

Note: if the server stops then command will not work
http error connection error

Views concepts:

Two types of views support in Django

1. Function based view (fbv)

2. class based view (cbv)

* fbv created by def keyword followed by view name and it
the request object as a parameter

def vn(request)

* cbv create by class keyword followed by class name and
by predefined

class cn(View):

* cbv - creating by using url like
url ('home/', views.N.as_view())

Note :-

when creating the class based url's if you are forgetting `as_view()`, along with view name. Then django internally consider this view is developed by using 'def' keyword as a function based view but not considering as a class based view

* then finally django is looking for fbr syntax we but created cbv syntax

Our user defined class based view (cbv) is the child class of `View` class, it is a base class for views this view class present in 'django.views.generic' module

fbr vs cbv :-

if we are using function based views if you want to send the get request then one function you have to do.

if we are sending post request then another functionality you have to do

for ex :-

```
def vn(req):
```

```
    if request.method == 'GET'
```

```
        <do this Get activity>
```

```
    elif request.method == 'POST':
```

```
        <do this post activity>
```

here we have to check which request is coming from partner application or from the browser like GET, POST, PUT, DELETE etc

this type of (browser) burden if there is 'fbr' but in 'cbv' no such type of problems

The 'cbv' are very simple and easy to developing the code 'cbv' are providing the code responsibility means creating one time using 'n' number of time reducing the duplicate code

for ex:-

if we developing the views by using the 'cbv' then we 'reuse' these views into an other class also by importing the 'classes'
if 'cbv' if user sending request is 'GET' then 'GET' method logic executing.

if user sending 'post' then post method logic executing so here the corresponding method only will be executing directly

```
class classname (view):
```

```
    def get (self, request):
```

```
        <do this acting>
```

```
    def post (self, request):
```

```
        <do this acting>
```

In cbv developer no need to check which type of method is coming from partner

our class can check it self which request is coming from partner