Article
Eduard Lebedyuk · Jun 3, 2021 7m read

Open Exchange

Running InterSystems Reports in containers

IMPORTANT NOTE InterSystems no longer provides a separate InterSystems Reports Server container. To run containerized InterSystems Reports Server, use Logi Reports Server container and your InterSystems Reports Server license. Documentation.

> InterSystems Reports is powered by Logi Report (formerly named JReport), a product of Logi Analytics. InterSystems Reports is supported by InterSystems IRIS and InterSystems IRIS for Health. It provides a robust modern reporting solution that includes:
>
> - Embedded operational reporting which can be customized by both report developers and end users.
> - Pixel-perfect formatting that lets you develop highly specific form grids or other special layout elements for invoices, documents, and forms.
> - Banded layouts that provide structure for aggregated and detailed data.
> - Exact positioning of headers, footers, aggregations, detailed data, images, and sub-reports.
> - A variety of page report types.
> - Large-scale dynamic report scheduling and distribution including export to PDF, XLS, HTML, XML, and other file formats, printing, and archiving for regulatory compliance.
>
> InterSystems Reports consists of:
>
> - A report designer, which provides Design and Preview Tabs that enable report developers to create and preview reports with live data.
> - A report server which provides end users browser-based access to run, schedule, filter, and modify reports.

From InterSystems documentation.

This article focuses on the Server part of InterSystems Reports and provides a guide on running Report Server in containers while persisting all the data.

Prerequisites

Before we start, this software must be available for the InterSystems Reports to work:

- Docker - while InterSystems Reports can work *without* Docker, this article focuses on Dockerised setup.
- (Optional) git - to clone this repo, otherwise download it as an archive.
- (Optional) InterSystems Reports Designer - to create new reports if desired.

Additionally, you'll need:
- Login on containers.intersystems.com Docker registry
- InterSystems Reports License (contact InterSystems for it)

Configuration

Before we start, here's what we're going to do:

- First, we are starting Reports and IRIS in setup mode to setup IRIS as a database (not DataSource!) for Reports.
- After that, we are configuring Reports and persisting this configuration on the host.
- Finally, we are running Reports with persisted data.

First start

Let's go. Note that all steps here - 1-8 use docker-compose_setup.yml as a docker-compose configuration file. All additional docker-compose commands during these steps must be run as docker-compose -f docker-compose_setup.yml.

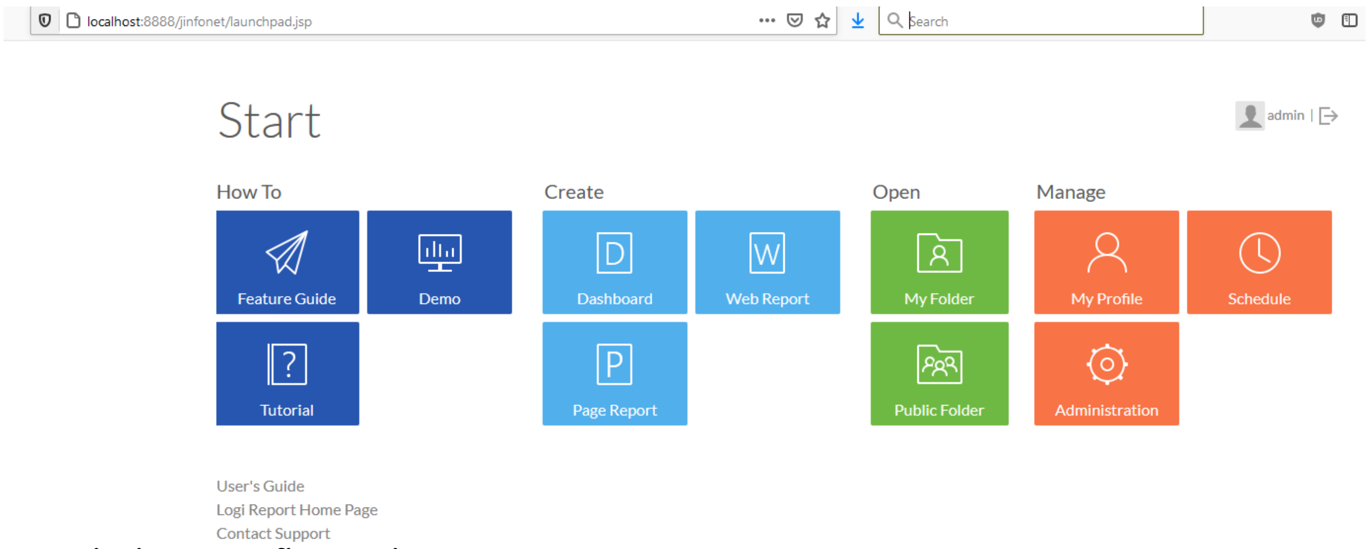
1. Clone this repo: git clone https://github.com/eduard93/reports.git or download an archive.

2. Edit config.properties and specify your InterSystems Reports Server license information (User and Key). If you don't have them - contact InterSystems. There are many other properties described in the documentation. Note that IRIS, in that case, refers to the database for Reports and not the data source for reports (which comes later).

3. Start InterSystems Reports Server with initialization: docker-compose -f docker-compose_setup.yml up -d

4. Wait for InterSystems Reports Server to start (check with docker-compose -f docker-compose_setup.yml logs reports). It can take 5-10 minutes. Reports Server is ready for work when logs show: reports_1 ¦ Logi Report Server is ready for service.

5. Open Reports Server. (User/pass: admin/admin). In a case, it shows an expired window enter the same license info again. It should look like this:



Persisting configuration

Now that Reports is running, we need to adjust configuration a little and persist it on a host (note that InterSystems IRIS part of a configuration is persisted using Durable %SYS.

6. Check Enable Resources from Real Paths option in the server console > Administration > Configuration > Advanced page. Docs. It would allow us to publish reports as simple as copying them into the reports folder in the repository.

7. Copy persistent storage files to host (docs):

```
docker cp reports_reports_1:/opt/LogiReport/Server/bin .
docker cp reports_reports_1:/opt/LogiReport/Server/derby .
docker cp reports_reports_1:/opt/LogiReport/Server/font .
docker cp reports_reports_1:/opt/LogiReport/Server/history .
docker cp reports_reports_1:/opt/LogiReport/Server/style .
```

8. Shutdown InterSystems Reports Server: docker-compose -f docker-compose_setup.yml down

Second start

Now we're ready to start Reports with persisted data storage - this is how it would run in production.

9. Start InterSystems Reports Server without initialization: docker-compose up -d

10. Create a new folder resource in Public Reports with Real Path: /reports. Docs. To do that open Public Reports and select Publish > From Server Machine:

Create a new folder pointing to /reports:

It should contain a catalog (which defines a connection to IRIS) and two reports (reportset1 and reportset2). Run them (use Run button to see it in a browser and Advanced Run to choose between HTML, PDF, Excel, Text, RTF, XML, and PostScript formats). Here's what reports look like:

As you can see, Reports supports Unicode out of the box. In this example, I'm using the same IRIS as a data source, but in general, it can be any other IRIS instance - as defined in a catalog. This demo uses the HoleFoods dataset (installed with zpm "install samples-bi"). To add new connections, create a new catalog in Designer. After that, create new reports and export everything in a new subfolder in a reports folder. Of course Server container must have network access to any data source IRIS instance.

That's it! Now, if you want to stop Reports, execute: docker-compose stop. And to start Reports again execute: docker-compose up -d. Note that all reports are still available.

Debugging

All logs are stored in /opt/LogiReport/Server/logs folder. In a case of errors, add it to volumes, restart Reports and reproduce the error.

Documentation describes how to adjust log levels. If Reports doesn't exactly get to the UI adjust LogConfig.properties file located in the bin folder:

```
logger.Engine.level = TRIVIAL
logger.DHTML.level = TRIVIAL
logger.Designer.level = TRIVIAL
logger.Event.level = TRIVIAL
logger.Error.level = TRIVIAL
logger.Access.level = TRIVIAL
logger.Manage.level = TRIVIAL
logger.Debug.level = TRIVIAL
logger.Performance.level = TRIVIAL
logger.Dump.level = TRIVIAL
```

Embedding and APIs

To embed reports in your web application, use Embedded API.
Other available APIs.

Summary

InterSystems Reports provides a robust modern reporting solution with embedded operational reporting.
InterSystems Reports Server provides end users browser-based access to run, schedule, filter, and modify reports.
InterSystems Reports Server can be efficiently run in a Docker environment.

Links

- Repository
- Documentation
- Logging

#Docker #System Administration #InterSystems IRIS
Check the related application on InterSystems Open Exchange

Source URL:https://community.intersystems.com/post/running-intersystems-reports-containers