In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sts
%matplotlib inline
```

Выборка размера $N = 10000$ из $Exp(\theta)$, где $\theta = 1$

In [2]:
```
N = 10000
t = 1
exp_rv = sts.expon(t)
sample = exp_rv.rvs(N)
```

В массиве хранятся элементы, соответсвующие выборочному k-му моменту и оценке, для каждого $n \leq N$

$$first \sim \bar{X^k}$$

$$result \sim \sqrt[k]{\frac{k!}{\bar{X^k}}}$$

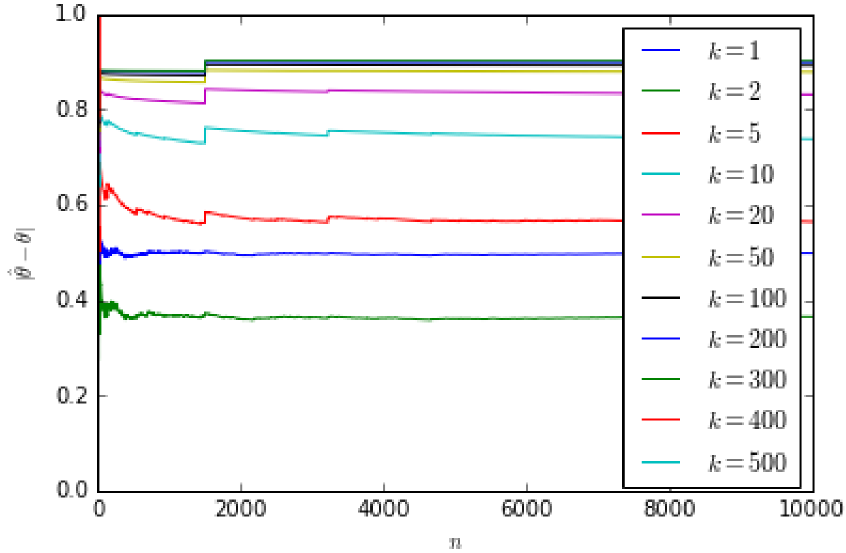
In [27]:
```
def f(k, sample):
    avrg = float(sample[0])**k

    first = np.array([avrg])
    for x in xrange(1,10000):
        avrg = (avrg*x + sample[x]**k)/(x+1)
        first = np.append(first, avrg)

    result = ((np.prod(k))/first)**(1.0/k)
    return result
```

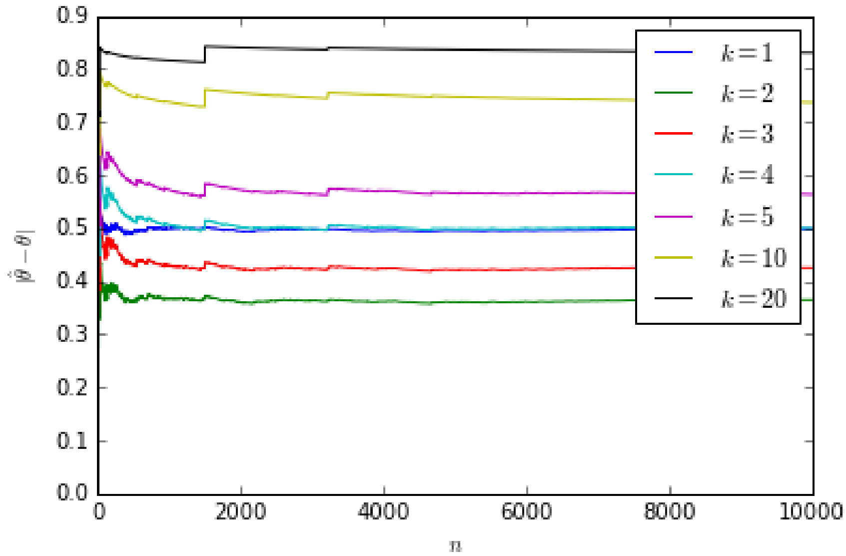
Построение графиков модуля разности оценки и θ

In [37]:
```
x = np.linspace(0, N, N)
plt.plot(x, np.abs(f(1, sample) - t), label = '$k = 1$')
plt.plot(x, np.abs(f(2, sample) - t), label = '$k = 2$')
plt.plot(x, np.abs(f(5, sample) - t), label = '$k = 5$')
plt.plot(x, np.abs(f(10, sample) - t), label = '$k = 10$')
plt.plot(x, np.abs(f(20, sample) - t), label = '$k = 20$')
plt.plot(x, np.abs(f(50, sample) - t), label = '$k = 50$')
plt.plot(x, np.abs(f(100, sample) - t), label = '$k = 100$')
plt.plot(x, np.abs(f(200, sample) - t), label = '$k = 200$')
plt.plot(x, np.abs(f(300, sample) - t), label = '$k = 300$')
plt.plot(x, np.abs(f(400, sample) - t), label = '$k = 400$')
plt.plot(x, np.abs(f(500, sample) - t), label = '$k = 500$')
plt.ylabel('$|\hat{\\theta} - \\theta|$')
plt.xlabel('$n$')
plt.legend()
plt.show()
```

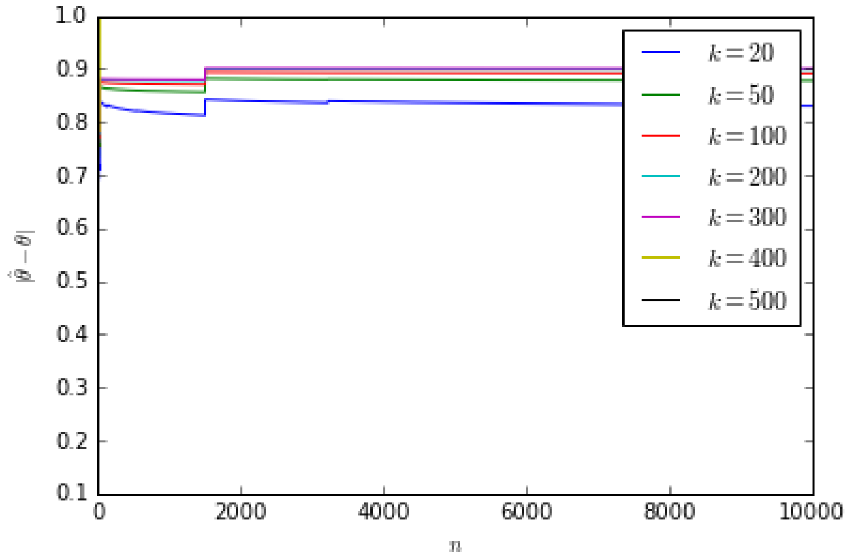


In [38]:

```
x = np.linspace(0, N, N)
plt.plot(x, np.abs(f(1, sample) - t), label = '$k = 1$')
plt.plot(x, np.abs(f(2, sample) - t), label = '$k = 2$')
plt.plot(x, np.abs(f(3, sample) - t), label = '$k = 3$')
plt.plot(x, np.abs(f(4, sample) - t), label = '$k = 4$')
plt.plot(x, np.abs(f(5, sample) - t), label = '$k = 5$')
plt.plot(x, np.abs(f(10, sample) - t), label = '$k = 10$')
plt.plot(x, np.abs(f(20, sample) - t), label = '$k = 20$')
plt.ylabel('$|\hat{\\theta} - \\theta|$')
plt.xlabel('$n$')
plt.legend()
plt.show()
```



In [39]:
```
x = np.linspace(0, N, N)
plt.plot(x, np.abs(f(20, sample) - t), label = '$k = 20$')
plt.plot(x, np.abs(f(50, sample) - t), label = '$k = 50$')
plt.plot(x, np.abs(f(100, sample) - t), label = '$k = 100$')
plt.plot(x, np.abs(f(200, sample) - t), label = '$k = 200$')
plt.plot(x, np.abs(f(300, sample) - t), label = '$k = 300$')
plt.plot(x, np.abs(f(400, sample) - t), label = '$k = 400$')
plt.plot(x, np.abs(f(500, sample) - t), label = '$k = 500$')
plt.ylabel('$|\hat{\\theta} - \\theta|$')
plt.xlabel('$n$')
plt.legend()
plt.show()
```



Вывод

Лучше всего приближает значение параметра θ оценка при $k = 2$. С ростом k оценка ухудшается, но отклонение не превышает 1.