

CREATING A TO-DOs APP FROM 0 USING FaaS WORKFLOWS

- THE STEP BY STEP COMPANION -

SPEAKER: DIEGO MARTIN
marg@zhaw.ch



0.- INDEX

1. Getting the code
2. Setting up the environment
3. Let's test it!
4. Future improvements...

1.- GETTING THE CODE

Before we start you should know that this tutorial is a bit different than the typical tutorial you might have encounter in your average events: it's not a full fledged real-time hands-on tutorial and neither one of those in which you might spent a few hours reading slides and getting tones of information and end up with nothing that works on your hands.

So, “what on Earth is this tutorial then?” you might wonder, well, the answer is something like this:

This tutorial is oriented to people that have never ever touched the technology we are going to use, so all the code needed to get the app running is provided with proper comments (and some are skipped on purpose, so you have an excuse to dive further in the tech, you will notice ;) so after the tutorial you shall have a working app and an environment with enough information to play with it on your own.

So, without taking more time on this, let's open your favorite web browser and go to <https://bit.ly/FWTCODE> to grab the file and unzip it wherever you prefer :)

2.- SETTING UP THE ENVIRONMENT

If you are one of those that cannot resist the temptation and went on and checked the code you might realize that there's a few "REPLACE_ME" around the code, some of them will be auto-handled by the code, but three of them are going to need your manual input (remember, I just said you will end up with something that "just worksTM", no that you won't have to dirty your hands a bit ;P)

So, for the first one, you need to open your favorite terminal: **[Ctrl+Alt+t]** should open the default one in your WM :)

Once there, you have to go the folder where you unzip the code. If you just downloaded and unzip there something like this should work:

```
cd Downloads/FWT_code
```

Now, you should issue **pwd** then enter and then copy the line, it should be something like this:

```
/home/YOURUSER/Downloads/FWT_code
```

Once you have that, go and open you favorite text editor and open the file called: **front_functions.sh**, inside the **src** folder, once there look for this line:

```
readonly gitdir="REPLACE_ME"
```

And yes, you're guessing right, you need to put the line we've just copied instead of the **REPLACE_ME**, note that I haven't included the quotes, so remember to keep them surrounding the line you just added.

After that, we have to go to the folder called **composer** and execute the file **setup.sh**, for doing that you should move to the folder (**cd composer**) and then issue a **./setup.sh** in the terminal. It will ask for your password at least once.

Meanwhile the previous step finish (it may take a couple of mins), for the other two **REPLACE_ME** that you might have just seen on top of that one we changed we need to go to <https://bluemix.net> and create and account.

Once you have the account (and have logged in) is where the tricky part begins.

First, open the **burger menu** (the icon with three vertical lines in a stack), go to **Functions** and then, under “**Getting started**” go to **CLI**.

At this point there should be a kind of popup telling you that “there’s no cloud foundry space” and inviting you to create one, so click on it.

Now, you will see your e-mail and “...” under “**Actions**”, click on it and go to “**Spaces**” and then create one with the “**Add a space**” button, you will have to choose “**UK**” as region and a name, let’s use “**FWT**” for example.

Once that’s done you can close the tab and go back to the other one with the popup.

So, again with the popup, just close it and notice that in the upper line you probably have “**Dallas**” as region, go and select “**London**” instead and the other two fields should’ve changed to your e-mail and the name you selected for your space (if not, just pick then :) and now you have to copy the line in the 3rd step of the list.

By this time the setup in your terminal should’ve already finished (if not, wait for it) so now you can paste the line we’ve just copied and hit [**Enter**]. It will ask for your *Bluemix credentials* and then probably for the region which should be **eu-gb**, and maybe it will suggest to update, if so, just accept the update.

Finally, we have to go again to the **front_functions.sh** file and change the **REPLACE_ME**s with your e-mail and the space you’ve just created, and we’re ready to go!

3.- LET'S TEST IT!

Once we have everything on place, let's try it!

For it, let's go to the **composer** folder again and issue a **./example.sh**
If everything was alright, after following the steps of the script and pressing **[Enter]** a couple of times you should be facing some like this:

```
> Welcome to the Simple Todos App!  
> Here you have your Todos  
  
Todos:  
  
    2 -> "Finish the ToDo App!"  
  
WiPs:  
  
    1 -> "Attend the FaaS Workflows Tutorial!"  
  
Done:  
  
    0 -> "Create the initial test-ToDo :)"  
  
> Please state your command:
```

Now let's try it, issue a **done 2**, and the todo #2 will move to the done stack and the interface will update to reflect the change. Nice, ah?

In the real-time tutorial, after playing a bit with the app we would be going file by file checking the code and getting a proper explanation with interactive Q/A.

To emulate that, the code has gotten all the necessary comments and extra explanations you may need to get a understanding of what each file does, so feel invited to check the code and play with the app as much as you want :)

If you have any doubt try googling it or ask me by e-mail, I'll try to answer back as soon as I can ;)

If by any chance you liked it and wonder how can you just use it without all the pre-loading stuff, just create a new file in the composer folder: “**app.sh**” should suffice, and add the following 3 lines:

```
#!/bin/bash  
source ../src/front_functions.sh  
cliTodosApp composer
```

Now you have to make it executable, so either **right-click** → **properties** → **permissions** → **Allow execution** or **chmod +x app.sh** in the terminal will get it done. For using it you will need to issue a **./app.sh** and done! :)

4.- FUTURE IMPROVEMENTS...

After this I hope this tutorial has piqued your curiosity and you are wondering what else can be done with just a bunch of functions linked together in the cloud...

One interesting step could be to port the example into *Fission Workflows* (a hint: you just need to **asynchronously** *module.export* the funtions ;) or use a more complex connector like **retain** and avoid the use of **showTodos.js** ;)

If you have any doubt, kudos, comments or suggestion to improve the tutorial please feel free to reach me by e-mail, I'll try to answer as fast as I can :)