**Unit Testing**

|  |  |
| --- | --- |
| Function | **convert\_date\_to\_timestamp** |
| Tested By | Rajeev Shukla |
| Details | This function uses mktime() function of python time library that converts local date-time into Unix time (also known as POSIX time or epoch time), epoch defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC) Thursday, 1, January 1970. |
| Input | date\_string (datetime in string format) |
| Output | Timestamp (seconds since the epoch for localtime) |

**Test Code:**

**import** datetime  
**import** time  
  
test\_date=**"01/01/1970 00:00"  
  
def** convert\_date\_to\_timestamp(date\_string):  
 date\_strip = **""** *# First try if the format is %m/%d/%Y %H:%M if not try the other format* **try**:  
 date\_strip = datetime.datetime.strptime(date\_string, **"%m/%d/%Y %H:%M"**)  
 **except**:  
 **try**:  
 date\_strip = datetime.datetime.strptime(date\_string, **"%m/%d/%Y %H:%M:%S"**)  
 **except** ValueError **as** ve:  
 **print** (**"Cannot acess the format of the data"**, ve)  
  
 time\_stamp = time.mktime(date\_strip.timetuple())  
 **return** time\_stamp  
  
result= convert\_date\_to\_timestamp(test\_date)  
**print "Timestamp(seconds since the epoch for localtime):"**,result

**Test Case1:**

Input:

“3/27/2017 1:43:00 PM”

Output(Error):

(('Cannot acess the format of the data', ValueError('unconverted data remains: PM',))

Justification:

Code executes with error because input date string should be in “mm/dd/YYYY hh:mm” or “mm/dd/YYYY hh:mm:ss” format

**Test Case2:**

Input:

“3/27/2017 01:43”

Output(Successful without any errors or warnings):

Timestamp(seconds since the epoch for localtime): 1490596980.0

Justification:

Function returns total number of seconds that have elapsed from “1/1/1970 00:00” to “3/27/2017 1:43” local-time.

**Test Case3:**

Input:

“1/1/1970 00:00”

Output(Successful without any errors or warnings):

Timestamp(seconds since the epoch for localtime): **21600**.0

Justification:

Function returns total number of seconds that have elapsed from “1/1/1970 00:00” to “1/1/1970 00:00” local-time.

Here, it looks wrong output as date is “1/1/1970 00:00” and epoch time started counting at “1/1/1970 00:00”, so output should be 0 seconds.

But **output is correct** because mktime converts local date-time into Unix time(number of seconds) where as actual Unix time counting is based on Coordinated Universal Time (UTC) date-time.

Here our input is Local Date and Time January 1, 1970 @ 12:00:00 am and if we convert out local Date-Time in UTC Date and Time than it will be January 1, 1970 @ 6:00:00 am which is 6hours after Unix Time counting got started, So output is 6hours = 3600\*6 = 21600 seconds.

**Test Case4:**

Input:

“13/27/2017 01:43”

Output(Error):

('Cannot acess the format of the data', ValueError("time data '13/27/2017 1:43' does not match format '%m/%d/%Y %H:%M:%S'",))

Justification:

Code executes with error because input date string should be in “mm/dd/YYYY hh:mm” or “mm/dd/YYYY hh:mm:ss” format but here month is 13, which is invalid month number.

**Test Case5:**

Input:

“1/1/1968 01:43”

Output(Error):

“Traceback (most recent call last):

File "C:/Users/Admin/PycharmProjects/netcool\_test/netcool.py", line 20, in <module>

result= convert\_date\_to\_timestamp(test\_date)

File "C:/Users/Admin/PycharmProjects/netcool\_test/netcool.py", line 17, in convert\_date\_to\_timestamp

time\_stamp = time.mktime(date\_strip.timetuple())

**OverflowError: mktime argument out of range**”

Justification:

Code executes with error because input date string year is 1968 and mktimefunction calculates seconds only from January 01, 1970 so year is out of range for mktime function.