In [2]:
```
#Author: Suryoday Basak
#suryodaybasak.info
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
plt.style.use('ggplot')
mpl.rcParams['figure.figsize'] = (10,8)
```

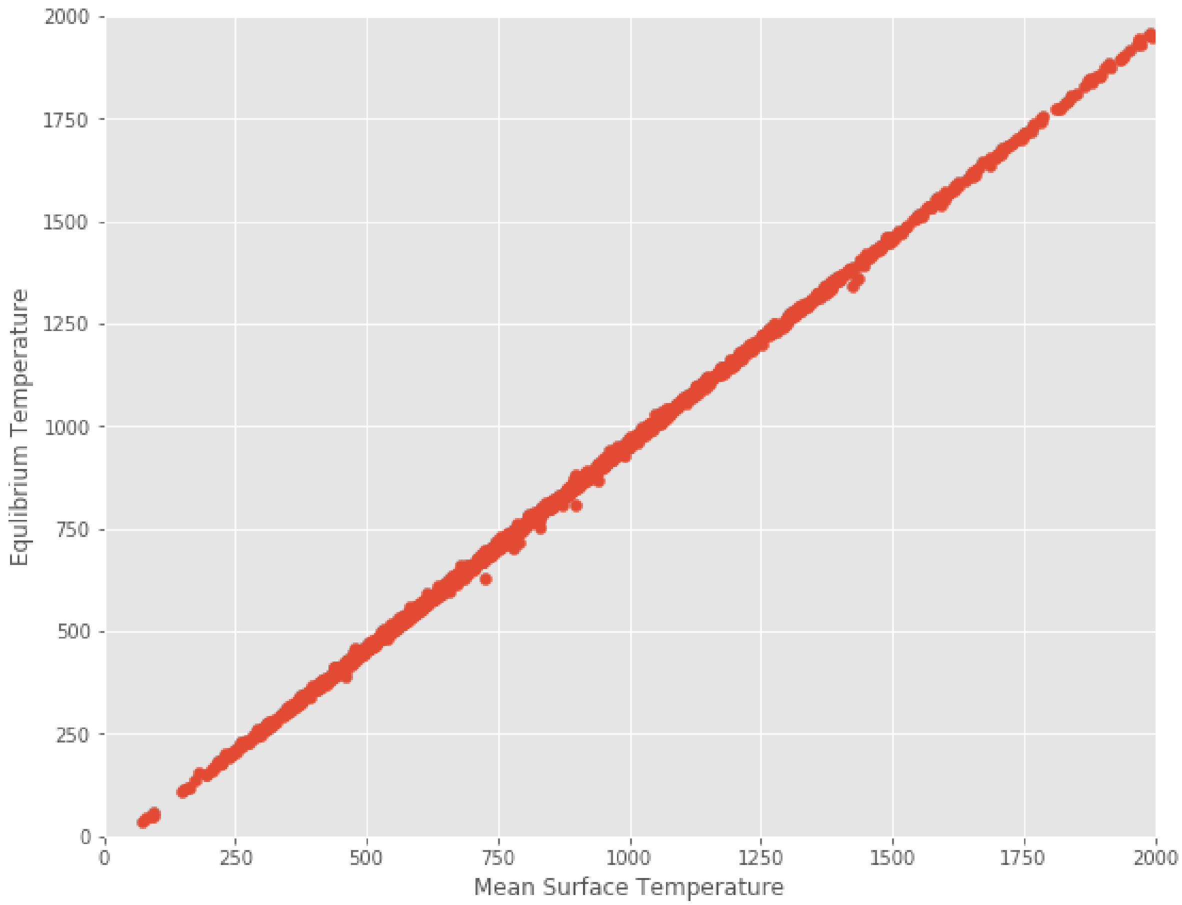
In [4]:
```
#Reading the data
df = pd.read_csv('../datsets/physics/exoplanet-temp.csv')
df = df.dropna() #For convenience, we drop any rows with any NaN entri
es
print(df)
```

	P. Teq Mean (K)	P. Ts Mean (K)
38	1790.4	1833.5
40	1054.4	1095.7
50	1068.0	1102.1
57	498.9	546.5
65	1177.4	1221.5
88	1664.6	1706.5
126	510.7	551.3
127	431.1	475.2
130	1244.9	1287.2
131	1015.0	1064.4
132	148.9	193.9
133	1359.1	1432.1
134	1232.2	1269.4
136	974.0	1014.8
139	1184.0	1224.0
140	857.8	907.3
141	1124.4	1168.6
142	846.7	892.0
143	523.9	568.3
147	368.0	415.6
148	549.8	595.1
149	541.4	589.0
150	711.4	755.4
151	605.9	643.5
153	793.3	829.3
154	955.7	993.7
158	671.0	708.5
159	576.1	618.2
160	513.9	555.0
162	432.0	473.0
...	...	...
3501	60.0	95.9
3517	37.8	72.9
3538	229.3	263.9
3540	882.6	898.9
3541	641.2	681.5
3542	567.2	607.2
3543	3053.8	3117.8
3544	111.2	147.8
3558	280.0	315.1
3572	646.5	683.3
3573	475.1	514.1
3574	343.0	382.5
3575	282.4	322.9
3576	180.6	223.4
3577	575.2	611.4
3578	424.2	460.6
3581	364.9	396.5
3582	311.8	347.9
3583	262.7	292.4
3584	229.1	260.4
3585	199.7	229.7
3586	181.1	216.1
3587	153.2	181.8
3658	1949.6	1994.6
3763	425.6	462.1

```
3764              276.4             315.8
3765              117.9             161.4
3779              442.6             474.8
3780              382.0             415.4
3781              331.9             365.9

[1724 rows x 2 columns]
```

In [5]:
```
#Plot the data here to visualize the trend
plt.scatter(df['P. Ts Mean (K)'], df['P. Teq Mean (K)'])
#plt.scatter(df['M (g)'], df['T (s)'])
plt.xlabel('Mean Surface Temperature')
plt.ylabel('Equlibrium Temperature')
plt.xlim(0, 2000)
plt.ylim(0, 2000)
plt.show()
#plt.clf()
```



In [6]:
```
n = df['P. Ts Mean (K)'].count()                    #Number of samples
p = np.sum(np.square(df['P. Ts Mean (K)']))         #The sum of x^2
q = df['P. Ts Mean (K)'].sum()                      #The sum of x
r = np.sum(df['P. Ts Mean (K)']*df['P. Teq Mean (K)'])    #The sum of
 the product of x and y
s = df['P. Teq Mean (K)'].sum()                      #The sum of y^2

#Print all of the above
print("The number of samples is:\t", n)
print("The sum of Ts^2 is:\t\t", p)
print("The sum of Ts is:\t\t", q)
print("The sum of Ts*Teq is:\t\t", r)
print("The sum of Teq is:\t\t",s)
```

```
The number of samples is:        1724
The sum of Ts^2 is:              1675245193.47
The sum of Ts is:                1526805.7
The sum of Ts*Teq is:            1613079581.14
The sum of Teq is:               1455489.0
```

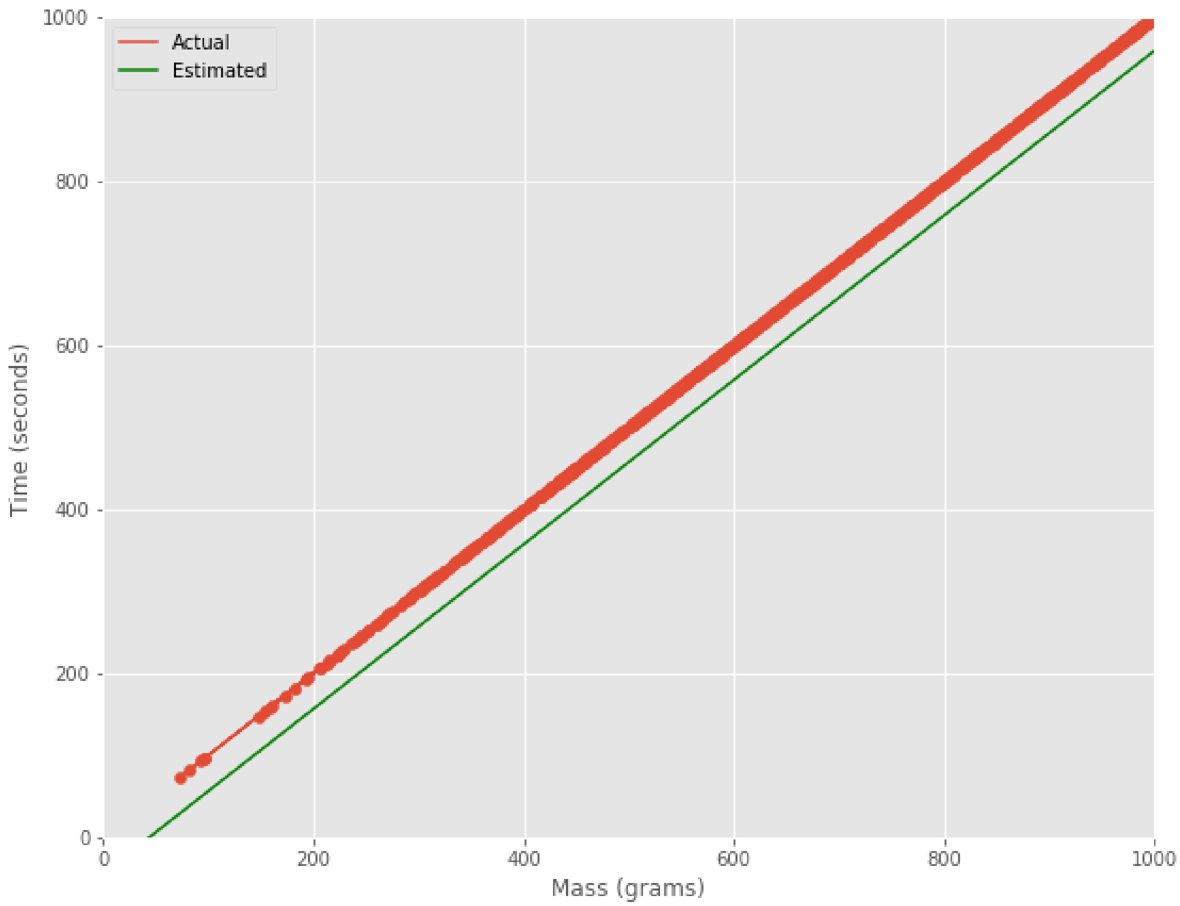
In [7]:
```
m = (1/((n*p) - (q**2)))*((n*r) - (q*s))    #The slope of the line
c = (1/((n*p) - (q**2)))*((p*s) - (r*q))    #The y-intercept of the li
ne

print("The slope of the estimated line is:\t\t", m)
print("The y-intercept of the estimated line is:\t", c)
```

```
The slope of the estimated line is:             1.00307588543205
The y-intercept of the estimated line is:       -44.09105534234351
```

In [12]:
```
#To visualize the estimated line, create an x-vs-y set using m and c
x = [x/10 for x in range (0, 80000)]
y = [m*xi + c for xi in x]

#Plot again to visualize how the estimated line fairs against the orig
inal data
orig, = plt.plot(df['P. Ts Mean (K)'], df['P. Ts Mean (K)'],label = "A
ctual" )
plt.scatter(df['P. Ts Mean (K)'], df['P. Ts Mean (K)'], label = "Actua
l")
est, = plt.plot(x, y, label = "Estimated", color='g')
plt.xlabel('Mass (grams)')
plt.ylabel('Time (seconds)')
plt.legend(handles=[orig, est])
plt.xlim(0, 1000)
plt.ylim(0, 1000)
plt.show()
#plt.clf()
```



In [13]:

```
#Finding the error
error = 0.0
for index, row in df.iterrows():
    error += ((m*row['P. Ts Mean (K)'] + c) - row['P. Teq Mean (K)'])*
*2 #(Estimated - original)^2
error/=n

print("The mean squared error is:\t\t", error)
print("The root means squared error is:\t", error**(0.5))
```

```
The mean squared error is:              42.14731137295646
The root means squared error is:        6.492096069295067
```