Dummy Cell Phones Factory on Kubernetes - README File

# Content of this folder

[This](https://drive.google.com/drive/folders/19rI6DyKyEJXM2Sim-3VJem4IV3jqXn9S?usp=drive_link) folder contains all the \*.deployment.yaml and \*.service.yaml to be used for simulating the production line on any Kubernetes cluster. Once all pods and services have been created, you can use the [Postman collection](https://drive.google.com/file/d/1pqjVFLNmC3i81_RZepLynLWHNuNYZY-h/view?usp=drive_link) to run an example production. The Postman collection also serves as a guide about the available APIs and how they can be used.

**Remark**: when you change the implementation of some Node-RED node, get a console of the pod, create a tar of the /data folder with tar cvf /data/backup.tar /data then copy the archive from the pod to your local system and upload it *as a new version* to the appropriate tar archive in the [Dummy Cell Phones Factory folder](https://drive.google.com/drive/folders/1ZwxWOUYW9rtP-W-n1mWPJiOhVCpHXbac?usp=drive_link). This is important because the deployment files refer to those archives, so if you make modifications but don’t update the archives, and then create the simulation on a different Kubernetes cluster, what happens is that the modifications you made will not be present there.

# High level description of the simulation

You first have to create an order. There’s an API for this, and it writes on the dedicated database table. Then you have to add to the magazine the components that are necessary for producing the items of the type and quantity that is specified in the order. There’s an API for adding items to the magazine.

One order cannot include products of different types. Product types are smallcellphone or largecellphone. Associated components are smallbottomcover, smalltopcover, circuitboard for smallcellphones, and largebottomcover, largetopcoder, circuitboard for largecellphone.

Once you have created the order and the components, you can start the production. There’s an API for that. When you call it you have to specify in the request body the speed of the belt, and whether you want that the CustomIIoT device could modify the delt speed or not. You can open the web interface of the conveyor belt at <https://172.20.28.110:30017/ui> to see the different items flowing and being worked along the belt.

You can also display the web interface of the other software components that simulate the different machines dislocated along the belt, and the orders and magazine systems, just by putting the appropriate port number.

Once the production is started, what happens is that the ExecSystem checks the orders, checks the availability of components in the magazine, and gives instructions to the Loader to take the components from the magazine and put them on the belt. Then the Rotator rotates the bottom cover so that it has a rotation of zero degrees. Then the Drill drills the bottom cover. Then the industrial arm puts the circuit board in the bottom cover, so you will see that the two components come to have the same position on the belt after that they have passed in front of the industrial arm. Then the camera checks if the circuit board is well positioned. If it’s not, the bottom cover and circuit board are automatically removed from the belt (we can imagine the presence of some kind of “unloader” connected to the same PLC of the camera). The top cover will continue to travel alone along the belt. The human operator will have to take it out. There’s an API defined on the conveyor belt software component that can be used for this. It is the same that you can find in the Postman collection and that you use for taking out the finished product from the belt. After the check, if everything is okay, the assembler puts the top cover over the bottom cover and circuit board so that you will see all the three components in the same position on the belt after that they have passed in front of the assembler. In the end, the Press presses everything and the three components cease to be represented as three different components and finally become a single finished product of the appropriate type. The human operator will have to remove the finished product from the belt (there’s an API for this), and add it to the magazine (there’s a separate API for this). Adding finished products to the magazine is important otherwise the ExecSystem thinks they have to be produced and keeps working on the same order forever. Once all products for an order have been produced and put in the magazine, the order can be delivered. There’s an API for this. As a result, all the finished products related to that order will disappear from the magazine and the order will disappear from the orders.

# Make modifications to single parameters in any of the software components at runtime via APIs

The software components that simulate the different parts of the production line and also the orders and the magazine are Crazy Nodes. They are crazy because they expose APIs that you use to modify any parameter on any node that is inside of them. Learn more about [Crazy Nodes](https://github.com/mircosoderi/State-of-the-art-Artifacts-for-Big-Data-Engineering-and-Analytics-as-a-Service/blob/main/crazynode.tar) in [Soderi, Mirco, and John G. Breslin. "Crazy nodes: towards ultimate flexibility in ubiquitous big data stream engineering, visualisation, and analytics, in smart factories." International Symposium on Leveraging Applications of Formal Methods. Springer, Cham, 2022.](https://link.springer.com/chapter/10.1007/978-3-031-19762-8_18). You may also want to run the [updated version](https://github.com/mircosoderi/State-of-the-art-Artifacts-for-Big-Data-Engineering-and-Analytics-as-a-Service/blob/main/demov2.postman_collection.json) of the [demo](https://github.com/mircosoderi/State-of-the-art-Artifacts-for-Big-Data-Engineering-and-Analytics-as-a-Service/blob/main/demo.postman_collection.json) presented at [SmartComp 2022](https://ieeexplore.ieee.org/abstract/document/9821056?casa_token=uRQH9MUeL0gAAAAA:CejSwY8ZaQ261we__UA3FK14_WfMSfJllAX8AzQwB6zfcmAiklJyoSBbWxowDtsjL6cm1zEa8f_O); you can use the dedicated step by step [PowerPoint presentation](https://github.com/mircosoderi/State-of-the-art-Artifacts-for-Big-Data-Engineering-and-Analytics-as-a-Service/blob/main/demov2.pptx) for guidance. You can also replace the entire business logic that is implemented in a Crazy Node, taking the implementation from a Transformation Library. Learn more in the [GitHub repository](https://github.com/mircosoderi/State-of-the-art-Artifacts-for-Big-Data-Engineering-and-Analytics-as-a-Service).

# Contacts

Mirco Soderi, PhD (he/him/his)  
[#StartsWithAName](https://www.startswithaname.net/) Mirco (first n., 🔊 **miracle**), Soderi (🔊 **so**rry - **de**spite - **ri**thm)  
[Data Science Institute (DSI)](https://dsi.nuigalway.ie/) – Postdoctoral Researcher  
[CoSE EDI Committee](https://www.universityofgalway.ie/science-engineering/edi/cose-edi/) – Researchers Representative  
[University of Galway](https://www.universityofgalway.ie/)  
Phone: +353894916707  
Working hours: Mon – Fri, 9:00 – 17:00, [Irish time](https://en.wikipedia.org/wiki/Time_in_the_Republic_of_Ireland)  
Office: [DSI Building](https://goo.gl/maps/386as6vwNamwNj9u7), 1st Floor, South Wing, Desk 225