeals.xlsx*

`utils.BONDS_HEADERS = ['da_case_id', 'jd_case_id', 'bond_id', 'date_bond_set', 'amt', 't`

Column order of *bonds.xlsx*

utils.**BOND_COMMENTS_HEADERS = ['da_case_id', 'jd_case_id', 'comment_id', 'date', 'commen**
Column order of *bond_comments.xlsx*

utils.**CHARGES_HEADERS = ['da_case_id', 'jd_case_id', 'charge_id', 'name_raw', 'offense_c**
Column order of *charges.xlsx*

utils.**COMPETENCY_HEADERS = ['da_case_id', 'jd_case_id', 'competency_id', 'hearing_date',**
Column order of *competency_data.xlsx*

utils.**DISPOSITIONS_HEADERS = ['da_case_id', 'jd_case_id', 'disp_id', 'ct_disp_no', 'verc**
Column order of *dispositions.xlsx*

utils.**GC_HEADERS = ['da_case_id', 'jd_case_id', 'comment_id', 'comment', 'date', 'last_u**
Column order of *general_comments.xlsx*

utils.**GC_WS_DATE_HEADERS = ['da_case_id', 'jd_case_id', 'comment_id', 'comment', 'commen**
Column order of *general_comments_ws_date.xlsx*

utils.**JUDICIAL_HEADERS = ['da_case_id', 'jd_case_id', 'name_raw', 'race', 'sex', 'dob',**
Column order of *judicial_information.xlsx*

utils.**MOTIONS_HEADERS = ['da_case_id', 'jd_case_id', 'motion_id', 'motion_filed', 'motic**
Column order of *motions.xlsx*

utils.**NAMES_HEADERS = ['da_case_id', 'jd_case_id', 'name_id', 'associated_name', 'name_r**
Column order of *names.xlsx*

utils.**PAYMENTS_HEADERS = ['da_case_id', 'jd_case_id', 'payment_id', 'ct_disp_no', 'date_**
Column order of *payments.xlsx*

utils.**PROB_REVOC_HEADERS = ['da_case_id', 'jd_case_id', 'ct_disp_no', 'verdict_date', 'v**
Column order of *probation_revocation.xlsx*

utils.**RED_ENH_HEADERS = ['da_case_id', 'jd_case_id', 'red_enh_id', 'desc', 'comt', 'typ'**
Column order of *reduced_enhanced_charges.xlsx*

utils.**SETS_HEADERS = ['da_case_id', 'jd_case_id', 'sp_id', 'set_for_date', 'set_for_time**
Column order of *sets_and_passes.xlsx*

utils.**clean_val**(*val*)
Returns cleaned value after replacing underscores (‘_’) and asterisks (‘*’) with an empty whitespace.

Parameters val (*str*) – Raw parsed string

Returns A `str` object after removing underscores and asterisks

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX