Embedding SQL Server Reporting Services (SSRS) reports within an iframe in a Streamlit application can be a bit tricky due to potential security restrictions like cross-origin resource sharing (CORS) and SSRS's own configuration options. However, you can generally achieve this by following these steps:

**Steps to Embed SSRS Reports in an iframe in Streamlit**

1. **Set up SSRS report to be embeddable in an iframe**:
   * Ensure your SSRS instance allows embedding in iframes. This requires setting certain configurations on the SSRS server:
     + Edit the rsreportserver.config file, typically located in C:\Program Files\Microsoft SQL Server\MSRSXX.MSSQLSERVER\Reporting Services\ReportServer.
     + Add or modify the <Add Key="SecureConnectionLevel" Value="0"/> line to allow HTTP connections (or set up proper HTTPS if necessary).
     + Also, ensure that the following in the RSSrvPolicy.config file allows embedding:

xml

Copy code

<CodeGroup class="UnionCodeGroup" version="1" PermissionSetName="Execution" Name="ReportServer\_CodeGroup" Description="This code group grants code from the report server site the necessary permissions to run.">

<IMembershipCondition class="UrlMembershipCondition" version="1" Url="http://your-server/ReportServer/\*" />

</CodeGroup>

1. **Enable CORS on the SSRS Server**:
   * If your Streamlit app is hosted on a different domain or port than your SSRS server, you might encounter CORS issues. You need to allow CORS on your SSRS server.
   * You can set up CORS rules in your web.config or on your Windows Server via IIS to allow access from different origins.
2. **Generate the URL for the SSRS report**:
   * SSRS reports can be embedded using the report URL:

perl

Copy code

http://<Your-SSRS-Server>/ReportServer/Pages/ReportViewer.aspx?<Your-Report-Path>

For example:

ruby

Copy code

http://yourserver/ReportServer/Pages/ReportViewer.aspx?/Reports/MyReport&rs:Embed=true

The rs:Embed=true parameter ensures the report viewer is shown without the SSRS toolbar.

1. **Create the iframe in Streamlit**: Use Streamlit's components.html to embed the iframe with the SSRS report. You can modify the URL for your specific report and server.

Here’s a sample Streamlit script:

python

Copy code

import streamlit as st

# Define the SSRS report URL

report\_url = "http://your-ssrs-server/ReportServer/Pages/ReportViewer.aspx?/Reports/MyReport&rs:Embed=true"

# Embed the SSRS report in an iframe using Streamlit's HTML components

iframe\_code = f"""

<iframe width="100%" height="800px" src="{report\_url}" frameborder="0" allowfullscreen></iframe>

"""

# Render the iframe

st.components.v1.html(iframe\_code, height=800)

1. **Troubleshooting**:
   * **CORS Errors**: Make sure the SSRS server has proper CORS headers. You might need to modify IIS settings for CORS.
   * **Mixed Content Issues**: If your Streamlit app is served over HTTPS, but SSRS is served over HTTP, the browser might block the iframe due to mixed content. Either serve both over HTTPS or configure the browser to allow mixed content.

By following these steps, you should be able to embed SSRS reports in an iframe within a Streamlit application running on Windows Server 2019.

**You said:**

how configure my CORS headers

**ChatGPT said:**

ChatGPT

To configure CORS (Cross-Origin Resource Sharing) headers on your Windows Server 2019, which hosts your SSRS (SQL Server Reporting Services), you need to modify the IIS (Internet Information Services) settings or the web server configuration where SSRS is hosted. Here’s how you can set it up:

**Steps to Enable and Configure CORS in IIS on Windows Server 2019**

**1. Open IIS Manager:**

* Go to the server where your SSRS is hosted.
* Open **Internet Information Services (IIS) Manager**.

**2. Select Your SSRS Website:**

* In the IIS Manager, on the left side, expand the node for your server.
* Find your SSRS site (e.g., ReportServer or Reports), and select it.

**3. Add CORS Headers via HTTP Response Headers:**

* In the central panel, double-click on the **HTTP Response Headers** option.
* On the right-hand side, click **Add** to insert new headers.
* Add the following headers:

**Header Name**: Access-Control-Allow-Origin  
**Header Value**: \* (or specify your allowed domain, e.g., http://your-streamlit-app-domain)

You can use \* to allow requests from any domain, or specify the exact domain(s) allowed to access your SSRS reports (which is more secure).

For example, to allow only http://your-streamlit-app.com, use:

arduino

Copy code

Access-Control-Allow-Origin: http://your-streamlit-app.com

* Click **OK** to apply the changes.

**4. Add Additional CORS Headers (Optional):**

Depending on the level of control you need, you may need to add more headers:

* To allow specific methods (GET, POST, etc.), add:

makefile

Copy code

Access-Control-Allow-Methods: GET, POST, OPTIONS

* To allow custom headers in requests, add:

mathematica

Copy code

Access-Control-Allow-Headers: Content-Type, Authorization

* To allow credentials (cookies, HTTP authentication), add:

yaml

Copy code

Access-Control-Allow-Credentials: true

**5. Restart IIS:**

* After adding the required headers, restart IIS to ensure the changes take effect.
* You can do this via the command line by running:

bash

Copy code

iisreset