

文章

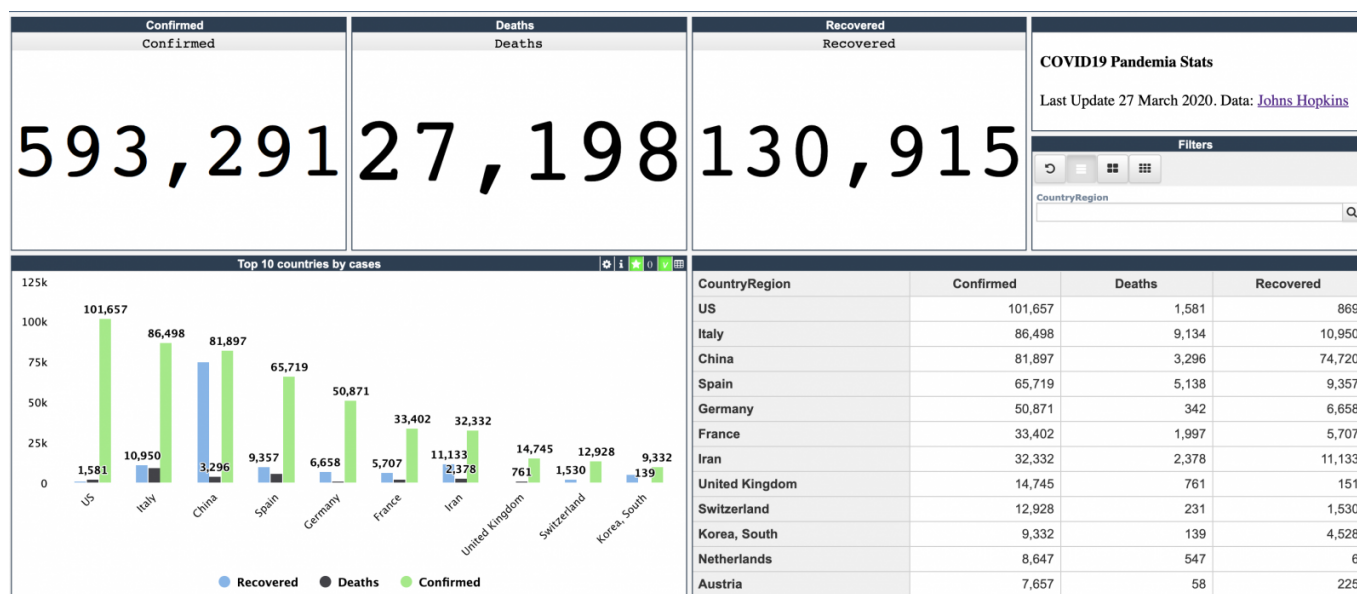
[Michael Lei](#) · 五月 8, 2021 阅读大约需 5 分钟[Open Exchange](#)

InterSystems IRIS 上的 COVID-19 分析

大家好！

约翰·霍普金斯大学每天都会发布有关 COVID-19 疫情的新数据。

我在部署于 GCP Kubernetes 上的 docker 中使用 InterSystems IRIS Community Edition 构建了一个[简单的 InterSystems IRIS Analytics 仪表板](#)，可显示疾病爆发的关键指标。



这个仪表板的示例说明了如何使用 IRIS Analytics 分析 CSV 中的信息并以 InterSystems IRIS Community Edition 的形式将其部署到 GCP Kubernetes。

添加了[美国的交互式地图](#)：

[下一个仪表板显示时间线](#)：

并且可以按国家/地区筛选。例如，这是[在美国](#)：

应用程序的源代码在 [Open Exchange](#) 上提供。

工作原理

演示使用 InterSystems IRIS Community Edition docker 容器运行，并通过 MDX2JSON REST API 使用 DeepSee Web 表示层公开 InterSystems IRIS BI 仪表板。它部署在 GCP 上，使用 Google Kubernetes Engine (GKE) 运行。

开发方式

数据以 CSV 格式提取自[约翰·霍普金斯仓库](#)。

类、多维数据集和初始数据透视通过 [Analyzethis](#) 模块生成，感谢 [@Peter Steiwer](#)！

随后使用 CSVTOCLASS 方法引入导入方法，感谢 [@Eduard Lebedyuk](#)！

使用 [DeepSee Web \(DSW\)](#) 模块渲染仪表板。

IRIS BI [项目（枢轴、仪表板）](#) 由 [ISC.DEV](#) 模块导出：

```
IRISAPP> d ##class(dev.code).workdir("/irisdev/app/src")
```

```
IRISAPP> d ##class(dev.code).export("*.dfi")
```

代码采用 [VSCode ObjectScript](#) 开发，感谢 [@Dmitry Maslennikov](#)。

为开发和部署构建 docker 镜像

所有部署均位于 [Dockerfile](#)。我们通过 Dockerfile 构建了一个包含数据、Web 应用和模块且正确设置的镜像，然后将映像部署到 GCP Kubernetes。

这个 Dockerfile 是 [Dockerfile 模板](#) 的修改版，[本文](#) 对其进行了很好的描述。

所以，我只会在新的部分停留。

```
17 COPY data files
```

在这里，我们将 CSV 文件从数据文件夹复制到容器

```
set pfile = "/opt/irisapp/files/covid-$_str($zd($h-1),"/","-")_.csv", rc=0 \
do ##class(AnalyzeThis.Generated.covid03162020).Import(,pfile,",", ,1,.rc) \
write "imported records: "_rc \
do ##class(%DeepSee.Utills).%BuildCube("covid03162020") \
```

这段代码会导入最新数据并构建多维数据集。

```
zpm "install dsw" \
```

安装 DeepSee Web。

```
do EnableDeepSee^%SYS.cspServer("/csp/irisapp/") \
```

这会为 /csp/irisapp web 应用启用 IRIS Analytics (DeepSee)。

```
zn "%SYS" \
write "Modify MDX2JSON application security...","! \
set webName = "/mdx2json" \
set webProperties("AuthEnabled") = 64 \
set webProperties("MatchRoles")=":%DB_IRISAPP" \
set sc = ##class(Security.Applications).Modify(webName, .webProperties) \
if sc<1 write $SYSTEM.OBJ.DisplayError(sc)
```

需要这段代码来使分析 Web 应用无需凭据即可使用。

```
COPY irisapp.json /usr/irissys/csp/dsw/configs/
```

这条命令可以帮助设置 DSW 配置。

部署到 Kubernetes

部署过程由 GitHub Actions 处理 - 此[工作流](#)会在每次提交到仓库时处理。

GitHub 工作流使用我们在上一步构建的 Dockerfile 以及 [Terraform](#) 和 [Kubernetes](#) 设置。

该过程与 [@Mikhail Khomenko](#) 在[这篇文章](#)中描述的过程相同。

如何在本地运行和开发

欢迎您运行和开发此项目或与此项目展开协作。

要在本地运行，请使用 docker：

将仓库克隆/git pull 到任意本地目录

```
$ git clone https://github.com/intersystems-community/objectscript-docker-template.git
```

在此目录中打开终端，然后运行：

```
$ docker-compose build
```

3. 运行 IRIS 容器：

```
$ docker-compose up -d
```

构建并运行容器后，在以下位置打开应用程序：

```
localhost:yourport/dsw/index.html#!/d/Covid19/Countries.dashboard?ns=IRISAPP
```

如何开发

此仓库可以通过 ObjectScript 插件在 VSCode 中编码。安装[VSCode](#)、[Docker](#) 和 [ObjectScript](#) 插件并在 VSCode 中打开文件夹。

如何贡献

创建[仓库](#)分支，进行更改并发送拉取请求。 [观看视频了解更多信息](#)。

期待您的贡献！

[#CSV](#) [#Docker](#) [#GCP](#) [#Kubernetes](#) [#ObjectScript](#) [#InterSystems Package Manager \(IPM\)](#) [#分析](#) [#InterSystems IRIS](#) [#InterSystems IRIS BI \(DeepSee\)](#)
[在 InterSystems Open Exchange 上检查相关应用程序](#)

源

URL:<https://cn.community.intersystems.com/post/intersystems-iris-%E4%B8%8A%E7%9A%84-covid-19-%E5%88%86%E6%9E%90>