Article
Peter Steiwer · Jan 14, 2020 2m read

Open Exchange

Portlets that use data from outside of DeepSee

In the previous part of this series, we saw how to include data in a portlet from within DeepSee. This used the built in data controller. In this part, we are going to be pulling in data from outside of DeepSee. This will include both information from within InterSystems IRIS and from the OS.

Why use this?

This is useful if you would like to create a dashboard that only contains information about your system. It is also useful if you want to display data about your system along side data that you have stored in DeepSee.

What will we learn?

In this part we will learn different ways of calling out of the portlet to gather the information we need.

Implementation

For this implementation, we will start with the portlet code from Part 1.

1) First we will add a new method to get the last message from either the cconsole.log or messages.log

```
Method GetLastMessage() As %String [ ZenMethod ]
{
}
```

We have created this as a ZenMethod so that it can be called from our renderContents method, which is written in javascript.

2) Now that we have defined this method, we can add some ObjectScript that interacts with files on the OS

```
Method GetLastMessage() As %String [ ZenMethod ]
{
    Set tFile=##class(%Stream.FileCharacter).%New()
    If ##class(%File).Exists(##class(%File).ManagerDirectory()_"messages.log") {
        Do tFile.LinkToFile(##class(%File).ManagerDirectory()_"messages.log")
    } ElseIf ##class(%File).Exists(##class(%File).ManagerDirectory()_"cconsole.log")
{
        Do tFile.LinkToFile(##class(%File).ManagerDirectory()_"cconsole.log")
    }
    Set tLine=tFile.ReadLine()
    While 'tFile.AtEnd {
        Set tLine=tFile.ReadLine()
    }
    Quit tLine
}
```

This code will check to see if either messages.log or cconsole.log exists and will output the last line of the file

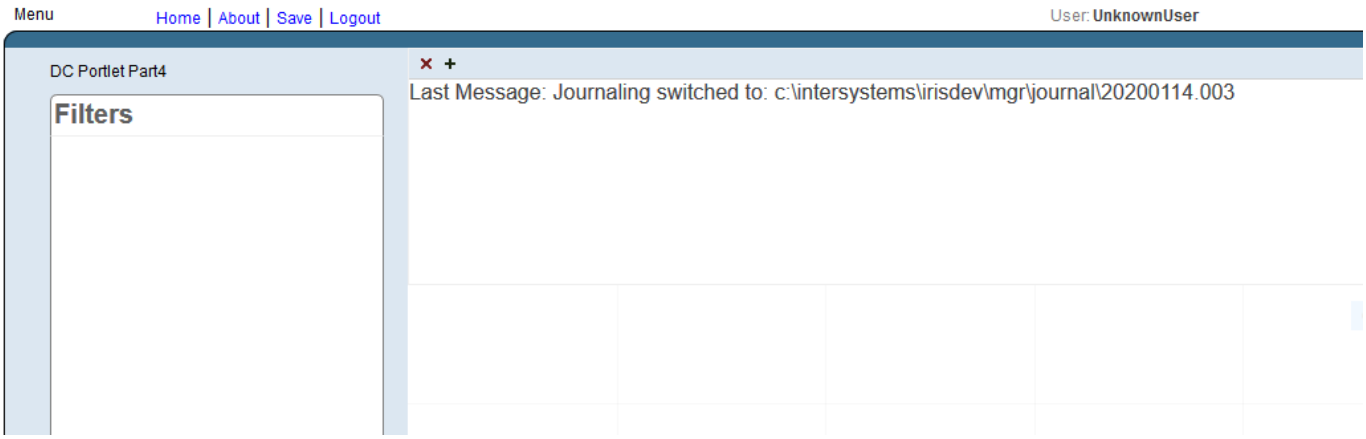
3) We are now going to modify our renderContents method to use the response from this method. We will be replacing the following line

```
html[html.length]="Testing"
```

And it will become

```
var lastMessage=this.GetLastMessage()
html[html.length]="Last Message: "+lastMessage
```

4) Lets see how our dashboard looks now



As you can see here, we can write ObjectScript to pull in any information we may need in our dashboard. This information can be in addition to DeepSee data, or it can be the only data in your dashboard.

Instead of just pulling in text, I could pull in other information about my system and then use charts to visualize the data

The associated Portlet and Dashboard can be found on Open Exchange

#Analytics #Dashboards #UI Development #Caché #InterSystems IRIS #InterSystems IRIS BI (DeepSee)
Check the related application on InterSystems Open Exchange

Source URL:https://community.intersystems.com/post/portlets-use-data-outside-deepsee