

# **Application Solution Template Webserver on MELSEC iQ-R C Intelligent Function Module (RD55UP12-V)**

Rev. 1.0  
Mar. 1<sup>st</sup>, 2021



Rev.	日付	変更内容
1.0	Mar. 1 <sup>st</sup> , 2021	Initial version

#### Cautions

- The copyright in this document belongs to Lineo Solutions, Inc.
- This document should not be used or duplicated, in any form, in whole or in part, without permission.
- In any case, Lineo Solutions, Inc. makes no related representations or warranties for results of using this document.
- The content of this document and the specification of this software might be changed without a previous notice.

#### Trademark

Lineo Solutions, Inc. is a trademark of Lineo Solutions, Inc. MELSEC, MELSEC iQ-R and GX Works are trademarks of Mitsubishi Electric Corporation. Linux is a registered trademark of Linus Torvalds. All other company and product names mentioned are the trademarks and registered trademarks of their respective companies.

Part No. : 25224-G5KA6S
-------------------------

Copyright© 2008-2021 Lineo Solutions, Inc. All rights reserved

## Index

1.	Introduction .....	4
2.	System Configuration .....	4
3.	Webserver access on C intelligent module .....	6
3.1.	Pre-requisite .....	6
3.2.	Installation of Webserver packages .....	6
3.3.	Configuration of Webserver .....	7
3.4.	Execution of Webserver.....	11
3.5.	Usage of Webserver .....	13
4.	Appendix .....	17
4.1	Manual method.....	17

## 1. Introduction

The purpose of this document is to explain the configuration, installation steps and usage of web server template on C intelligent function module.

## 2. System Configuration

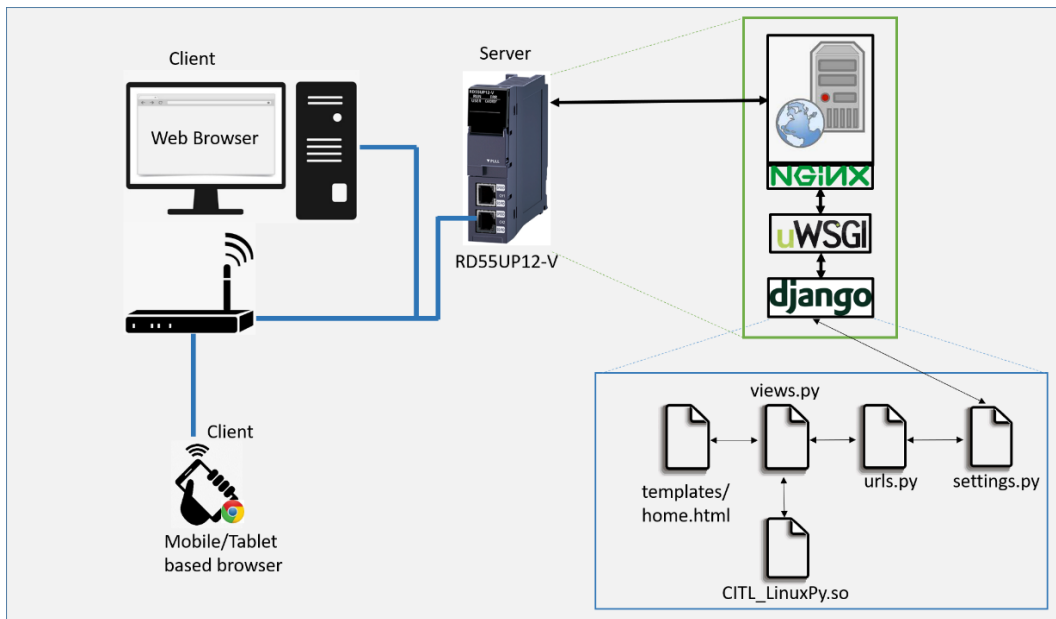


Figure 2-1: Webserver access function on C Intelligent module

File Name	Description
/etc/nginx/sites-available/WebServerTemplate_nginx.conf	<ul style="list-style-type: none"> <li>- The way nginx and its modules work is determined in the configuration file</li> <li>- It contains IP address, port, socket file path, etc configuration related to web server</li> </ul>
/etc/nginx/sites-enabled/WebServerTemplate_nginx.conf	<ul style="list-style-type: none"> <li>- Link of file /etc/nginx/sites-available/WebServerTemplate_nginx.conf</li> </ul>
/home/LAT_01_WebServer/Template/WebServerTemplate/WebServerTemplate.sock	<ul style="list-style-type: none"> <li>- Socket file for communication between nginx and uwsgi</li> <li>- File is mentioned in the WebServerTemplate_nginx.conf for nginx to communicate</li> </ul>
/home/LAT_01_WebServer/Template/WebServerTemplate/manage.py	<ul style="list-style-type: none"> <li>- This file is the command line utility of the project</li> <li>- Use to deploy, debug and test django project</li> <li>- The file contains the code for starting the server</li> </ul>
/home/LAT_01_WebServer/Template/WebServerTemplate/WebServerTemplate/settings.py	<ul style="list-style-type: none"> <li>- Main settings file of the Django project</li> <li>- Django applications WebServerApp is added here</li> </ul>
/home/LAT_01_WebServer/Template/WebServerTemplate/WebServerTemplate/urls.py	<ul style="list-style-type: none"> <li>- Contains the project level URL information</li> <li>- Provides the address of the resource like images, webpages, web-applications for website.</li> <li>- Connect the WebServerApp with the project</li> </ul>
/home/LAT_01_WebServer/Template/WebServerTemplate/WebServerApp/forms.py	<ul style="list-style-type: none"> <li>- Contains the web page form architecture and specification</li> <li>- This file defines input fields which appears on the web page</li> </ul>
/home/LAT_01_WebServer/Template/WebServerTemplate/WebServerApp/views.py	<ul style="list-style-type: none"> <li>- It processes gorm data sent back to a Django website</li> <li>- Contains code which is responsible for buffer read and write</li> <li>- It handles most of the logical part of the application.</li> </ul>
/home/LAT_01_WebServer/Template/WebServerTemplate/WebServerApp/templates/home.html	<ul style="list-style-type: none"> <li>- Contains the web page html code of the web server template</li> <li>- The html page is render from views.py with the appropriate context data</li> </ul>

## 3. Webserver access on C intelligent module

### 3.1. Pre-requisite

Before proceeding to actual step(s) of setting of Webserver on C intelligent module, following are the prerequisites.

1. Extract the “*LAT\_01\_WebServer.7z*” and copy it to C intelligent module using any file sharing software (e.g winscp) to **/home directory**.
2. To login to the C Intelligent module use any SSH client software (e.g PuTTY).

### 3.2. Installation of Webserver packages

A script named “*webserver\_installation.sh*” installs the prerequisite packages required for Webserver solution template.

1. Navigate to path `/home/LAT_01_WebServer/Scripts/`

```
# cd /home/LAT_01_WebServer/Scripts/
```

```
root@rd55up~v:~# cd /home/LAT_01_WebServer/Scripts/  
root@rd55up~v:/home/LAT_01_WebServer/Scripts#
```

2. Change the permission of “*webserver\_installation.sh*”

```
# chmod +x webserver_installation.sh
```

```
root@rd55up~v:/home/LAT_01_WebServer/Scripts# chmod +x  
webserver_installation.sh  
root@rd55up~v:/home/LAT_01_WebServer/Scripts#
```

3. Execute the “*webserver\_installation.sh*”

```
# ./webserver_installation.sh
```

(Note: Installation time may vary from 15 to 20 mins depending on internet speed.)

```
root@rd55up~v:/home/LAT_01_WebServer/Scripts# ./webserver_installation.s  
h  
Installing of WebServerTemplate depending packeages...  
Executeing ulimit -s unlimited  
Command Succeeded  
Executing apt-get install -y python3-pip  
Reading Package lists... Done  
Buildong dependency tree... 75%
```

4. Give permission for nginx installation by entering Y  
(note: depend on version, additional disk space bytes are changed.)

```
root@rd55up~v:/home/LAT_01_WebServer/Scripts# ./webserver_installation.s
h
Installing of WebServerTemplate depending packages...
Executeing ulimit -s unlimited
Command Succeeded
Executing apt-get install -y python3-pip
~~~
.
.
.
~~~
After this operation, 16.3MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

5. Confirm installation process completed successfully with below message  
*“Installation of WebServerTemplate dependent packages are done. Successfully!!”*

```
Executing nginx -v
nginx version: nginx/1.10.3
Command Succeeded
Installation of WebServerTemplate dependent packages are
done.Successfully!!
root@rd55up12-v:/home/LAT_01_WebServer/Scripts#
```

### 3.3. Configuration of Webserver

A script named “*webserver\_configuration.sh*” performs the configuration at the server (i.e. server-side configuration) required.

1. Navigate to the /home/LAT\_01\_WebServer/Template/ directory  
# cd /home/LAT\_01\_WebServer/Template/

```
root@rd55up12-v:# cd /home/LAT_01_WebServer/Template/
root@rd55up12-v: /home/LAT_01_WebServer/Template#
```

2. Open “*WebServerTemplate\_nginx.conf*” with nano editor  
# nano WebServerTemplate\_nginx.conf

```
root@rd55up12-v: /home/LAT_01_WebServer/Template# nano
WebServerTemplate_nginx.conf
```

3. Edit highlighted line and modify present IP address with your target device IP address.

server\_name <RD55UP12-V IP ADDRESS>; # substitute your machine's IP address

After that, save the file “WebServerTemplate\_nginx.conf” and exit the nano editor.

```
GNU nano 2.7.4                               File: WebServerTemplate_nginx.conf

# WebServerTemplate_nginx.conf

# the upstream component nginx needs to connect to
upstream django {
    server unix:///home/LAT_01_WebServer/Template/WebServerTemplate/WebServerTemplate.sock; #
for a file socket
}

# configuration of the server
server {
    # the port your site will be served on
    listen      8000;
    # the domain name it will serve for
    server_name 172.31.78.2; # substitute your machine's IP address
    charset     utf-8;

    # max upload size
    client_max_body_size 75M; # adjust to taste

    # Django media
    location /media {
        alias /home/LAT_01_WebServer/Template/WebServerTemplate/media; # your Django project's
media files - amend as required
    }

    location /static {
        alias /home/LAT_01_WebServer/Template/WebServerTemplate/static; # your Django project's
static files - amend as required
    }

    # Finally, send all non-media requests to the Django server.
    location / {
        uwsgi_pass  django;
        include     /home/LAT_01_WebServer/Template/WebServerTemplate/uwsgi_params; # the
uwsgi_params file you installed
    }
}
```

[ Read 33 lines ]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page M-Y First Line M-W WhereIs Next  
^X Exit ^R Read File ^Y Replace ^U Uncut Text ^T To Spell ^\_ Go To Line ^V Next Page M-/ Last Line M-] To Bracket



4. Navigate to the /home/LAT\_01\_WebServer/Scripts/ directory

```
# cd /home/LAT_01_WebServer/Scripts/
```

```
root@rd55up12-v:# cd /home/LAT_01_WebServer/Scripts/  
root@rd55up12-v: /home/LAT_01_WebServer/Scripts#
```

5. Change the permission of “*webserver\_configuration.sh*”

```
# chmod +x webserver_configuration.sh
```

```
root@rd55up12-v: /home/LAT_01_WebServer/Scripts# chmod +x  
webserver_configuration.sh  
root@rd55up12-v: /home/LAT_01_WebServer/Scripts#
```

6. Execute the “*webserver\_configuration.sh*”

```
# ./webserver_configuration.sh
```

```
root@rd55up12-v:
/home/LAT_01_WebServer/Scripts# ./webserver_configuration.sh
Configuring of WebServerTemplate dependent tools...
Copying nginx configuration
Executing cp
/home/LAT_01_WebServer/Template/WebServerTemplate_nginx.conf
/etc/nginx/sites-available/
Command Succeeded
~~~
...
```

7. Confirm configuration process completed successfully with below message

*“Configuration of WebServerTemplate dependent tools are done. Successfully!!”*

```
root@rd55up12-v:
/home/LAT_01_WebServer/Scripts# ./webserver_configuration.sh
Configuring of WebServerTemplate dependent tools...
Copying nginx configuration
Executing cp
/home/LAT_01_WebServer/Template/WebServerTemplate_nginx.conf
/etc/nginx/sites-available/
Command Succeeded
Applying nginx configuration
Executing sudo ln -s /etc/nginx/sites-
available/WebServerTemplate_nginx.conf /etc/nginx/sites-enabled/
Command Succeeded
Configuration of WebServerTemplate dependent tools are done.
Successfully!!
root@rd55up12-v:/home/LAT_01_WebServer/Scripts#
```

### 3.4. Execution of Webserver

A script named “*webserver\_execution.sh*” executes the webserver application. Before accessing the webserver from browser, it is required to execute this script. **If a session is closed or after power cycle it is required to re-execute this script to use webserver.**

1. Navigate to the `/home/LAT_01_WebServer/Scripts/` directory

```
# cd /home/LAT_01_WebServer/Scripts/
```

```
root@rd55up12-v: # cd /home/LAT_01_WebServer/Scripts/  
root@rd55up12-v: /home/LAT_01_WebServer/Scripts#
```

2. Change the permission of “*webserver\_execution.sh*”

```
#chmod +x webserver_execution.sh
```

```
root@rd55up12-v: /home/LAT_01_WebServer/Scripts# chmod +x  
webserver_execution.sh  
root@rd55up12-v: /home/LAT_01_WebServer/Scripts#
```

3. Execute the “*webserver\_execution.sh*”

```
# ./webserver_execution.sh
```

```
root@rd55up12-v:/home/LAT_01_WebServer/Scripts# ./webserver_execution.sh  
Executing WebServerTemplate...  
To access WebServer visit http://192.168.1.3:8000/  
Hit [CTRL+C] to stop!  
Starting Nginx web server...  
Executing sudo /etc/init.d/nginx restart  
[ ok ] Restarting nginx (via systemctl): nginx.service.  
Command Succeeded  
Executing cd /home/LAT_01_WebServer/Template/WebServerTemplate/  
~~~  
...  
...
```

4. Confirm execution process started successfully with below message

“\*\*\* Starting uWSGI 2.0.19.1 (32bit) on [Sun Oct 4 19:48:10 2020] \*\*\*”

```
root@rd55up12-v:/home/LAT_01_WebServer/Scripts#./webserver_execution.sh
Executing WebServerTemplate...
To access WebServer visit http://192.168.1.3:8000/
Hit [CTRL+C] to stop!
Starting Nginx web server...
Executing sudo /etc/init.d/nginx restart
[ ok ] Restarting nginx (via systemctl): nginx.service.
Command Succeeded
Executing cd /home/LAT_01_WebServer/Template/WebServerTemplate/
Command Succeeded
Starting Django web server app through uwsgi...
Executing uwsgi --socket
/home/LAT_01_WebServer/Template/WebServerTemplate/WebServerTemplate.sock
--module WebServerTemplate.wsgi --chmod-socket=666
*** Starting uWSGI 2.0.19.1 (32bit) on [Fri Feb 26 11:45:17 2021] ***
compiled with version: 6.3.0 20170516 on 26 February 2021 01:09:07
os: Linux-4.9.76-rt61-ltsi-mei-v2+ #3 SMP PREEMPT RT Mon Jan 25 18:48:15
IST 2021
nodename: rd55up12-v
machine: armv7l
~~~
...

```

**Note**

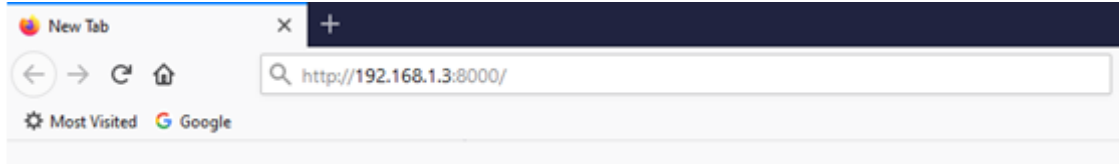
Read actual IP address  
of your RD55

### 3.5. Usage of Webserver

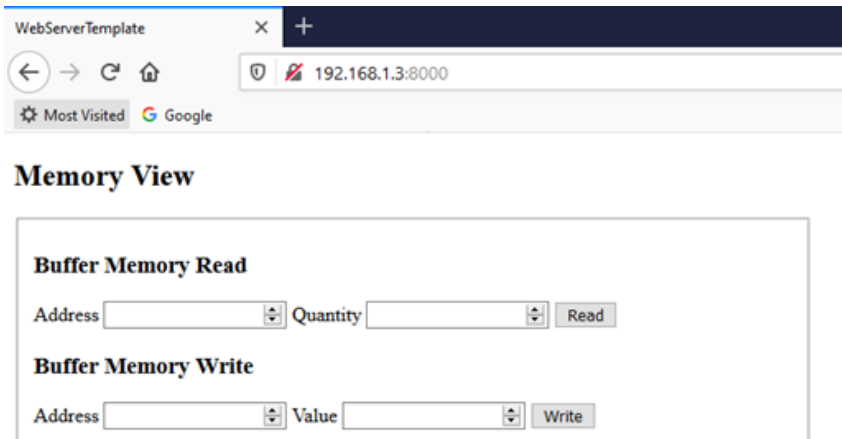
1. Open any web browser on a host PC.

Type web server address (e.g `http://192.168.1.3:8000/`) in the address field and hit enter.

Address is “`http://<RD55UP12-V IP ADDRESS>:8000/`”



Webserver template web page will appear in the web browser as shown below (default).



2. Read Operation:

- a. Enter “Address” and “Quantity” in the input box available under Buffer Memory Read.



**Memory View**

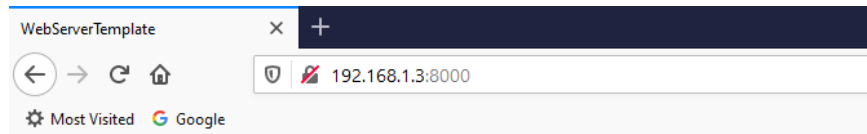
**Buffer Memory Read**

Address  Quantity

**Buffer Memory Write**

Address  Value

- b. Click the Read button to read the specified quantity of data from the specified address.



**Memory View**

**Buffer Memory Read**

Address  Quantity

**Buffer Memory Write**

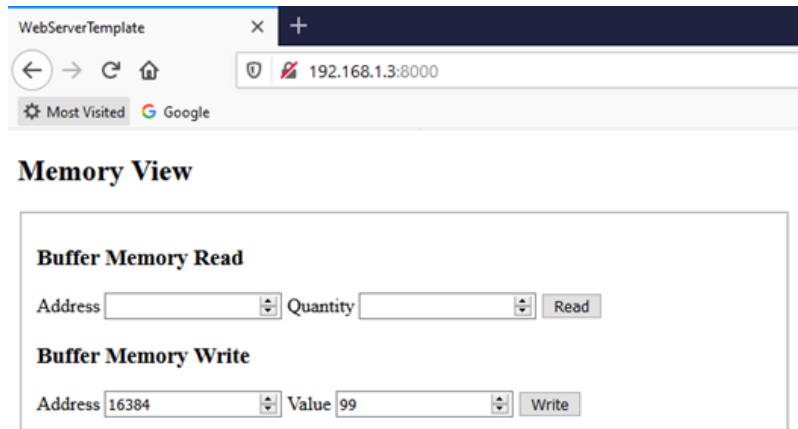
Address  Value

**STATUS :: Read Success**

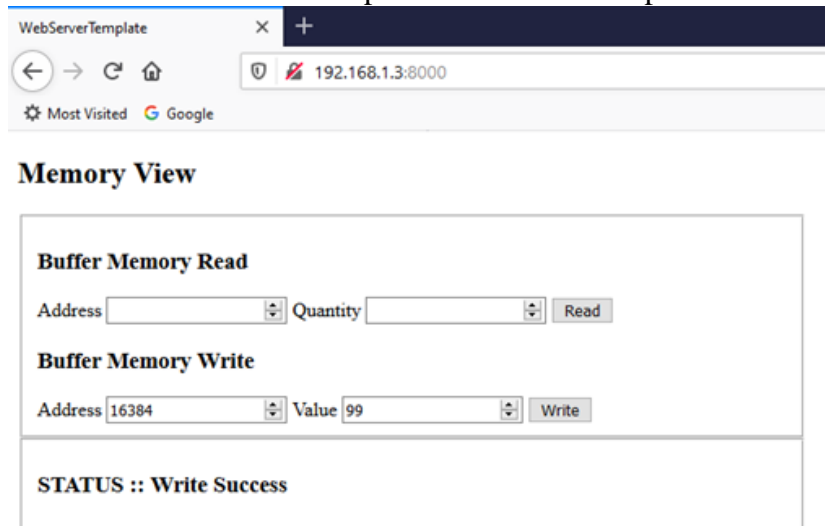
Buffer Memory Address	Value
50	11832
51	11825
52	51
53	45496
54	46772

3. Write Operation:

- a. Enter “Address” and “Value” in the input box available under Buffer Memory Write



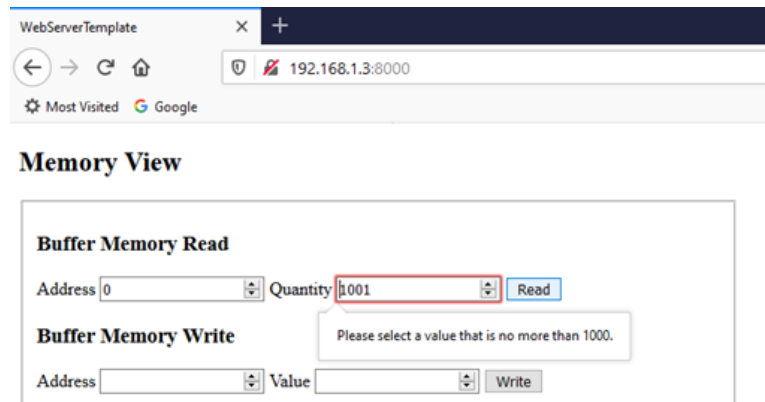
- b. Click the Write button to write the specified value to the specified address



#### 4. Range

	Address	Quantity / Value
Read Operation	0 to 2097151	1 to 10000
Write Operation	163841 to 2097151	0 to 65535

If you try to read over range quantity, it would be below.



#### 5. Status Description

STATUS	Description
Read Success	Buffer memory read operation completed successfully
Write Success	Buffer memory write operation completed successfully
Values out of range	Address and Quantity combination goes beyond max buffer memory address 2097151, try by reducing Address Or Quantity appropriately.



## 4. Appendix

### 4.1 Manual method

#### 4.1.1 installation of Webserver

1. Connect to RD55UP12-V SSH terminal using any SSH client (for example PuTTY) and Login to C intelligent module using username and password.
2. Run below command to install pip  
`#apt-get install -y python3-pip`  
(Note: - Installation time may vary from 15 to 20 mins depending on internet speed.)
3. Confirm installation of “*pip 9.0.1 from /usr/lib/python3/dist-package (python 3.5)*” with below command  
`#python3 -m pip --version`
4. Run below command to install Django  
`# python3 -m pip install Django`
5. Confirm installation version 2.2.16 with below command  
`# python3 -m django --version`
6. Run below command to install uwsgi  
`# python3 -m pip install uwsgi`
7. Run below command to install nginx  
`# sudo apt-get install nginx`
8. Give permission for nginx installation by entering Y when prompted below message, Do you want to continue? [Y/n]
9. Confirm installation of nginx version(nginx/1.10.3) with below command  
`# nginx -v`

#### 4.1.2 Configuration of Webserver

1. Navigate to the /home/LAT\_01\_WebServer/Template/ directory  
`# cd /home/LAT_01_WebServer/Template/`
2. Open WebServerTemplate\_nginx.conf using nano editor  
`# nano WebServerTemplate_nginx.conf`
3. Edit highlighted line and modify present IP address with your target device IP address.  
`server_name <RD55UP12-V IP ADDRESS>; # substitute your machine's IP address`
4. Save the file WebServerTemplate\_nginx.conf and exit the nano editor.
5. Copy the file WebServerTemplate\_nginx.conf to the /etc/nginx/sites-available/ with below command  
`#cp/home/LAT_01_WebServer/Template/WebServerTemplate_nginx.conf /etc/nginx/sites-available/`
6. Link the file WebServerTemplate\_nginx.conf with below command to enable the configuration  
`#ln -s /etc/nginx/sites-available/WebServerTemplate_nginx.conf /etc/nginx/sites-`

enabled/

### 4.1.3 Execution of Webserver

1. Start nginx server with below command  
# /etc/init.d/nginx restart
2. Navigate to the /home/LAT\_01\_WebServer/Template/WebServerTemplate/  
# cd /home/LAT\_01\_WebServer/Template/WebServerTemplate/
3. Start Django app through uwsig with below command  
#uwsgi --socket  
/home/LAT\_01\_WebServer/Template/WebServerTemplate/WebServerTemplate.sock  
--module WebServerTemplate.wsgi --chmod-socket=666
4. Confirm execution process started successfully with below message  
\*\*\* Starting uWSGI 2.0.19.1 (32bit) on [Sun Oct 4 21:26:05 2020] \*\*\*