
Article

[Piyush Adhikari](#) · Oct 19, 2022 3m read

Utilizing InterSystems IRIS's PEX Framework and Python for Machine Learning with Interoperability using HuggingFace

I am demonstrating a use case of how we can create an IRIS Interoperability Production for special use in an external language. InterSystems IRIS, within Interoperability has a framework called Production Extension (PEX), using which we can create productions and program them as per their purpose using external languages like Java, Python etc, and also develop custom inbound and outbound adapters to communicate with other applications. Here in this demo, I will demonstrate a PEX framework-based production created by [@Guillaume Rongier](#) on Python, and I am using 'task-specific' business operations created by [@Lucas Enard](#) in that production to interoperate with a third party open source machine learning platform called HuggingFace, and bring its machine learning capabilities to InterSystems IRIS via PEX interoperability framework and Python.

Walkthrough video: <https://www.loom.com/share/239a15e8c510406faac1bcdea8030d1d>

Prerequisite

Docker and Docker Compose

Installation and testing

1. Clone the repository <https://github.com/LucasEnard/Contest-Sustainability>
2. Open the folder where the above repo is cloned, and open command prompt and start the docker container using the following command: `docker-compose up`
3. The container may start 'unhealthy' therefore, it is also a nice practice to start the container with the command: `docker-compose up -d`
4. Identify the port that the localhost is running on by using Docker command line and run the IRIS instance via the docker container
5. Open the production by clicking on the link as follows: <http://localhost:port/csp/irisapp/EnsPortal.ProductionConfig.zen?RODUCTION=INFORMATION.QuickFixProduction>
6. There are a few business operations, each for some HuggingFace models. Enable an operation and go to settings.
7. And on the 'Python' section, fill the boxes as per the model you would like to use.
8. Follow the content on the link as follows for reference: <https://github.com/LucasEnard/Contest-Sustainability#521-settings>
9. And click 'Apply'.
10. Go to 'Actions' and run a test.
11. In the dialogue box that next appears, select request type as Grongier.PEX.Message.
12. And, the classname as msg.MLRequest.
13. And the JSON must include arguments needed by the model.
14. In the content on the link as follows, there are JSON content for some models: <https://github.com/LucasEnard/Contest-Sustainability#522-testing>
15. Click on 'Invoke Testing Service' to see test results.
16. Click on 'Visual Trace' and click on messages to see the response and content.

Exception/s

One exception I came across while running this demo on my machine was the docker container starting as 'unhealthy'.

Adding the HEALTHCHECK statement as follows in the Dockerfile seems to fix the issue.

```
USER root
RUN apt-get update
USER ${ISC_PACKAGE_MGRUSER}

RUN \
  --mount=type=bind,src=src,dst=src \
  --mount=type=bind,src=iris.script,dst=/tmp/iris.script \
  --mount=type=bind,src=Installer.cls,dst=Installer.cls \
  iris start IRIS && \
  iris session IRIS < /tmp/iris.script && \
  iris stop iris quietly

HEALTHCHECK --interval=5m --timeout=2m --start-period=45s \
  CMD curl -f --retry 6 --max-time 5 --retry-delay 10 --retry-max-time 60 "http://localhost:8080/health" || bash -c 'kill -s 15 -1 && (sleep 10; kill -s 9 -1)'

ENV IRISUSERNAME "SuperUser"
ENV IRISPASSWORD "SYS"
ENV IRISNAMESPACE "IRISAPP"
```

References

1. Lucas, E. (2022). Contest-Sustainability. [online] GitHub. Available at: <https://github.com/LucasEnard/Contest-Sustainability> [Accessed 20 Sep. 2022].
2. Enard, L. (n.d.). Sustainable Machine Learning for the InterSystems Interoperability Contest. [online] InterSystems Developer Community. Available at: <https://community.intersystems.com/post/sustainable-machine-learning-intersystems-interoperability-contest> [Accessed 20 Sep. 2022].

[#Interoperability](#) [#Python](#) [#InterSystems IRIS for Health](#)

Source

URL: <https://community.intersystems.com/post/utilizing-intersystems-iriss-pex-framework-and-python-machine-learning-interoperability-using>