

instalation plone backend step by step			
create folder----->mkdir new_project			
cd new_project			
create buildout .cfg ----->touch buildout.cfg			
And add this configuration			
<pre> [buildout] extends = /root/Vollabor/versions.cfg parts = instance [instance] recipe = plone.recipe.zope2instance eggs = Plone plone.volto user = admin:admin </pre>			
put requirments.txt and versions.cfg file in new_project directory			
create virtual envirment in this directory----->python -m venv .			
go to virtual envirment ----->source ./bin/activate			
install requirments ----->pip install -r requirments.txt			
run buildout.cfg -----> ./bin/buildout			
run app ----->./bin instance start			
go to 127.0.0.1:8080			
create addon and content_type steps			
plonecli create addon source/addon_name			
plonecli add content_type			
python setup.py install			
instance restart			
uninstall addon and reinstall addon on frontend			
create addon and create view for sum 2 numbers			
plonecli create addon source/addon_name			
plonecli add view			
edit class (my_view) in my_view.py file			

my_view file content :			
<pre> # from vollabor.app import _ from Products.Five.browser import BrowserView from zope.interface import Interface # from Products.Five.browser.pagetemplatefile import ViewPageTemplateFile class IMyView(Interface): """ Marker interface for IMyView """ class MyView(BrowserView): # If you want to define a template here, please remove the template from # the configure.zcml registration of this view. # template = ViewPageTemplateFile('my_view.pt') def __init__(self, context, request): self.context = context self.request = request def __call__(self): num1=self.request["num1"] num2=self.request["num2"] #return "Hello world. You are rendering this view at the context of %s" % self.context return int(num1)+int(num2) #def __call__(self): # # Implement your own actions: # # return self.index() </pre>			
python setup.py install			
instance restart			
uninstall addon and reinstall addon on frontend			
Documentions			

for create view and get request in plone :			
https://docs.plone.org/develop/plone/views/browserviews.html			
https://docs.plone.org/develop/plone/serving/http_request_and_response.html			
https://docs.plone.org/develop/plone/views/browserviews.html !			
forcreate content type : chapter 18 and 19			
https://training.plone.org/mastering-plone/instructions.html			
for instalations :			
nginx, Frontend, Backend, PostgreSQL container example – Install – Containers – Examples of Plone 6 using containers — Plone Documentation v6.0			
https://www.cybrosys.com/blog/how-to-install-plone-6-on-ubuntu-2204-lts-with-volto-frontend			
8. Set up Plone for the Training – Mastering Plone 6 Development — Plone Training 2023 documentation			
(141) How to Deploy a Plone Cookiecutter Project - YouTube			
collective/cookiecutter-plone-starter: Cookiecutter Plone Starter is a framework for jumpstarting Plone 6 projects quickly.			
for rest api :			
Introduction – REST API — Plone Documentation v6.0			
https://plonerestapi.readthedocs.io/en/latest/usage/exploring.html			