



4D Web Server Code Templates

Customized code with fast deployment

Hector Flores Garza November 15, 2023

hectorfg1@me.com

4D Web Server Code Templates

Benefits

- **Customized Code**

Clients running our 4d apps may need tailored code to allow communication with third party applications, and enterprise applications. You are able to deploy customized code to your clients quickly without needing to build and distribute a new version of your core product.

- **Multiple Connections**

One 4d Web Server license gives your client the ability to handle simultaneous multiple connections.

- **Multiple Threads Execution**

When run in 4d preemptive mode, each process handling the http request can be allocated to each CPU separately on a multi-core machine. Note: You must make sure your template code plus all 4d methods being called by the Http request inside your app are thread-safe.

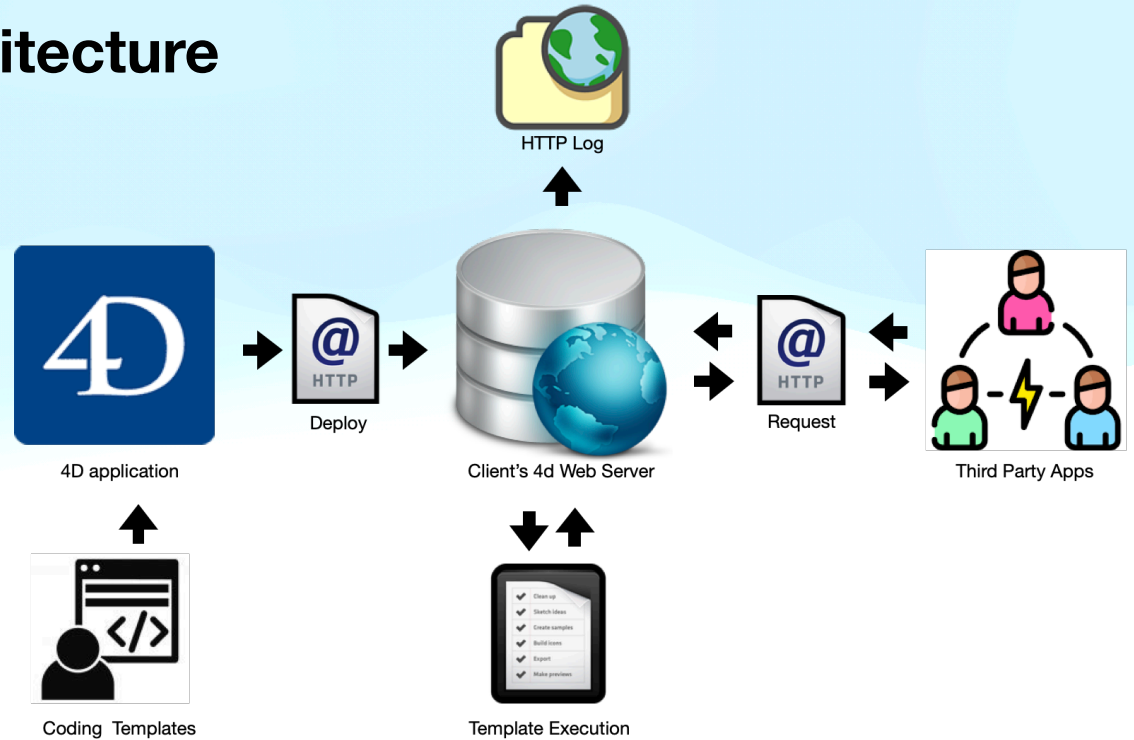
- **Fast Deployment**

Multiple templates can be deploy via HTTP to different 4d app instances within a process session.

4D Web Server Code Templates

Core Components and Architecture

- 4D Web Server
- Web Server Code Templates
- Template Deployment
- PROCESS 4D TAGS
- HTTP
- HTTP Log



4D Web Server Code Templates

Templates

- Description
- Security
- Code
- Register WebServers
- Templates Selector

Description	Security	Code	Register WebServers						
ID	Interpreter	Category	Response Type	Latest Build	Name	Version	<input type="checkbox"/> Request/JSON Content		
AddStockTikItems	4D Tags	Tickets	Text/Plain		AddStockTikItems	1.10			
Description <input type="checkbox"/> On Schedule									
<p>Version History</p> <ul style="list-style-type: none">1.10: added ReadOnlyAll1.9: Added reduce selection1.8: Added ability to override price1.7: Rolled back 1.61.6: Changed Stock Product ID to Stock Product ProductNo in main query for product1.5: Updated code to set the JobType_Local field1.4: Updated to run under LT 8.21.3: Fixed error when calling FindStockPrice method, was calling an Alert on API Server, replaced with direct call to another method1.2: Added more FC calculations1.1: Updated [Ticket]TicQuantity prior to Ticket Items created, needed for calculation for Linked to Cust Prods1.0: Template created <p>Inputs</p> <ul style="list-style-type: none">1. ticket Number: ticketNum2. Customer Number: customerNum3. Array Stock Product IDs : stockProdID4. Array Stock Prod Qty: stockProdQty <p>NOTE: The ID array is the Stock Product ID, not the Stock Product Number!!!</p> <p>Example JSON: {"ticketNum": "109083","customerNum": "1603","stockProdID":["946","1032"],"stockProdQty": [1000,2000]}</p> <p>Output is TicketItem IDs, or error message (text)</p>									

4D Web Server Code Templates

Templates

- Description
- Security
- Code
- Register WebServers
- Templates Selector

Description **Security** Code Register WebServers

ID: AddStockTikItems Interpreter: 4D Tags Category: Tickets Response Type: Text/Plain Latest Build: Version: AddStockTikItems 1.10 Request/JSON Content

Authorize IP's
 Enable
IP Allowed

Authorize Users
 Enable Show Password
User Password

SSL
 Enable

Encrypt Method: AES128 Password: 4dmethod

4D Web Server Code Templates

Templates

- Description
- Security
- Code
- Register WebServers
- Templates Selector

Description	Security	Code	Register WebServers				
ID	Interpreter	Category	Response Type	Latest Build	Name	Version	
AddStockTikItems	4D Tags	Tickets	Text/Plain		AddStockTikItems	1.10	<input type="checkbox"/> Request/JSON Content

```
C_TEXT($1;RequestBody_txt;W2P_TicketNum_txt;W2P_CustomerNum_txt)
ARRAY TEXT(W2P_StockProdID_atxt;0) //change to zero
ARRAY LONGINT(W2P_StockProdQty_an;0) //change to zero
ARRAY REAL(W2P_StockProdPrice_an;0)
C_OBJECT(RequestJson_obj)

ReadOnlyAll

RequestBody_txt:=$1

RequestJson_obj:=JSON Parse(RequestBody_txt;Is object) // Convert Message body to a JSON Object

W2P_TicketNum_txt:=OB Get(RequestJson_obj;"ticketNum")
W2P_CustomerNum_txt:=OB Get(RequestJson_obj;"customerNum")

OB GET ARRAY(RequestJson_obj;"stockProdID";W2P_StockProdID_atxt)
OB GET ARRAY(RequestJson_obj;"stockProdQty";W2P_StockProdQty_an)
OB GET ARRAY(RequestJson_obj;"stockProdPrice";W2P_StockProdPrice_an)

API_Output_vt:= ""

C_LONGINT(Counter_n)
C_TEXT(resultMakeTicket)

READ WRITE([Ticket])
QUERY([Ticket];[Ticket]Number=W2P_TicketNum_txt)
C_LONGINT(Counter_TicQuantity;StockProdQuantity)
StockProdQuantity:=0
For (Counter_TicQuantity;1;Size of array(W2P_StockProdQty_an))
    StockProdQuantity:=StockProdQuantity+W2P_StockProdQty_an(Counter_TicQuantity)
End for

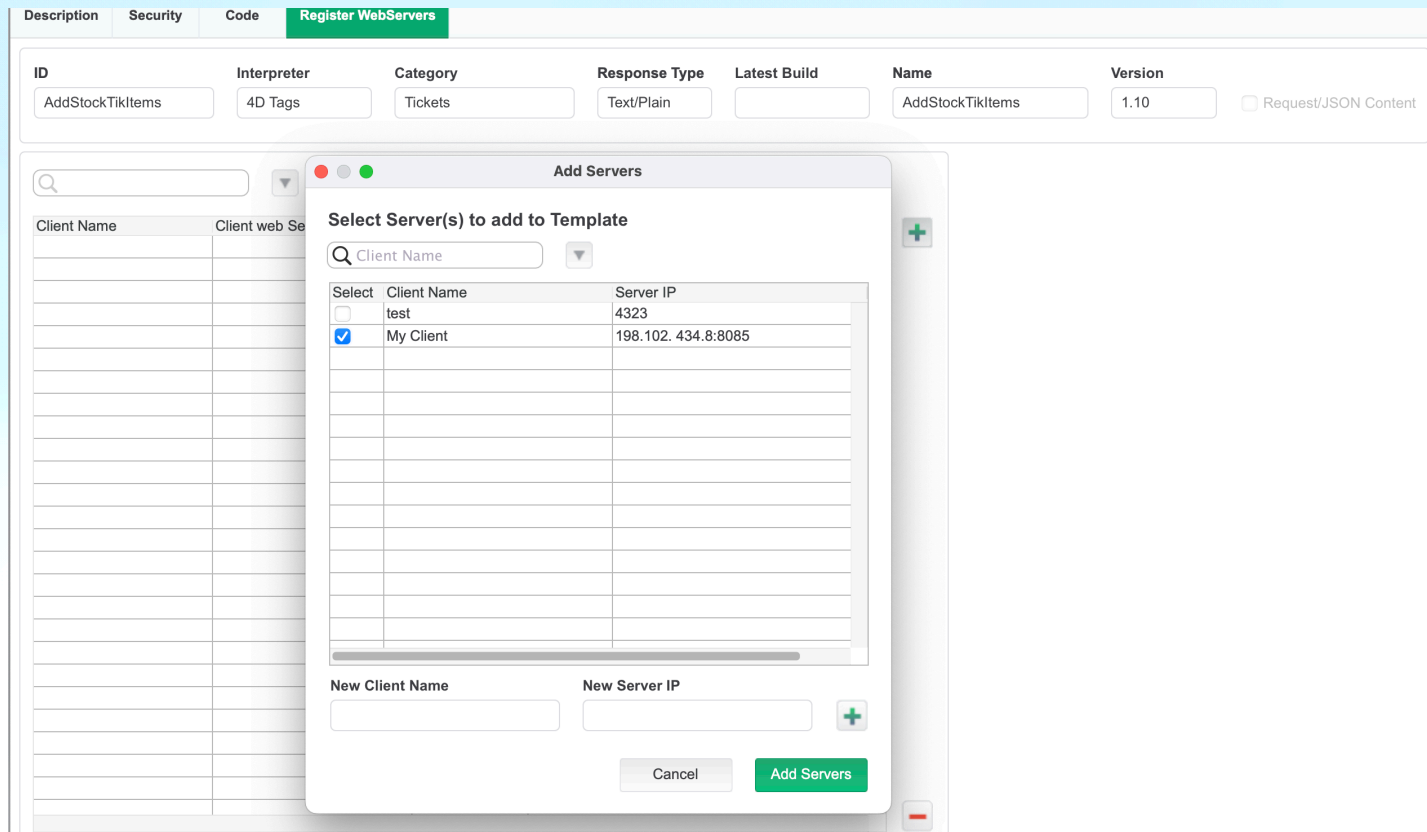
[Ticket]TicQuantity:=StockProdQuantity

For (Counter_n;1;Size of array(W2P_StockProdID_atxt))
    resultMakeTicket:= ""
```

4D Web Server Code Templates

Templates

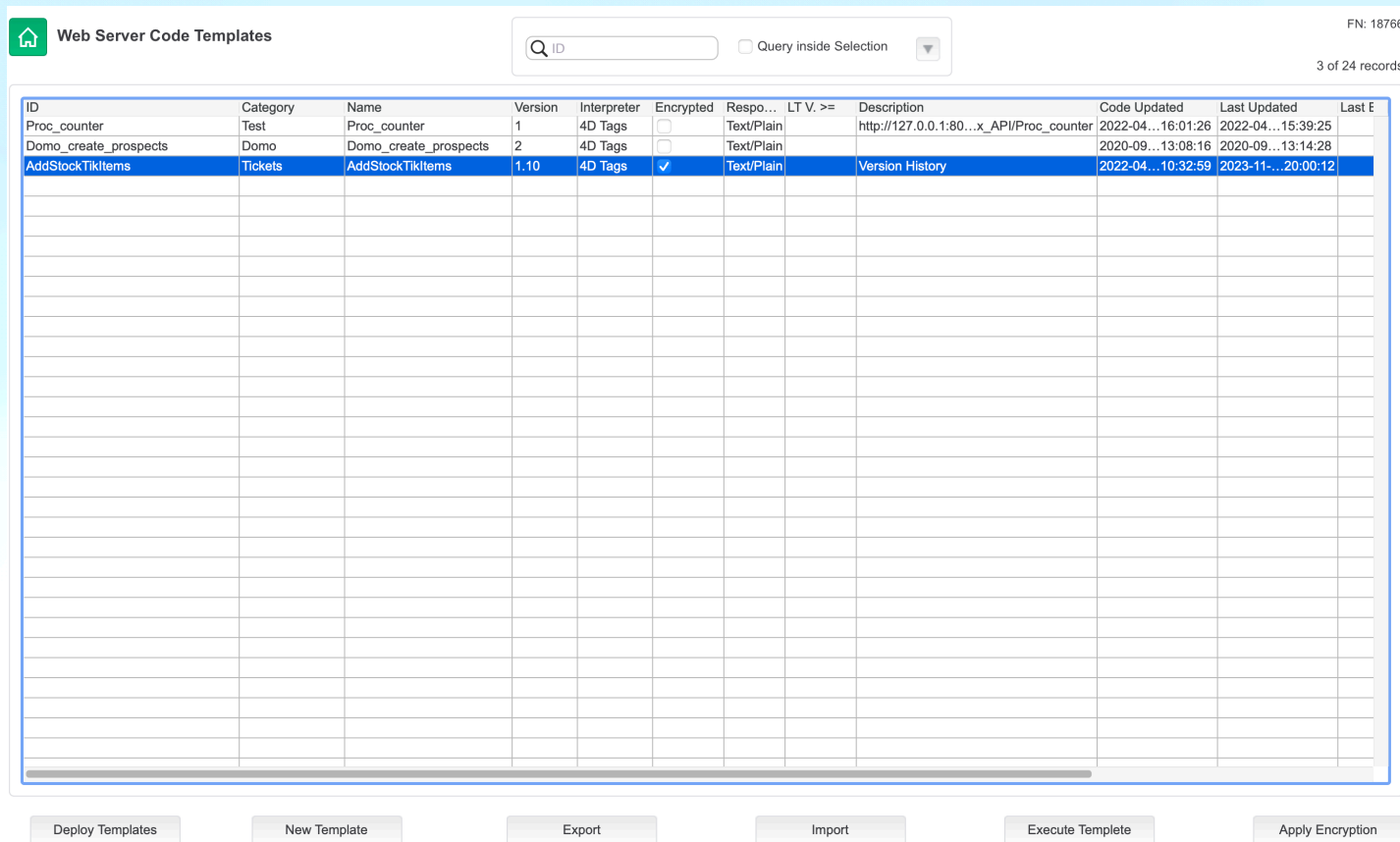
- Description
- Security
- Code
- Register WebServers
- Templates Selector



4D Web Server Code Templates

Templates

- Description
- Security
- Code
- Register WebServers
- Templates Selector



The screenshot displays the 'Web Server Code Templates' management interface. At the top, there is a search bar for 'ID', a 'Query inside Selection' checkbox, and a dropdown arrow. The main area is a table with the following columns: ID, Category, Name, Version, Interpreter, Encrypted, Respo..., LT V. >=, Description, Code Updated, Last Updated, and Last E. The table contains three rows of data:

ID	Category	Name	Version	Interpreter	Encrypted	Respo...	LT V. >=	Description	Code Updated	Last Updated	Last E
Proc_counter	Test	Proc_counter	1	4D Tags	<input type="checkbox"/>	Text/Plain		http://127.0.0.1:80...x_API/Proc_counter	2022-04...16:01:26	2022-04...15:39:25	
Domo_create_prospects	Domo	Domo_create_prospects	2	4D Tags	<input type="checkbox"/>	Text/Plain			2020-09...13:08:16	2020-09...13:14:28	
AddStockTikItems	Tickets	AddStockTikItems	1.10	4D Tags	<input checked="" type="checkbox"/>	Text/Plain		Version History	2022-04...10:32:59	2023-11-...20:00:12	

At the bottom of the interface, there are several buttons: 'Deploy Templates', 'New Template', 'Export', 'Import', 'Execute Template', and 'Apply Encryption'.

4D Web Server Code Templates

Template Deployment

4d methods (Deployment)

- WCT_deployTemplates()
- WCT_addClientsWebServers()
- WTC_createRegistry()
- WCT_sendTemplates()

Templates to Send

ID	Category	Name	Version
Proc_counter	Test	Proc_counter	1
AddStockTikItems	Tickets	AddStockTikItems	1.10

Select Server IP to send to

Q Client Name ▼

Select	Client Name	Server IP
<input type="checkbox"/>	test	4323
<input checked="" type="checkbox"/>	My Client	198.102. 434.8:8085

New Client Name New Server IP

4D Web Server Code Templates

PROCESS 4D TAGS

```
var $templateCode_t; $outputResult_t; $body_t : Text
var $param :Object

// add #4DCODE TAG to the templateCode_t script
$templateCode_t:="<!--#4DCODE"+Char(Carriage_return)+$TemplateCode_t+Char(Carriage_return)+"-->"
PROCESS 4D TAGS($TemplateCode_t; $WCT_Output_t; $Content_t; $Param)
```

4D Web Server Code Templates

HTTP (4d methods)

HTTP

- WCT_verifyURL()
- WCT_httpErrorHandler()

HTTP Log

- WCT_httpLog()
- WCT_httpLogSelector

Web Authentication

- WCT_OnWebAuthentication()
- WCT_authentication()

Http Request

- WCT_OnWebConnection()
- WCT_exeHttpRequest()
- WCT_exeTemplate()
- WCT_encryptDecrypt()

4D Web Server Code Templates

HTTP Log

- HTTP Log Selector
- Request
- Respond
- Reports

The screenshot displays an HTTP Log viewer interface. At the top, there are buttons for 'Select All', 'Clear All', and 'Create Report', along with a '1 of 1' indicator. The main area shows a log entry for ID 678914, dated 11/4/23 at 20:19:20. The URL is /Mytemplates/AddStockTikItems. The log is divided into 'Request' and 'Respond' tabs, with 'Request' selected. The 'Request' tab shows the following details:

- Reference:
- Header:

```
GET /Mytemplates/AddStockTikItems HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate, br
Authorization: Basic MTE3MjlxOjY3VmMxMDRmNDk=
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 101
```
- Body:

```
{"ticketNum": "109083","customerNum": "1603","stockProdID":["946","1032"],"stockProdQty":{1000,2000}}
```

On the right side, there is a sidebar with various filters and settings, all of which are checked:

- Duration: 00:00:00
- Module: API
- Module status:
- Error_EXE: Yes
- User_Name: 117221
- Password: 67Vc104f49
- Authentication Req.: Yes
- Authentication Failed: Yes
- Restricted IP: Yes
- SSL_Required: Yes
- SSL_Required_module: Yes
- SSL_Connection: Yes
- Server_IP_Address: 192.168.0.15
- Client_IP_Address: 192.168.0.15

4D Web Server Code Templates

HTTP Log

- HTTP Log Selector
- Request
- Respond
- Reports

Http Log 1 of 1

ID : 678914 Date : 11/4/23 Time : 20:19:20

URL_Request : /Mytemplates/AddStockTikItems

Status : 401

Header : X-STATUS: 401 Unauthorized

Response :

Duration : 00:00:00

Module : API

Module status :

Error_EXE : Yes

User_Name : 117221

Password : 67Vc104f49

Authentication Req. : Yes

Authentication Failed : Yes

Restricted IP : Yes

SSL_Required : Yes

SSL_Required_module : Yes

SSL_Conexion : Yes

Server_IP_Address : 192.168.0.15

Client_IP_Address : 192.168.0.15

4D Web Server Code Templates

Future Improvements

- Deployment Log



4D Web Server Code Templates

Customizable code with no build required



Q&A



Hector Flores Garza November 15, 2023
hectorfg1@me.com