
Article

[Oliver Wilms](#) · Sep 11, 2022 2m read

[Open Exchange](#)

About interoperability-test app for unittesting interfaces in productions

I cloned iris-interoperability-template for my app interoperability-test. I wanted to develop an app to test interfaces in Interoperability productions.

I have never run a container in Docker Desktop. I always run containers in AWS. My first problem was that I got an error when I tried to build the image using docker-compose. The error was about BuildKit. I worked around the error by building the image with a docker build command relying on DOCKERBUILDKIT environment variable:

```
DOCKERBUILDKIT=1 docker build --progress=plain --no-cache --tag testint .
```

Next I updated docker-compose.yaml to use image testint instead of building an image.

My team was working on POC interface when I started this app. I updated the Demo Production to use POC interface. POC interface needs to interact with an external system (DMLSS) to get data for the response. First POC process sends an authorization request to DMLSS. DMLSS responds with authorization token. The token is used in a second request sent to DMLSS. The data from the second response is used to generate the response to the incoming request.

I wanted to test POC interface without being connected to DMLSS. So I created an Authorization process to simulate the interaction with DMLSS. I was able to make this work using Generic REST Service to receive the authorization request. I confused the port numbers and I got this to work using port 57700 which I thought was the web port but internally the web port is 52773. The problems began when I added a second Generic REST service to receive the second request. And I am still troubleshooting...

[#Contest](#) [#REST API](#) [#Ensemble](#) [#InterSystems](#) [IRIS](#)
[Check the related application on InterSystems Open Exchange](#)

Source

URL: <https://community.intersystems.com/post/about-interoperability-test-app-unittesting-interfaces-productions>