Https Configuration with Domain and Certificate

Document Information

Software Version:	4.0.4.0 patch2
Creation Date:	19/01/2024
Author:	Fabio Carvalho
Editor:	Fabio Carvalho
Last Edit Date:	19/01/2024
Version:	0.1

ADISRA · 3432 Greystone Drive, Suite 125 · Austin, TX 78731
Phone: 1-833-5ADISRA (1-833-523-4772)
www.ADISRA.com

1. Scope

This document details different steps to allow an ADISRA SmartView application to be published on the web using ssl certificate and a registered domain on a server computer under a LAN.

2. Summary

The following chapters contain information about the Web Server, Domain provider, ZeroSSL certificate, Router port forwarding configuration and different tests that can be performed to make sure different parts are working properly.

3. Architecture

Before diving into more details, let's understand the architecture of ADISRA SmartView when serving web pages in https in this sample application.

To serve pages on http on the web, the user needs to perform some simple actions:

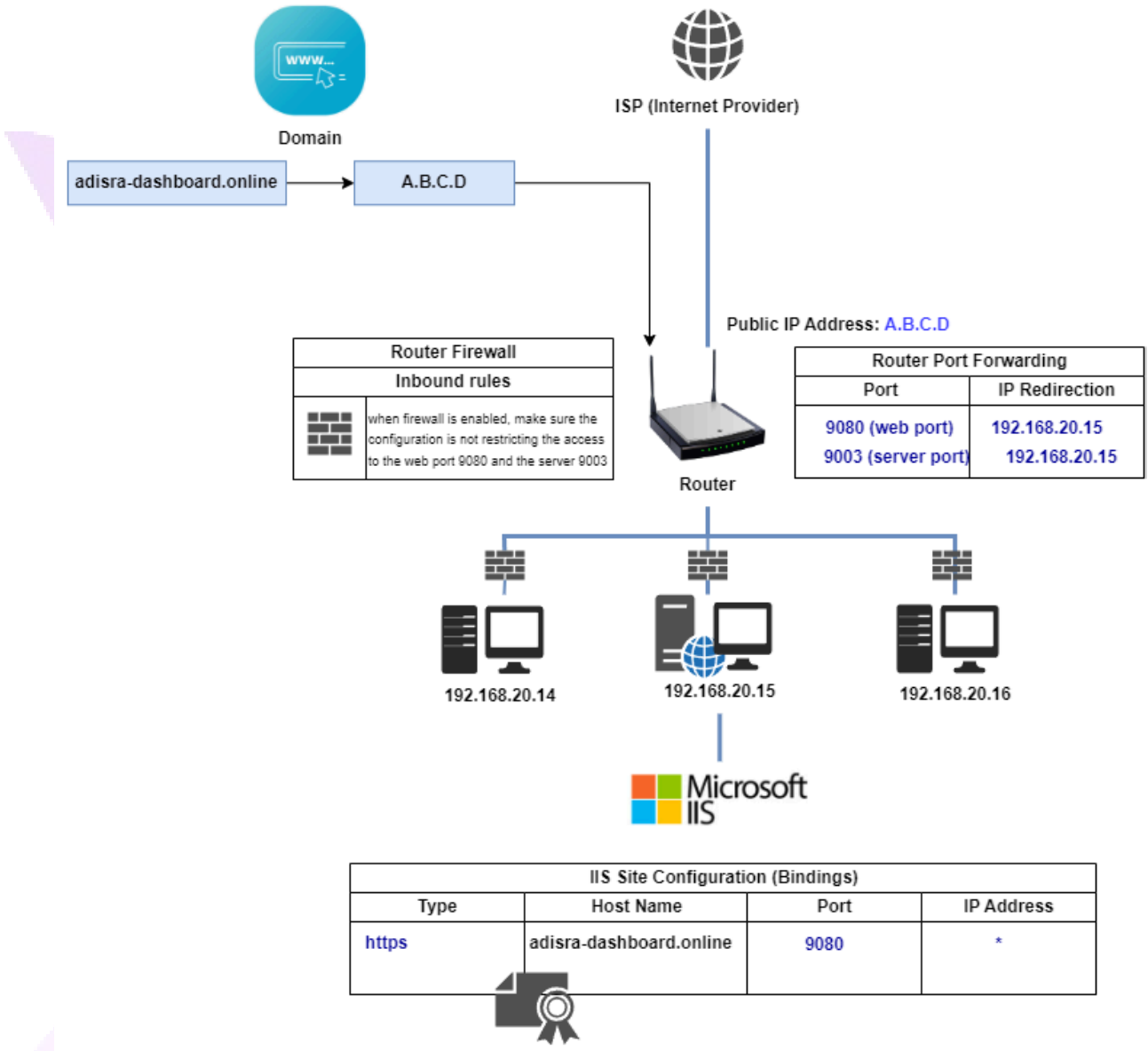
1- Create a sample application

2- Save the Graphics to the web using the engineering interface.

3- Configure a Web Server to host the web pages.

To use https, there are some extra steps that will be addressed in this document.

The image below shows all elements of the architecture and includes some important details to understand the relationship between these different elements.



The following sub-chapters will detail the following elements of the architecture:

- Domain provider
- ISP
- LAN
- Web Server

- ADISRA SmartView

With this architecture, we expect that the user can open the ADISRA SmartView Web application using any html5 web browser using the domain "adisra-dashboard.online" instead of using a public ip address with secure https access.

https://adisra-dashboard.online:9080/Main.html

The url above is the final goal.

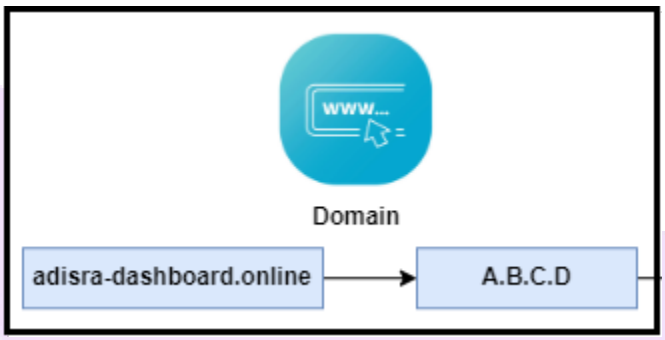
It means we will need to enable https on our Web Server and on ADISRA SmartView which also acts as a Web Api Server, we also need to somehow define port 9080 as https listening port and we need to make sure that any request coming from outside my LAN will reach my server hosting the web pages and hosting the Web Api Server and not any other computer on the LAN.

For https communication, we will also need a valid certificate.

"Certificates play a crucial role in securing websites, especially those using the HTTPS protocol. HTTPS encrypts data exchanged between a user's browser and the website, ensuring privacy and protecting against unauthorized access. Certificates, specifically SSL/TLS certificates, authenticate the identity of the website, assuring users that they are connecting to the legitimate site and not a malicious one. Certificate Authorities (CAs) issue certificates after verifying the authenticity of the website owner. While CA-issued certificates offer a high level of trust, some individuals or small businesses may opt for self-signed certificates due to cost considerations. Self-signed certificates, however, lack the third-party verification provided by CAs, potentially leading to security warnings for users. Services like ZeroSSL offer a middle ground by providing free CA-signed certificates, making it easier for smaller entities to secure their websites while maintaining a higher level of user trust compared to self-signed alternatives."

Let's see those elements in the following sub-chapters:

3.1. Domain Provider



Overview:

A domain provider, also known as a domain registrar, is a company or organization that manages the registration of domain names on the internet. Domain names serve as human-readable addresses for websites, translating easily memorable names into numeric IP addresses that computers use to identify each other on the internet.

Domain providers offer services that allow individuals and businesses to register, renew, and manage their domain names. These services often include domain name registration, domain renewal, DNS (Domain Name System) management, and sometimes additional features like email services, website hosting, and security options.

Popular domain providers include companies like GoDaddy, Namecheap, Google Domains, and many others. When you register a domain with a provider, you essentially lease the right to use that domain name for a specific period, typically on an annual basis. It's essential to choose a reliable domain provider as they play a crucial role in ensuring the availability, security, and proper functioning of your online identity.

Tips:

In this sample we used GoDaddy with the registered domain “adisra-dashboard.online”. But after registering a domain, we still need a service from the Domain Provider that makes sure any requests to “adisra-dashboard.online” will be forwarded to the web server. In case you have the public ip address, you can proceed to configure it on the domain dns configuration, but if you are not sure the public ip

address is correct or if it will change, please refer to the ISP sub-chapter.

Forwarding domain to public ip address:

This service mentioned above is called "Domain Forwarding" or "Domain Redirection." This service allows you to redirect incoming requests to your domain to a specific public IP address where your web server is hosted. Here's how it generally works:

Domain Registration: First, you need to register your domain with a domain provider or registrar. During the registration or within the domain management settings, you'll find an option for domain forwarding or redirection.

Configuration: In the domain management settings, look for the option related to domain forwarding or redirection. Different domain providers might have slightly different interfaces, but the general idea is to find the forwarding settings.

Specify IP Address: Once you're in the forwarding settings, you'll usually be asked to provide the specific public IP address to which you want to redirect the incoming requests. Enter the IP address of your web server.

Choose Redirect Type: Some domain providers allow you to choose the type of redirect. Common options include a 301 (permanent) or 302 (temporary) redirect. A 301 redirect is typically used when you want to permanently direct all traffic from one domain to another.

Save Changes: After entering the necessary information, save your changes. The domain provider will then update its DNS records to reflect the new configuration.

Propagation: Keep in mind that changes to DNS settings may take some time to propagate across the internet. During this period, some users may still be directed to the old IP address until the new DNS information fully spreads.

By configuring domain forwarding in this manner, anyone who enters your domain in their browser will be automatically redirected to the specified IP address where your web server is hosted. This is a convenient way to manage traffic and ensure that users reach the

correct web content, especially if your web server is hosted on a different IP address than your domain registrar's default settings.

3.2. ISP

Summary:

Internet Service Providers (ISPs) play a crucial role in delivering stable and reliable connectivity to users, and setting a static public IP address is one of the key aspects in this regard. A static public IP provides a fixed address that facilitates consistent remote access to devices or services hosted on a network. This is particularly important for businesses or individuals running servers, webcams, or any services requiring external connectivity. Moreover, the release of access to all ports is essential for ensuring that various applications and services can seamlessly operate without restrictions. However, in some cases, ISPs implement Carrier-Grade Network Address Translation (CGNAT) to address the shortage of available IPv4 addresses. While CGNAT enables multiple users to share a single public IP, it can complicate certain applications and services that require direct external access. In such scenarios, a static public IP and unrestricted port access become even more critical, enabling users to efficiently manage their online presence and services without hindrance.

Tips:

While building this application, the ISP consumed a big amount of time. Initially, the public ip address returned by

"https://www.whatismyip.com/" couldn't be reached from any computers (in/outside my network). When checking on my router's configuration page, the public ip address displayed there was different from the one returned by whatsmyip page. After contacting the ISP multiple times, they released all ports to the router and also changed the dynamic ip address to a static ip address. That internet service provider also uses CGNAT and it was disabled after my request.

Consequently, I was able to ping the router's public ip address and I could confirm on my router's configuration page that this was the only public ip address.

```
C:\Users\fabio>ping 201.80.21.70

Disparando 201.80.21.70 com 32 bytes de dados:
Resposta de 201.80.21.70: bytes=32 tempo<1ms TTL=64
Resposta de 201.80.21.70: bytes=32 tempo<1ms TTL=64
Resposta de 201.80.21.70: bytes=32 tempo<1ms TTL=64
Resposta de 201.80.21.70: bytes=32 tempo<1ms TTL=64
```

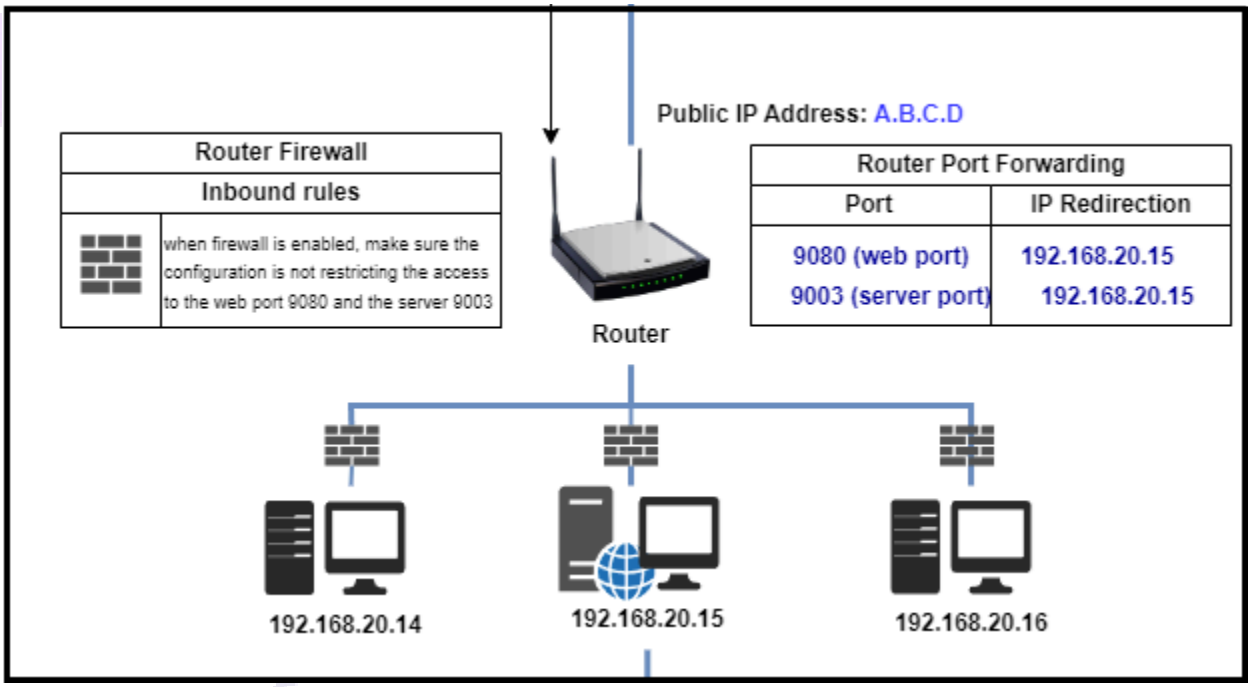
*replace the above ip with your public ip address.

The ability to ping the public IP address of a router provided by an Internet Service Provider (ISP) can vary, and there is no universal rule that applies to all ISPs. Some ISPs may allow ping requests to the public IP address of the router, while others may block them for security reasons.

There was also an advanced configuration on the router's page called "Block WAN" that had to be disabled. You must confirm that your router is configured properly, its configurations vary from hardware to hardware, so it is important to check the entire configuration.

After making sure the public ip address can be reached, you can configure your registered domain to forward any requests to this public ip address. At this point we can already reach the router, but we still need to reach the server on the LAN that will host the web app. If you are ready to move to that, please refer to the LAN sub-chapter below.

3.3. LAN



Summary:

The image above represents a small LAN with a router and 3 computers connected and their network addresses. The computer in the center (192.168.20.15) is the web server.

Router:

The router is connected to the ISP internet service and it can be configured in 192.168.20.1 address. The router plays an important role in the architecture since it allows redirecting requests to the web server when it matches pre-defined rules. In this sample application, we have defined port **9080** for https access and the web api that runs on the ADISRA SmartView runtime we have defined port **9003**, so we will use the router port forwarding to redirect any requests on those ports to the computer 192.168.20.15.

Service Name	Ext.Port Begin	Ext.Port End	Protocol	In.Port Begin	In.Port End	Client Address	WAN Interface	Remove
ADSV-9080	9080	9080	TCP	9080	9080	192.168.20.15	wanbridge	☐
ADSV-9080	9080	9080	UDP	9080	9080	192.168.20.15	wanbridge	☐
ADSV-9080	9003	9003	TCP	9003	9003	192.168.20.15	wanbridge	☐
ADSV-9080	9003	9003	UDP	9003	9003	192.168.20.15	wanbridge	☐

**Forwarding port 9080 to 192:168.20.15:9080*
**Forwarding port 9003 to 192:168.20.15:9003*

Url example containing port 9080:

https://adisra-dashboard.online:9080/Main.html

Url example containing port 9003:

https://adisra-dashboard.online:9003/security/CheckServer

**this url is used internally by ADISRA SmartView.*

You can use https://reqbin.com/ to test if the above service "CheckServer" is available on the runtime server. For more information, please refer to the following chapters. First it is important to guarantee the Domain and the ISP are correctly configured and that the correct certificate is also properly included in the architecture.

3.4. Web Server

Summary:

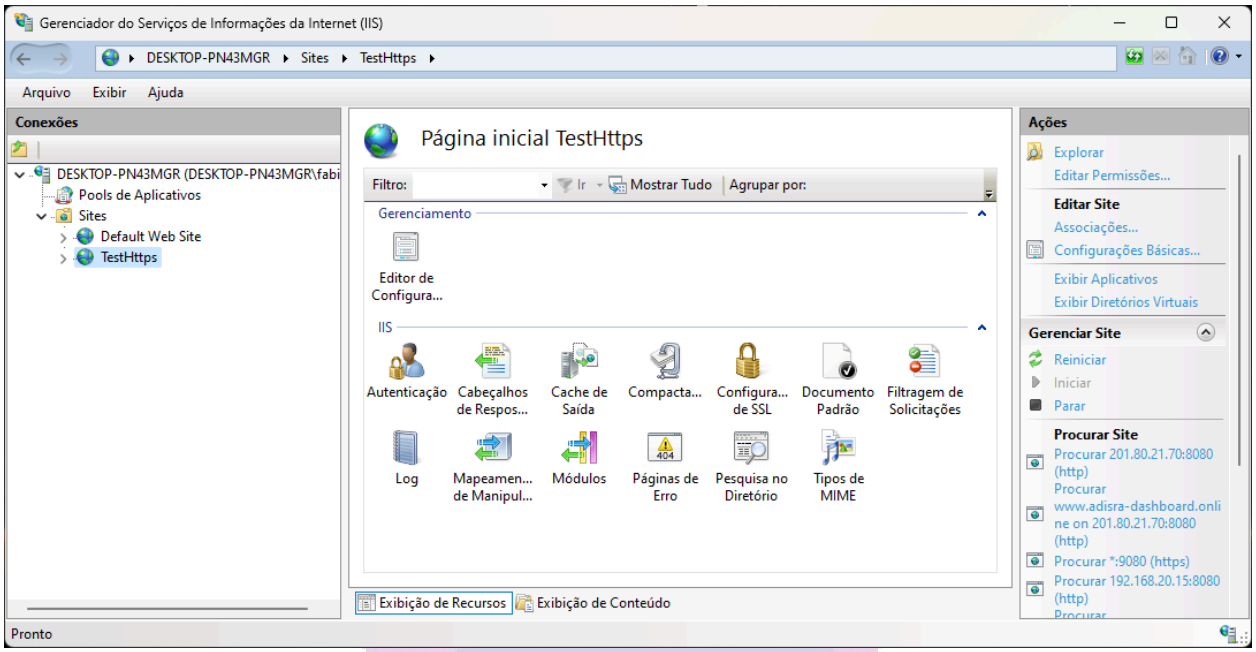
In this sample application we used IIS. IIS (Internet Information Services) is a web server developed by Microsoft for hosting and managing websites and web applications on Windows servers. It provides a platform for serving HTTP and HTTPS content, as well as other web-related services.

In a web server architecture, users have the flexibility to choose from various web server software, with Microsoft Internet Information Services (IIS) being one prominent option among several available. Alternatively, users can opt for web servers like Apache, Nginx, or others, depending on their specific needs and preferences. A web server plays a pivotal role in managing and responding to incoming requests from clients, typically web browsers, by delivering the requested web pages or resources. It acts as an intermediary between the user's device and the web application, ensuring the seamless transfer of data. The chosen web server is responsible for processing dynamic content, executing server-side scripts, handling security protocols, and managing various internet protocols. Each web server software has its strengths and features, making the selection crucial in determining the performance, security, and scalability of a web application. Ultimately, the user's choice of a web server contributes significantly to the efficiency and reliability of their web hosting environment.

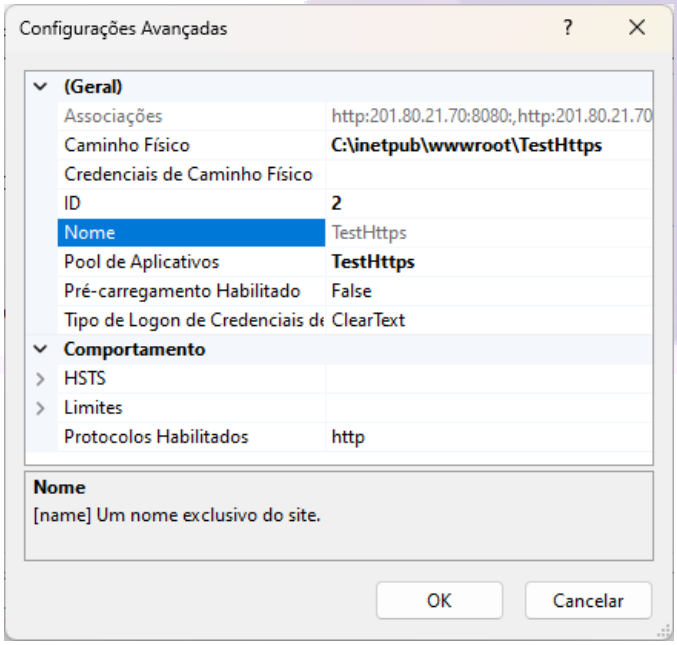
Tip 1 (Creating a new Site on IIS):

In this sample we used IIS to host our web pages. By default, ADISRA SmartView already saves the pages to the c:/inetpub/wwwroot folder

which is the default web folder for IIS, but I created a new Site called "TestHttps" on IIS to represent this sample test.



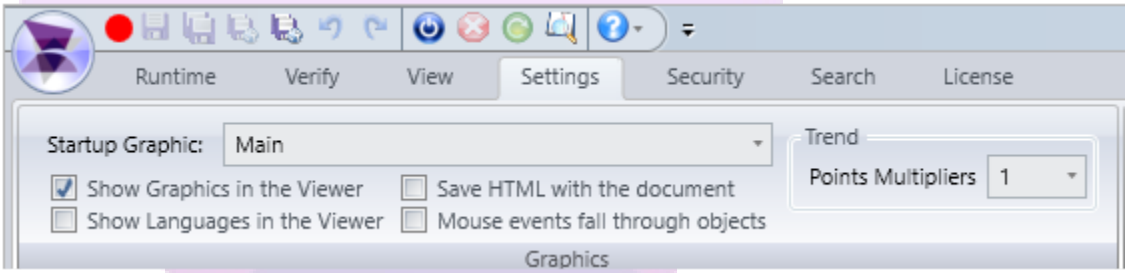
And this new Site is configured with the physical path of the application web folder on "c:/inetpub/wwwroot/TestHttps" (image below)



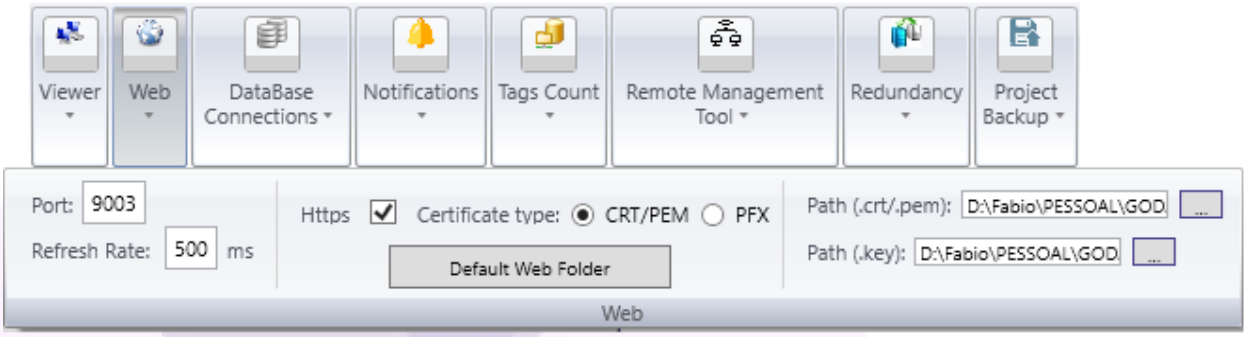
The physical path above matches the ADISRA SmartView web application folder.

In case you need to confirm what's the current web folder, please follow the instructions below:

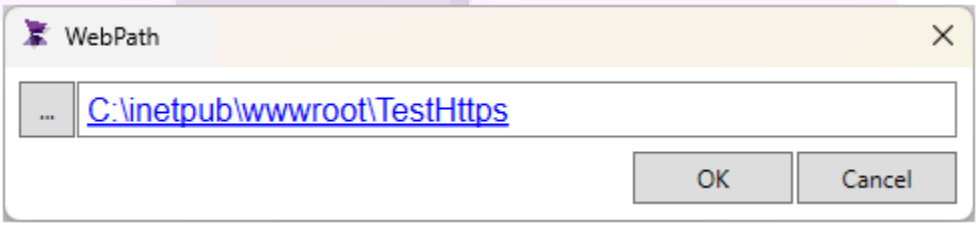
1. Open the Settings configuration located on the Top Menu



2. Locate the Web settings and click to expand.



3. Click on the Default Web Folder button



4. This is the current location of your last saved web pages. It is not mandatory to have the same locations of the Physical path on IIS and the web folder on ADISRA SmartView, but it

guarantees that whenever you save the html pages, they will be instantly available on the web server.

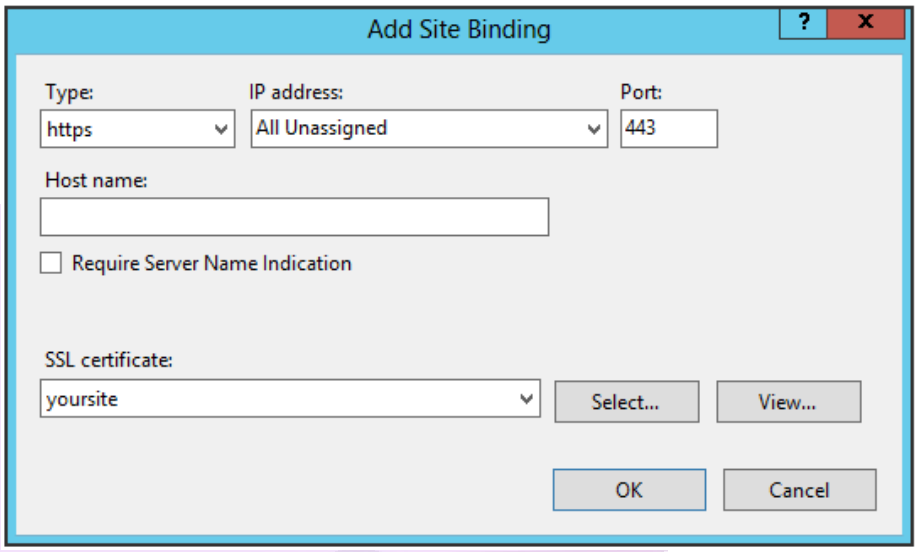
Tip 2 (Configuring the bindings on IIS):

Ports play a pivotal role in the functioning of Internet Information Services (IIS), particularly when it comes to hosting multiple websites on a single server. IIS uses ports to differentiate between various web services running on the same machine. Each port serves as a unique entry point, allowing incoming network requests to be directed to the appropriate web application or site based on its designated port number. By assigning distinct ports to different websites within IIS, administrators can effectively host multiple sites on a single server, each responding to requests on its designated port. This port-based separation enables IIS to manage and route incoming traffic to the correct site, ensuring that each website operates independently and securely.

On our sample application, we have already defined the port 9080 while using https protocol.

```
https://adisra-dashboard.online:9080/Main.html
```

So the next step is to configure our newly created Site "TestHttps" that already has the correct physical path but needs to start listening on port 9080. Since the port 9080 will be used for https, we will also need a valid certificate at that point.



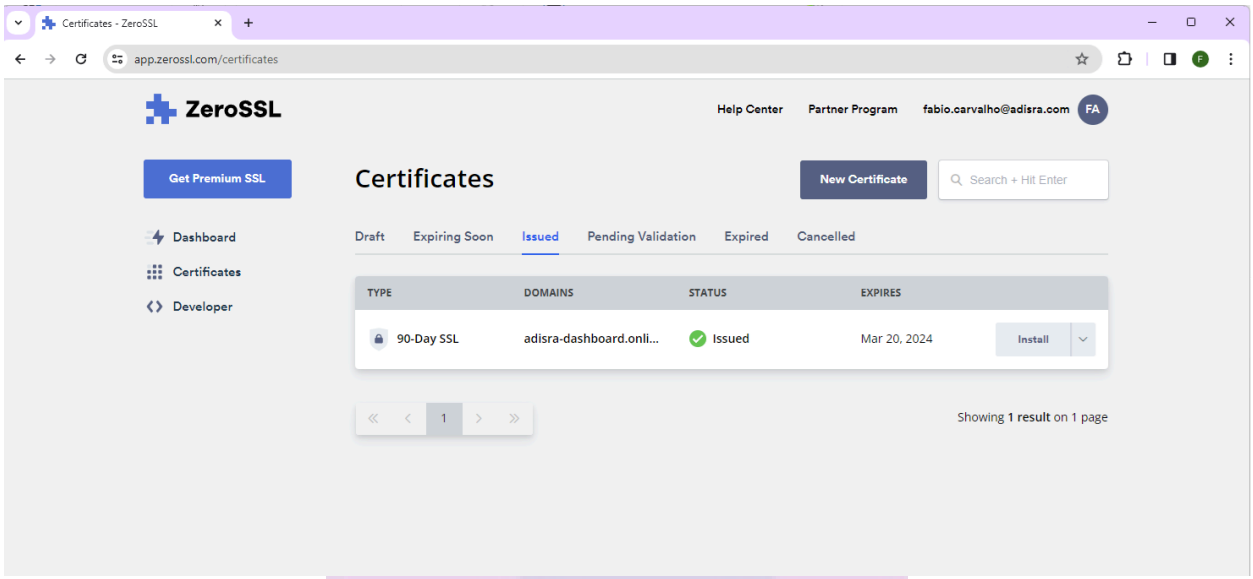
Tip 2.1 - **Create a valid SSL certificate:**

Creating a valid Certificate Authority (CA)-signed certificate is essential for ensuring secure and trusted communication between a user's browser and a website. CA-signed certificates are issued by trusted third-party authorities, providing a level of authentication and encryption that instills confidence in users regarding the legitimacy and security of the website. To generate a CA-valid certificate, a generic process involves generating a Certificate Signing Request (CSR) on the server where the website is hosted. This CSR is then submitted to a reputable CA, which validates the identity of the certificate requester before issuing the signed certificate. This process involves proving ownership of the domain and passing through various verification steps to establish trust. One convenient option for obtaining a CA-signed certificate is ZeroSSL, which offers a user-friendly online platform for generating free CA-signed certificates.

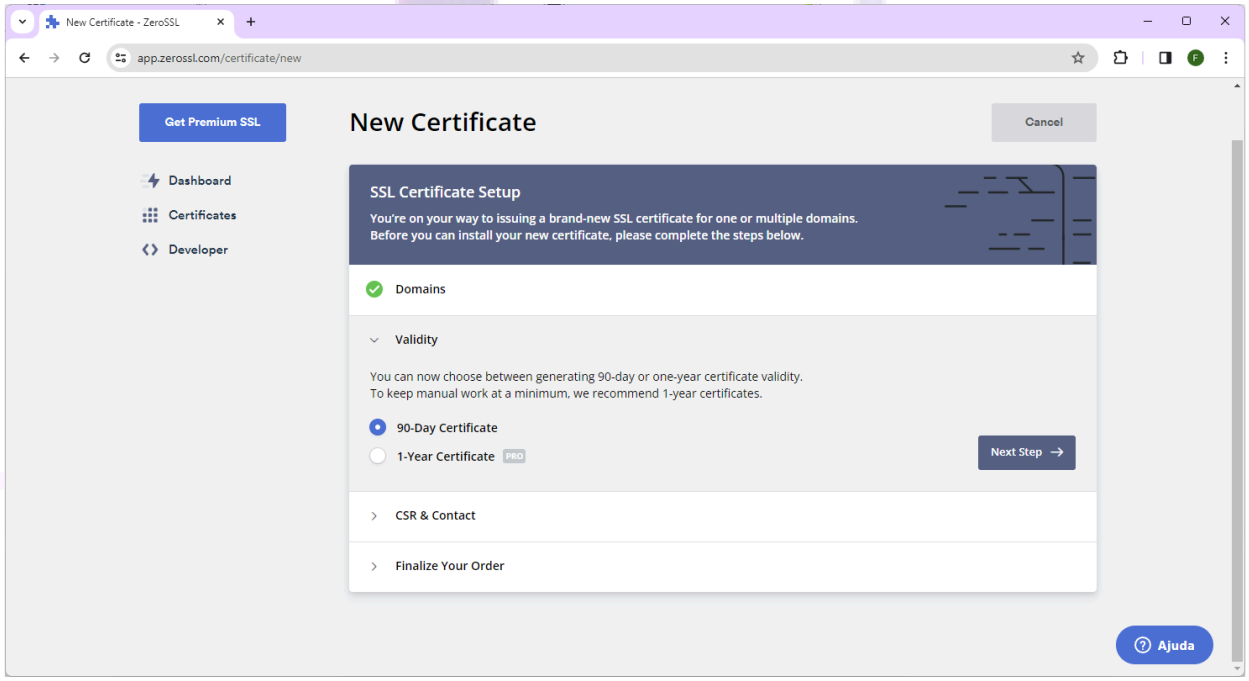
It is up to the user to define how this certificate will be generated, and in case you already have it, you can go directly to (Tip 2.2 - Create a https binding).

In this architecture, we used a certificate generated by https://zerossl.com/. The steps to generate a certificate includes a

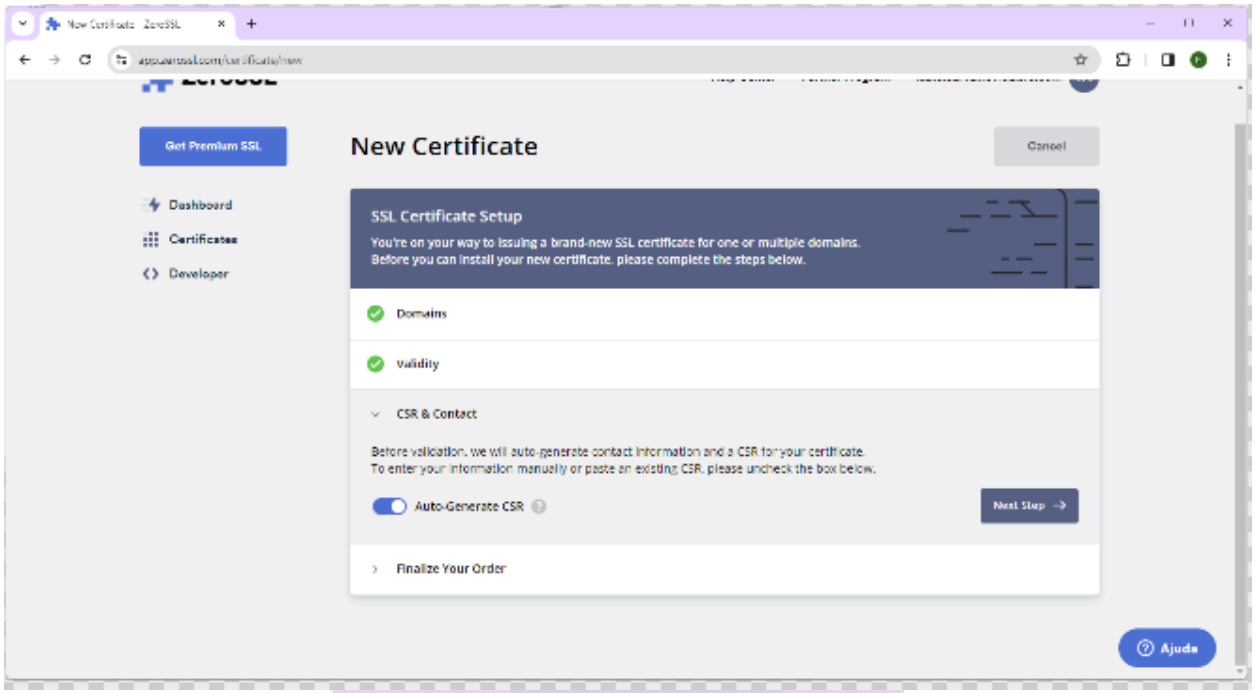
step to prove you are the owner of the domain. Please see the steps below:

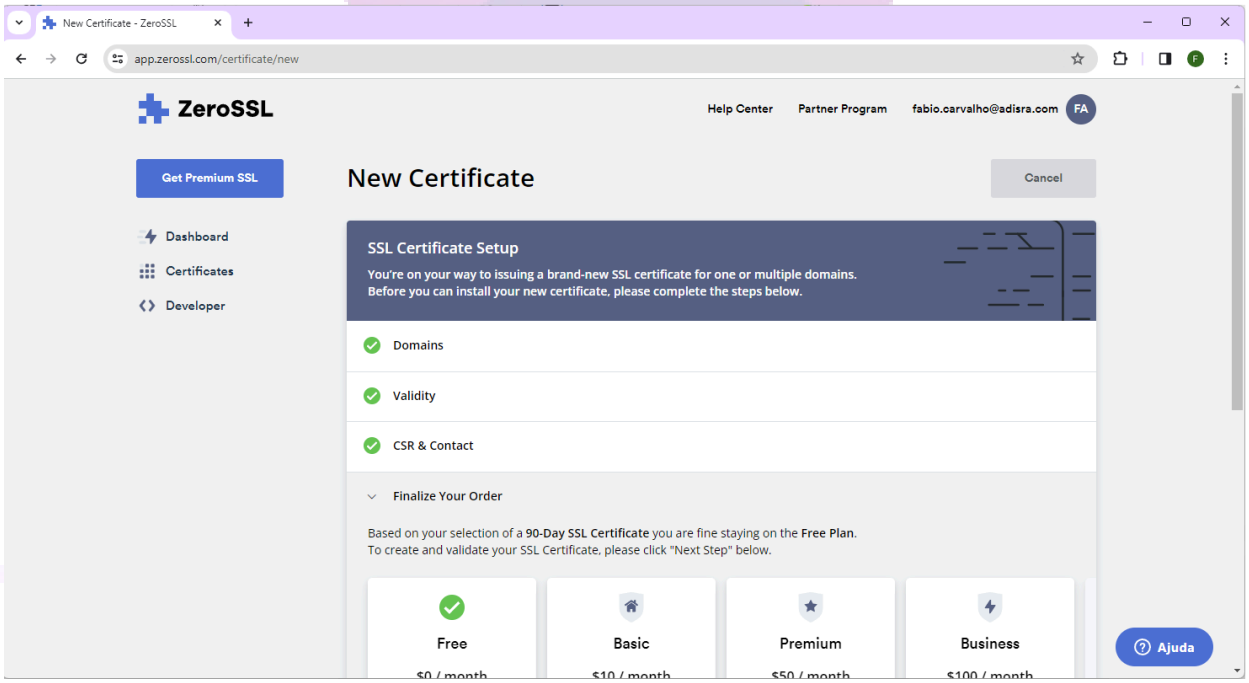


1. Create an account on ZeroSSL
2. Create a “New Certificate”

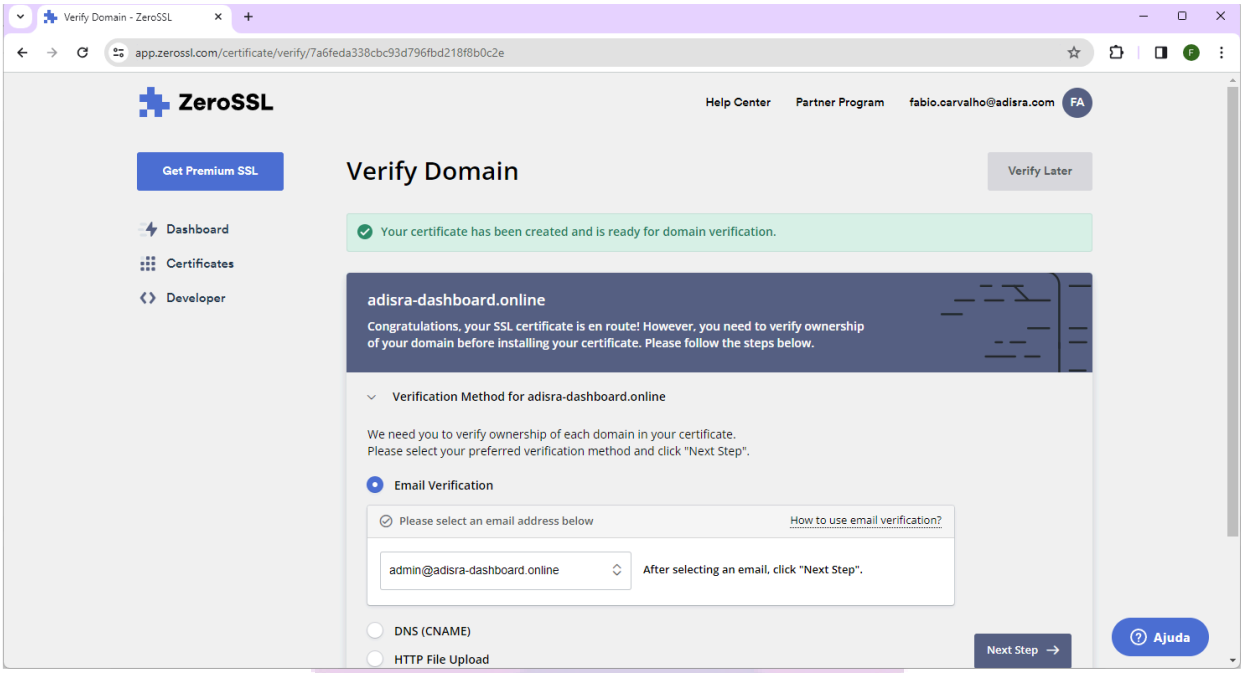


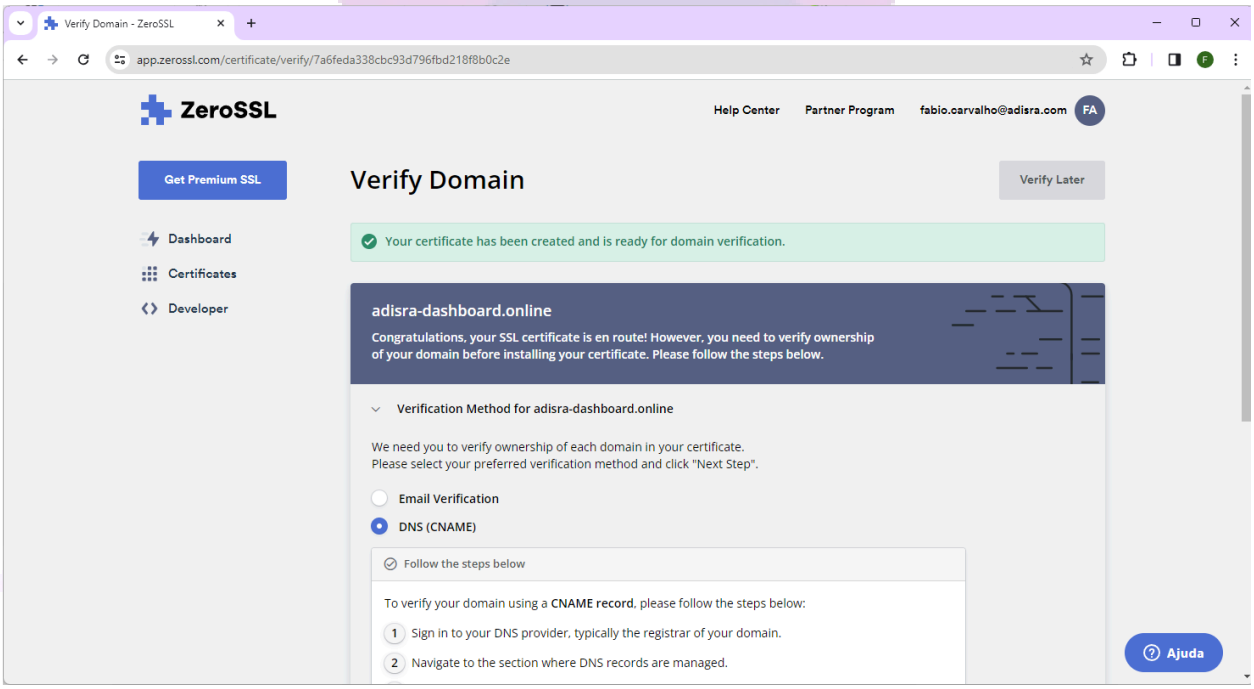
3. Move to next step



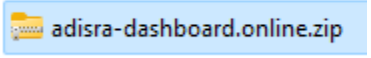


4. Then we will need to verify the Domain as mentioned before. It offers an email verification but I used the CNAME verification.

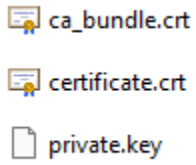




After inserting a CNAME in the domain's DNS configuration, please run the test. If it succeeds and proves you are the owner of the domain, the certificate files will be available to be downloaded. Please download the certificate zip file.



It contains the CRT and KEY files that will be used in the https binding of IIS and also used in ADISRA SmartView Web configuration.



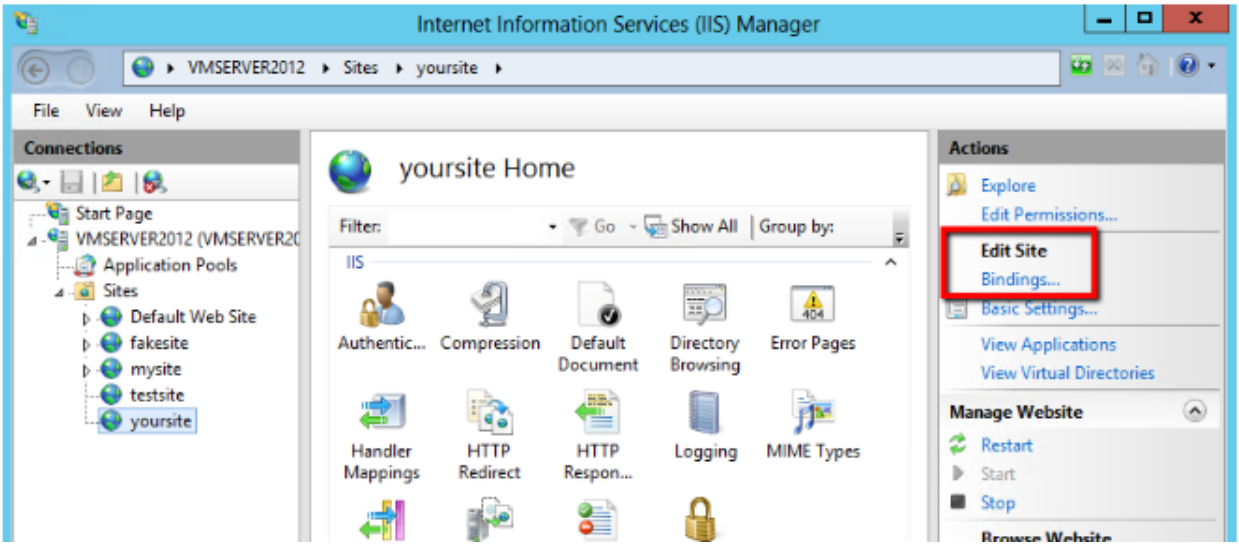
Tip 2.2 - Create https binding on IIS

Since we are using a ZeroSSL certificate, we can use their documentation to install the above certificate on IIS.

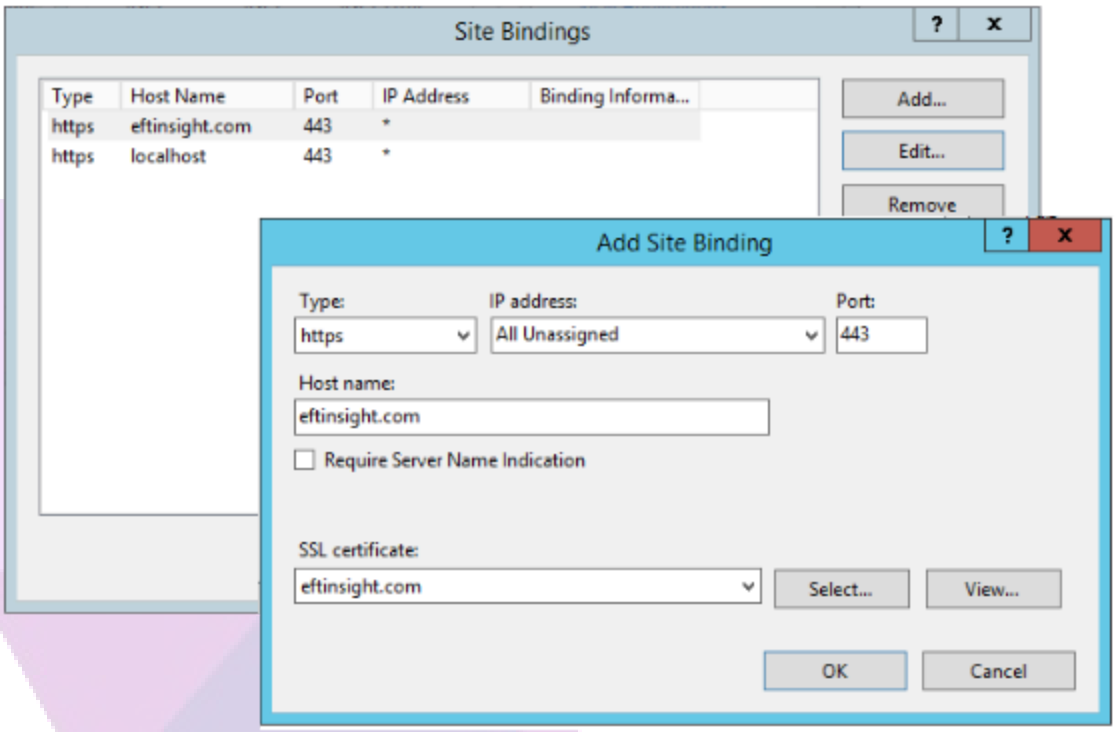
https://help.zerossl.com/hc/en-us/articles/6052154874909-Installing-SSL-Certificate-on-IIS

After it is properly installed, it will be available in the Bindings configuration.

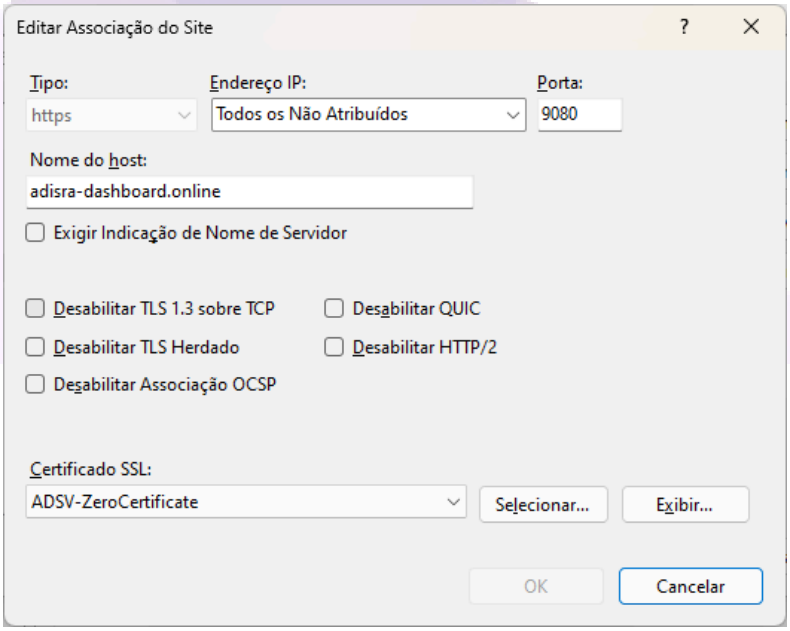
Let's go back to the bindings. You will find the "Bindings" of your Site on the right side (image below).



The dialog below will be displayed when you click the "Bindings" above. You will click "Add" and select Type "https".



In this binding configuration, we will set the port 9080, the host name “adisra-dashboard.online” and then select the newly installed zerossl certificate.



In case you want different bindings, you can add them at any point and create new routes. After updating the Site configuration on IIS, please restart it (on the right side under *Manage Website*).

At this point (chapters above) we have guaranteed a domain and a certificate, we have confirmed the ISP is not blocking any communication to our web server and we have configured the router to forward requests to our web server in the LAN and we have configured the IIS Web Server to host pages on the selected physical path and to listen on port 9080 for https secure connection using the certificate generated by zeroSSL.

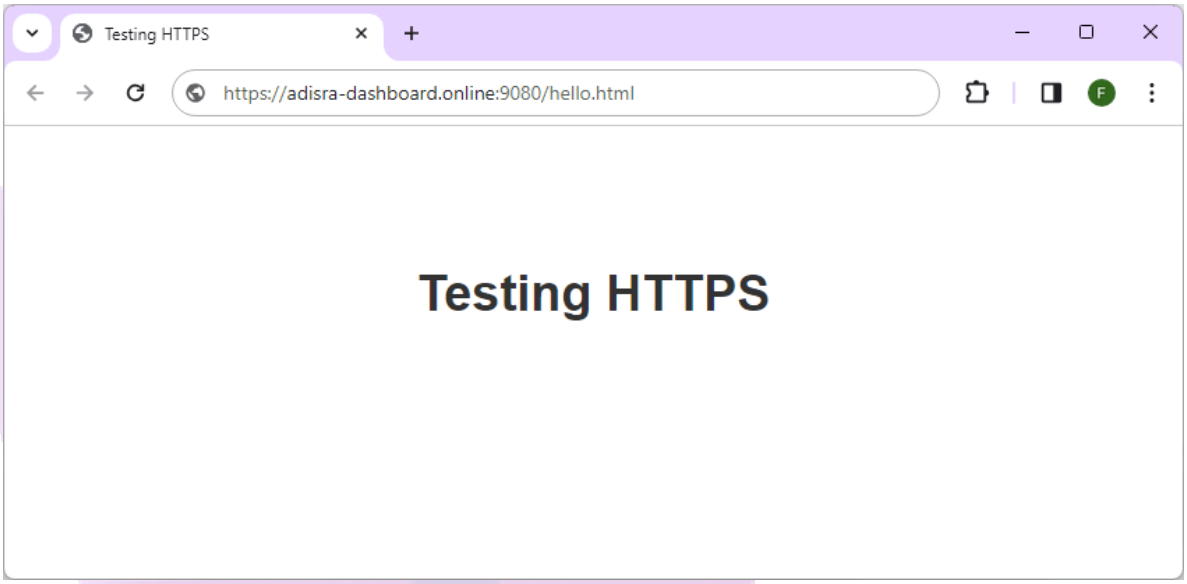
At this moment I suggest testing a simple html file before moving to test ADISRA SmartView.

We can add a "hello.html" to the physical path of our TestHttps website.

```
<!DOCTYPE html>
<html>
<head>
   <title>Testing HTTPS</title>
   <style>
      /* CSS to style the label */
      body {
         font-family: Arial, sans-serif;
         text-align: center;
         margin-top: 100px;
      }
      h1 {
         font-size: 36px;
         color: #333;
      }
   </style>
</head>
<body>
   <h1>Testing HTTPS</h1>
</body>
</html>
```

If everything is configured properly, we will be able to open the hello.html using the following url:

https://adisra-dashboard.online:9080/hello.html



If it is loaded correctly, you can move to the next sub-chapter. If not, please review the architecture and make sure all elements in it are correctly configured.

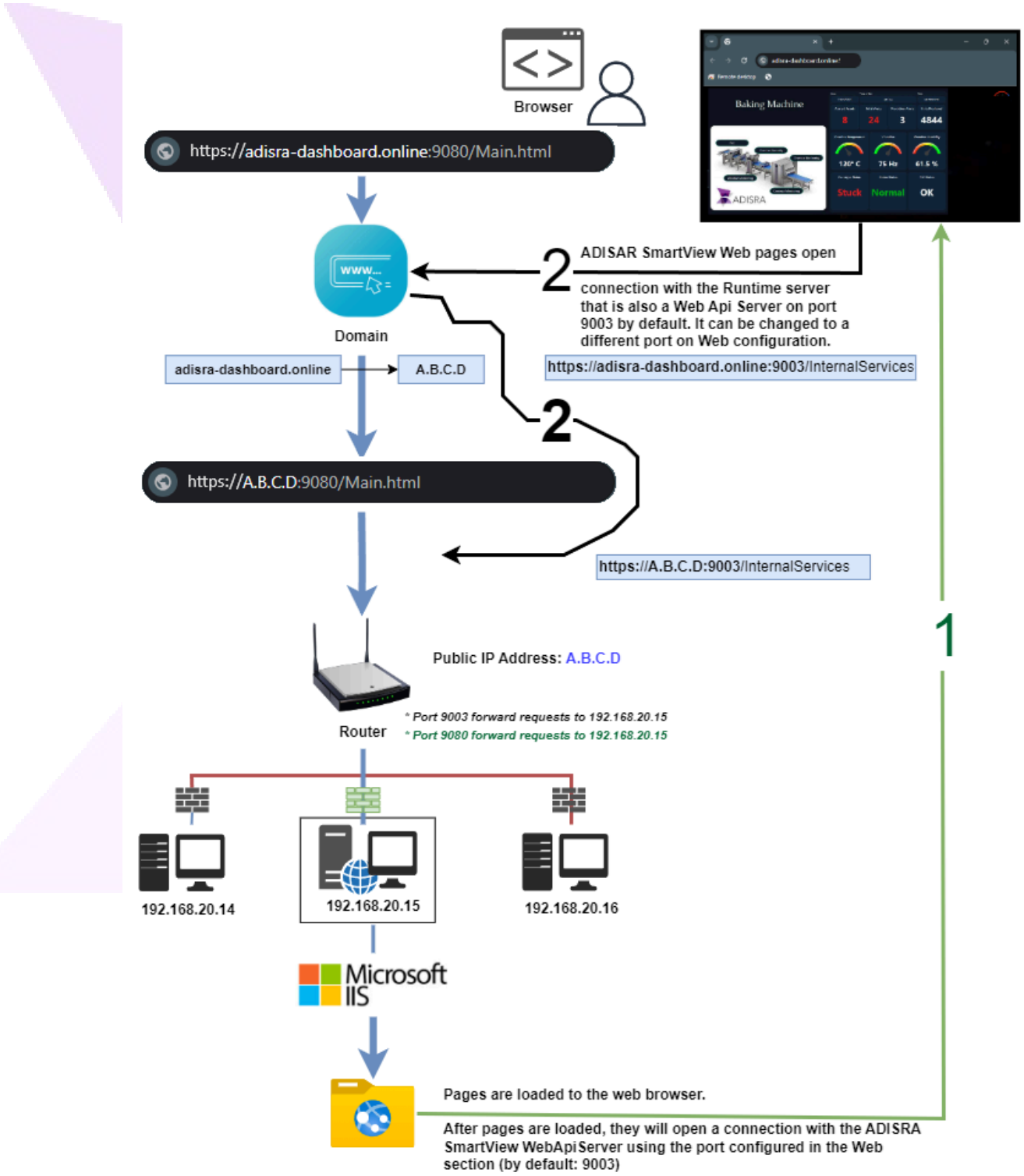
3.5. ADISRA SmartView

Summary:

Considering all above configuration is working properly, we can start configuring ADISRA SmartView.

ADISRA SmartView Architecture includes the Runtime process which is responsible for different modules such as Drivers, Recipes, Database communication, etc, but it also includes a Web Api Server. This server enables communication between the web clients (web pages generated by ADISRA SmartView) and the runtime.

The image below illustrates what happens when a user searches for the url "https://adisra-dashboard.online:9080/Main.html"

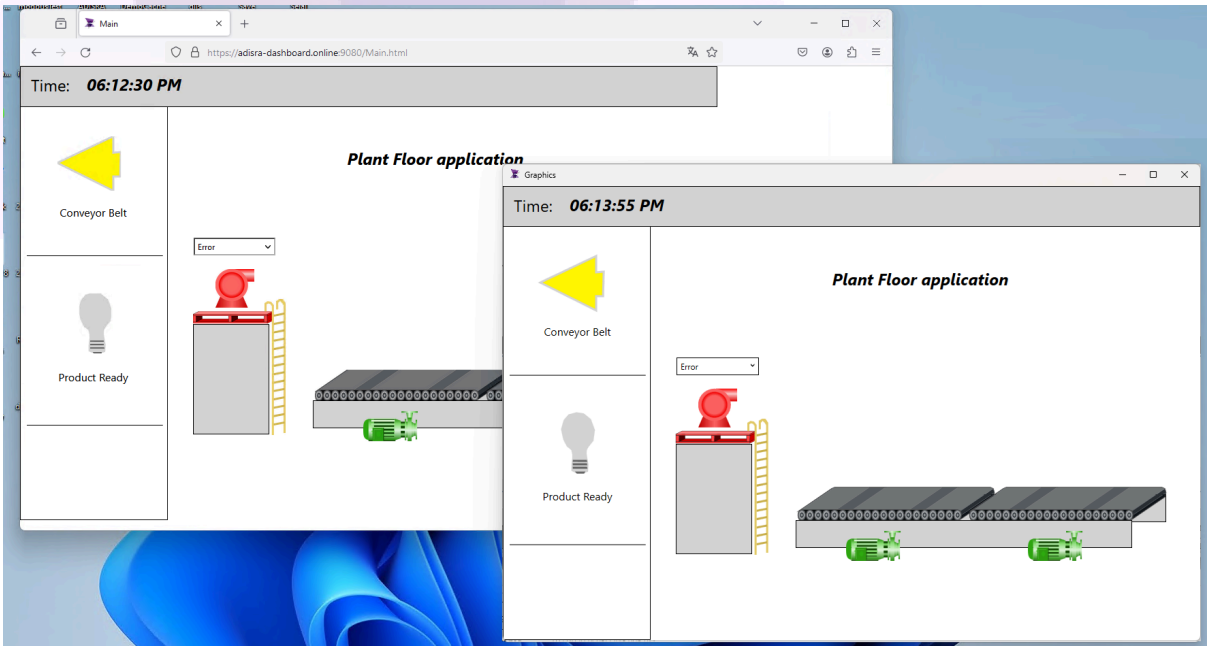


1. User types "https://adisra-dashboard.online:9080/Main.html" on the web browser;
2. Since the domain "adisra-dashboard.online" is set to forward traffic to my public ip address "A.B.C.D", the request will reach my Router.
3. The router will identify port 9080 and forward to my web server on 192.168.20.15 ip address.
4. The IIS listening on port 9080 will respond with the Main.html page
5. The Main.html page will start to be displayed on the client side.
6. At this moment, Main.html will need to communicate to ADISRA SmartView runtime server on port 9003 to synchronize data. (For example https://adisra-dashboard.online:9003/security/CheckServer)
7. It will follow the same route as mentioned above, but this time it won't be IIS listening on port 9003, but it will be ADISRA SmartView runtime process.

Step by Step:

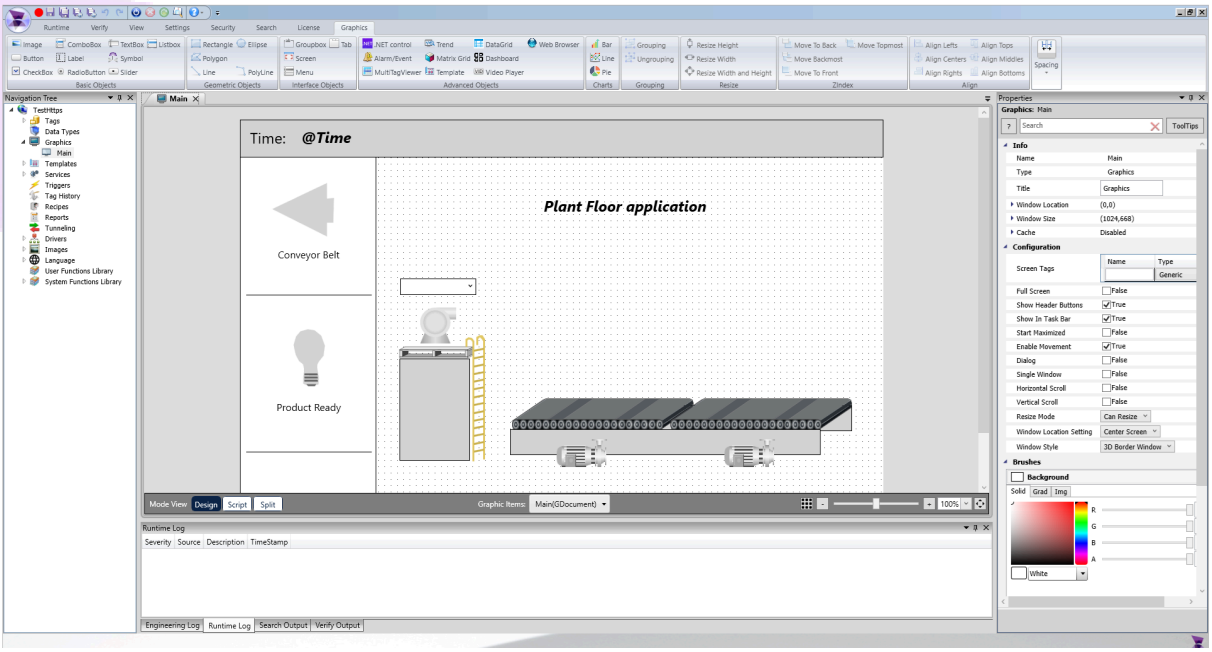
In this sample application, we will have one single screen created and we will be able to access it using the Desktop Viewer sessions and also Web Viewer Sessions through the url below:

https://adisra-dashboard.online:9080/Main.html



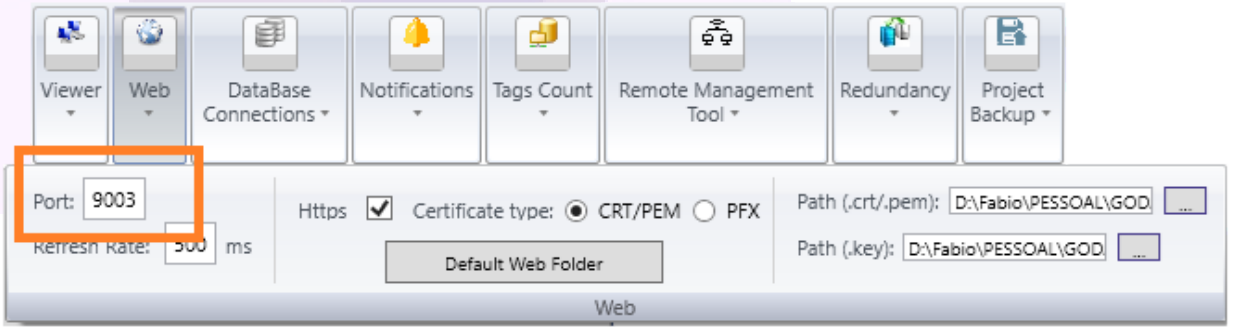
3.5.1. Creating the application

This is not the focus for this document, so we won't show the details of the application. The image below shows the application Main screen.



3.5.2. Checking the Web Api Server port number

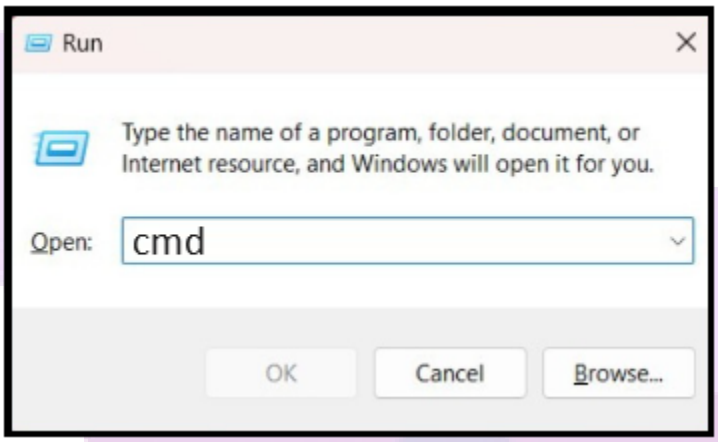
Open the Settings tab and expand the Web settings.



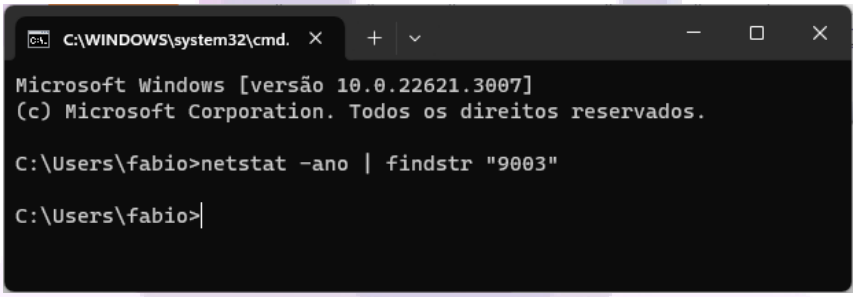
The port 9003 will be used to establish connection to the Web Api Server.
You can test if everything is correct with your port number. First, without running the application, try to check if port 9003 is in use.

Testing if port 9003 is available:

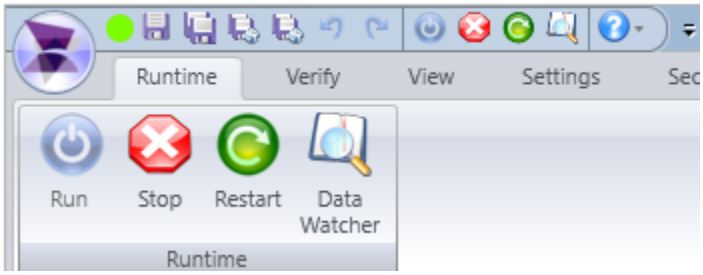
1. Open CMD.exe



2. Use netstat to check port 9003. The following command filters port 9003.

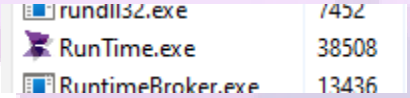


3. In the test above, ADISRA SmartView application was not running. Now let's run it.
4. Click "Run"



5. Check again the netstat command.

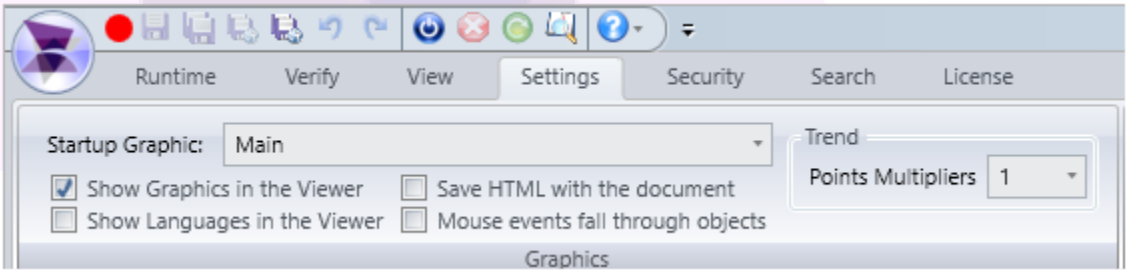
6. Now it responds that the port 9003 is being used by process id 38508, which matches the Runtime.exe process from ADISRA SmartView.



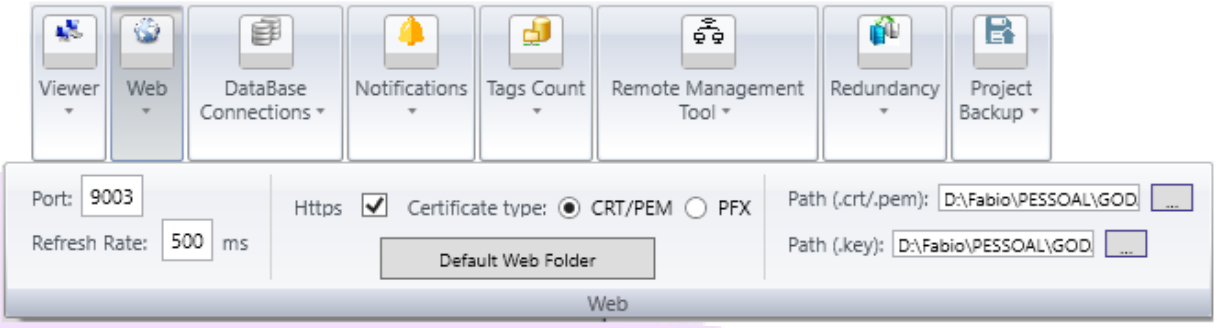
7. In case the web port is not in Listening state when performing the test with the application running, please check the runtime logs for further details. It might be a misconfiguration on the Web Settings.

3.5.3.Enabling HTTPS

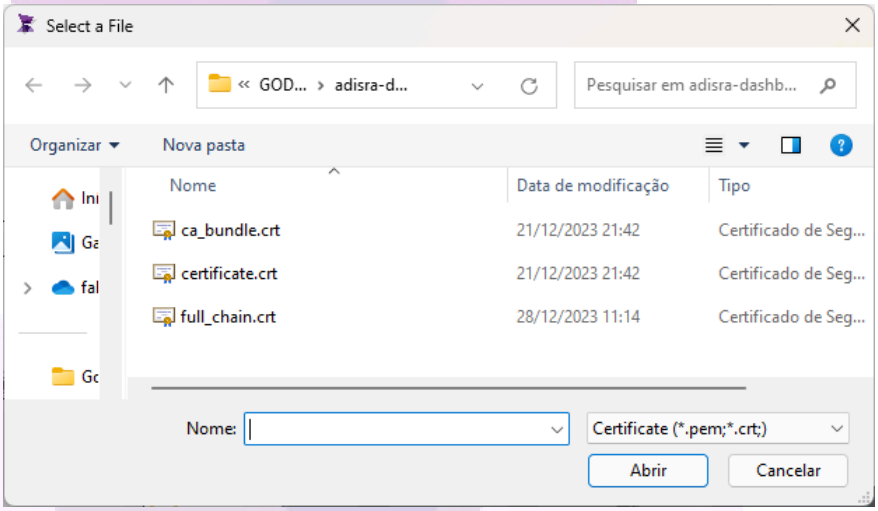
1. Open Settings tab



2. Expand Web Settings

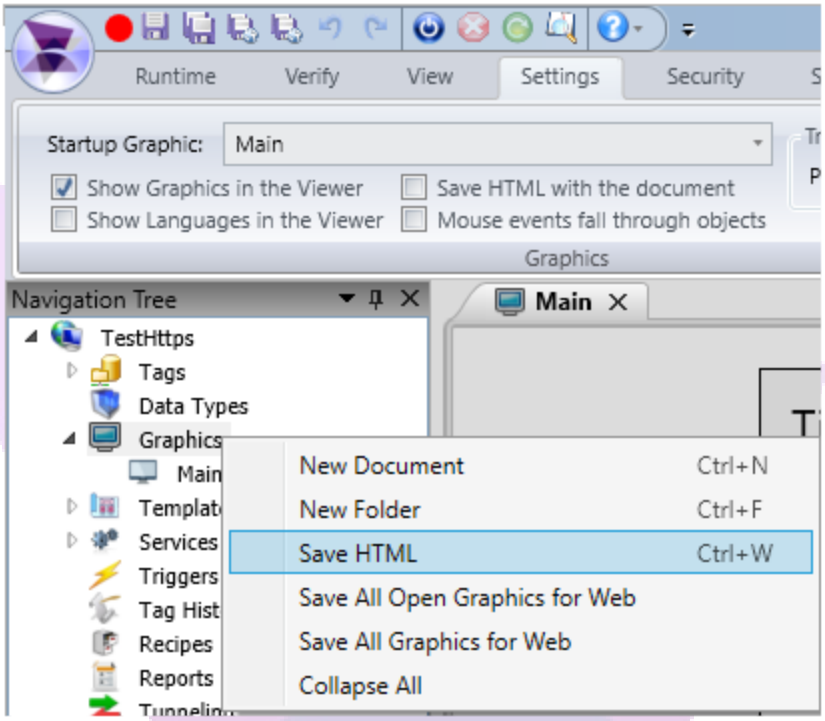


3. Enable Https and select certificate type. It supports CRT/PEM/PFX certificates. After selecting the type, on the right side will be able to select the files. In this application we have selected the certificate.crt and the private.key provided by ZeroSSL.

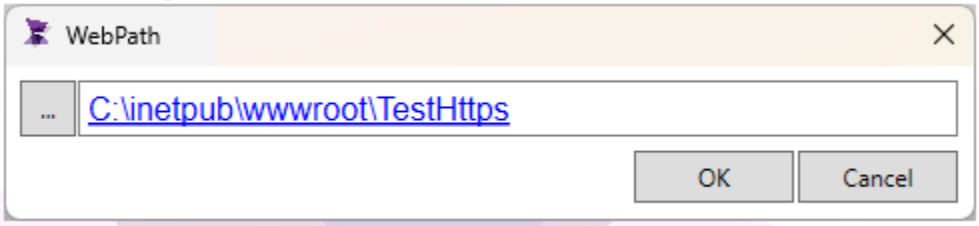


3.5.4.Saving the application to the Web

Please right-click on the Graphics node and select “Save HTML”. It will start a background task to save everything to the web. Please check the progress bar on the bottom of ADISRA SmartView to check when the save process is finished.



It will be saved to the default folder and ready to be used.



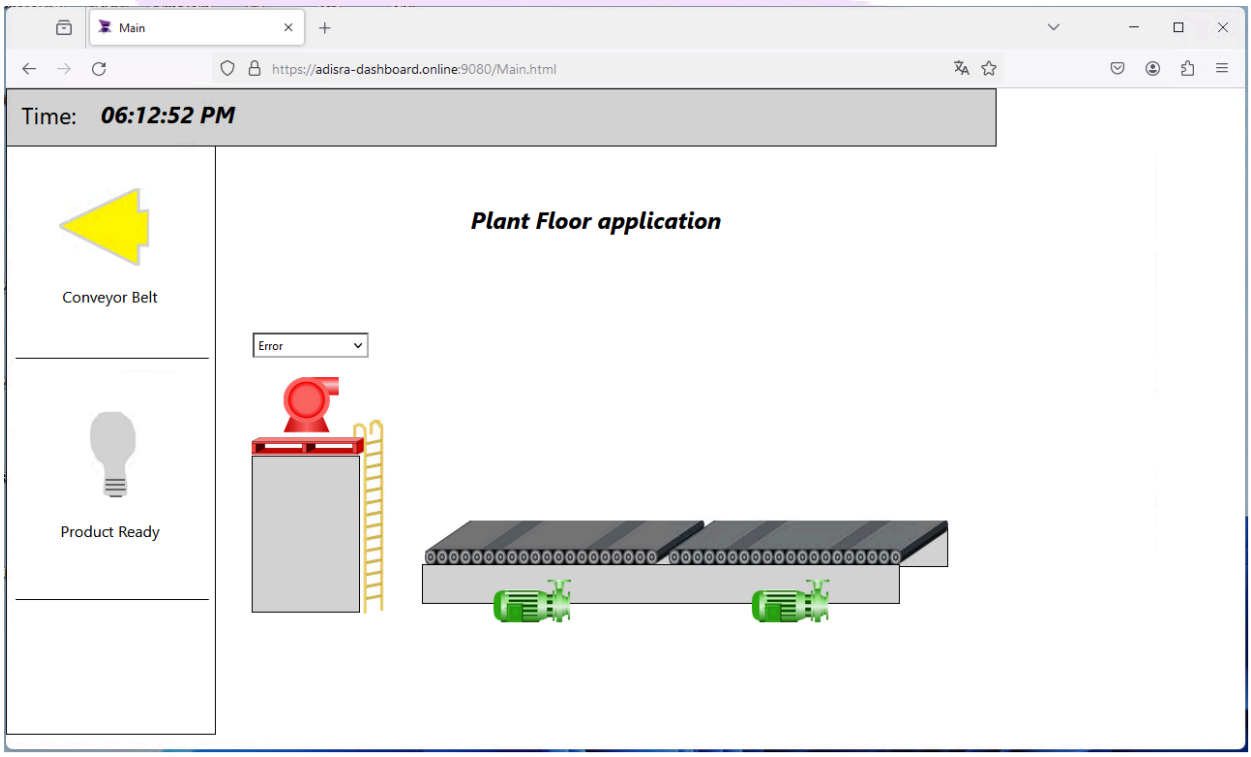
3.5.5.Running the application

At this point, we can start the application and test it. If there is no error to start the Web Api Server, we should be able to load the pages using the correct url:

```
https://adisra-dashboard.online:9080/Main.html
```

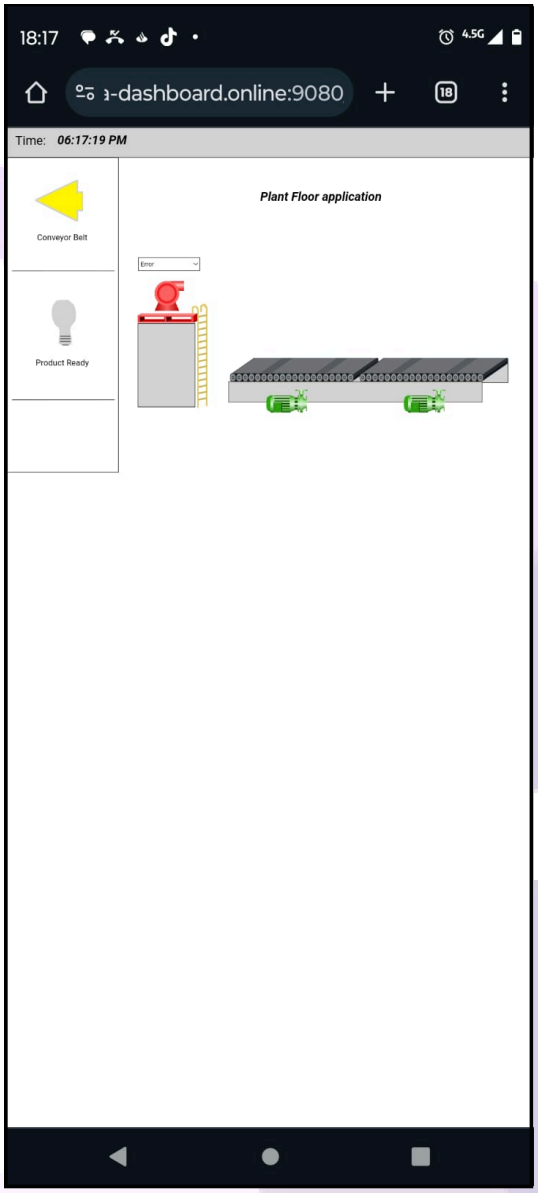
3.5.6.Opening on web browser

When the user searches for "https://adisra-dashboard.online:9080/Main.html", it is forwarded to our router's public ip address, then the router forwards the request to the web server (192.168.20.15). The IIS website TestHttps will be listening on port 9080 and will respond with the Main.html page. When the page is loaded on the client side, its internal features will connect to the ADISRA SmartView runtime web api server on port 9003



3.5.7.Opening on a mobile phone

This is the same process as above. Just open the link "https://adisra-dashboard.online:9080/Main.html", and the page must be loaded as in the image below.

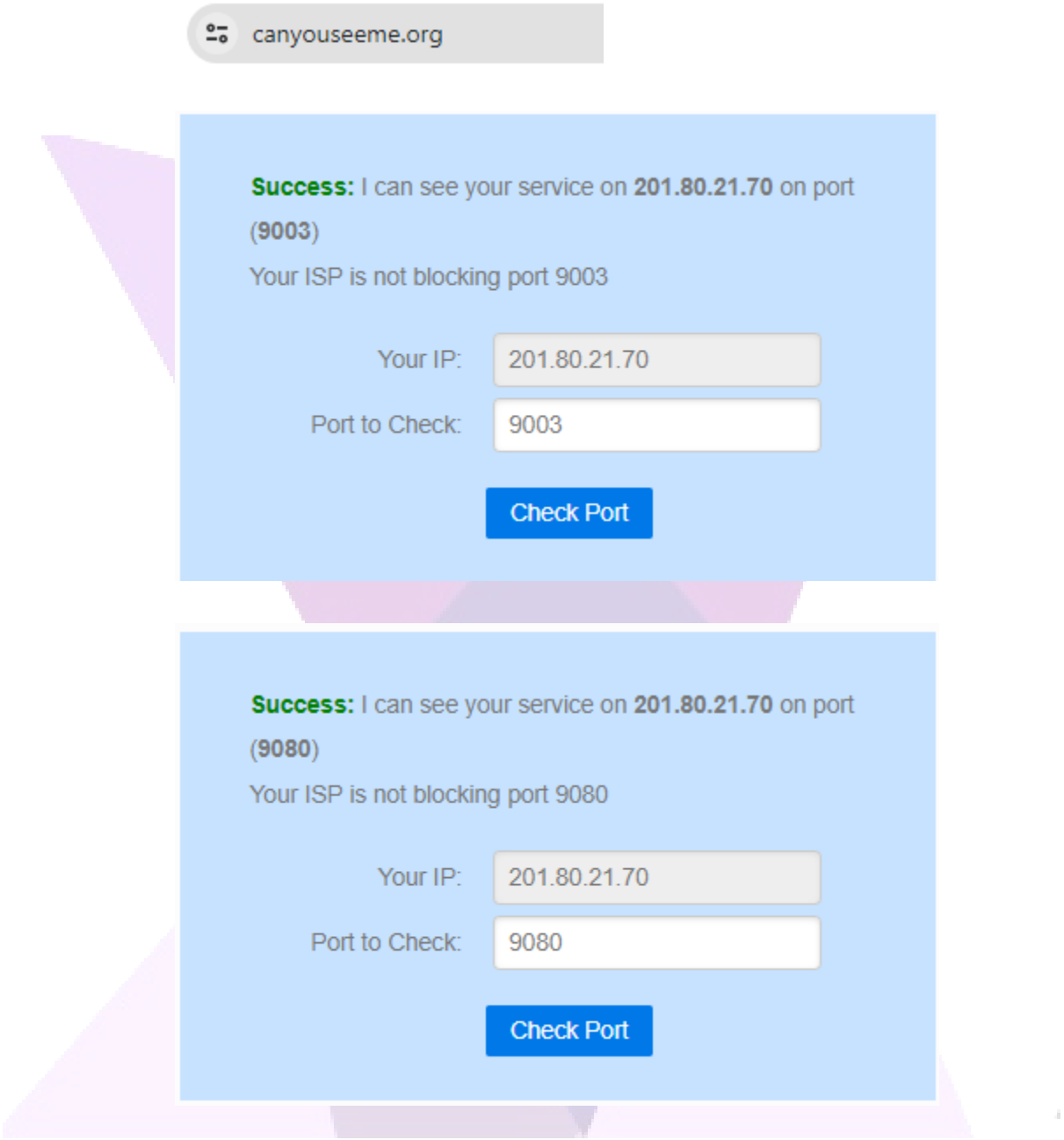


4. Additional Tips

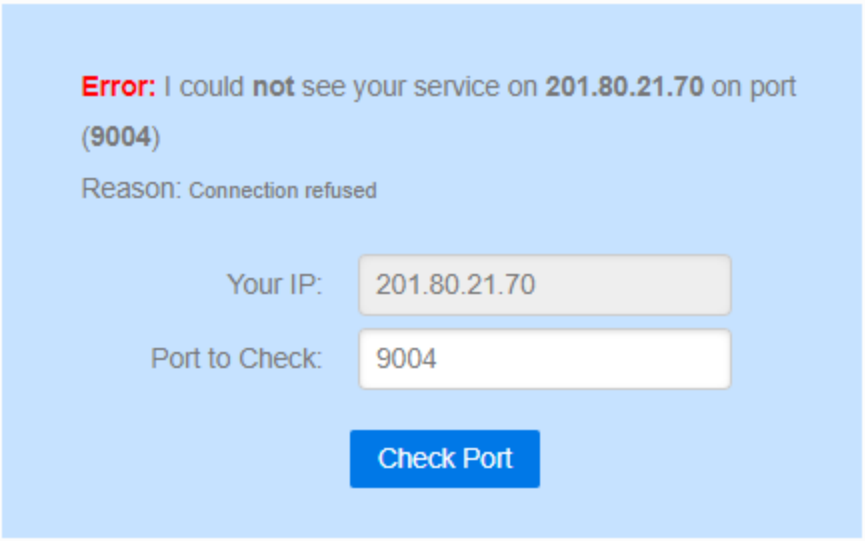
This chapter contains some useful tools that will help you during the test phase. Some of them were already mentioned in the documentation above.

4.1. Test ports using online tool

You can use https://canyouseeme.org/ to test ports 9003 and 9080.

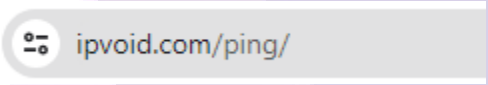


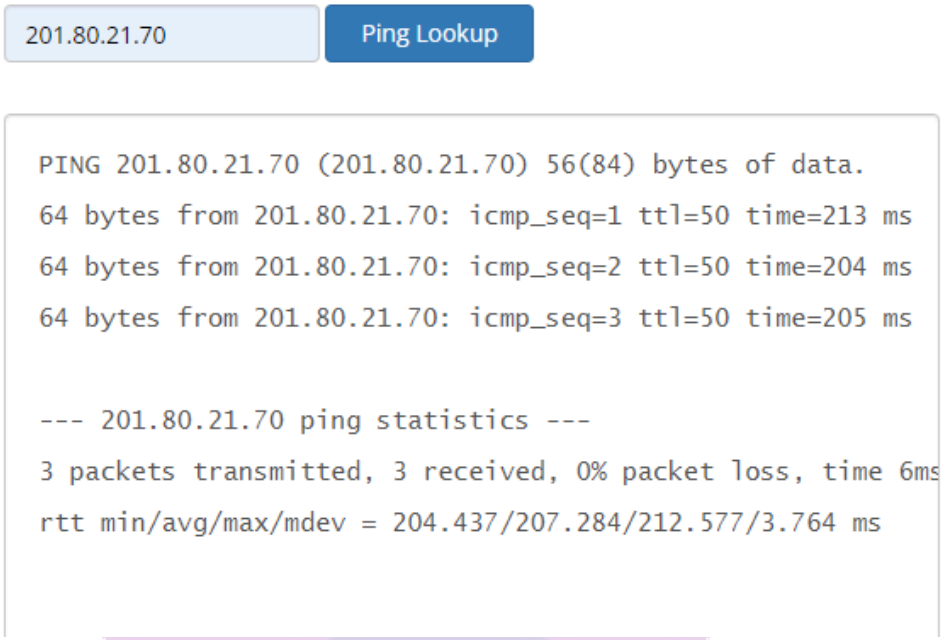
In case the port is unavailable, it will return an error.



4.2. Ping public ip address

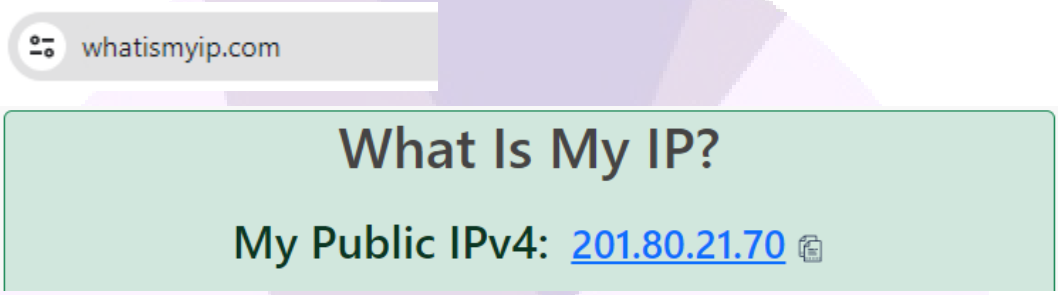
You can use https://www.ipvoid.com/ping/ to ping your public ip address or your domain. This is just an alternative to the CMD "ping" command. And keep in mind as mentioned before that some environments don't support ping responses.





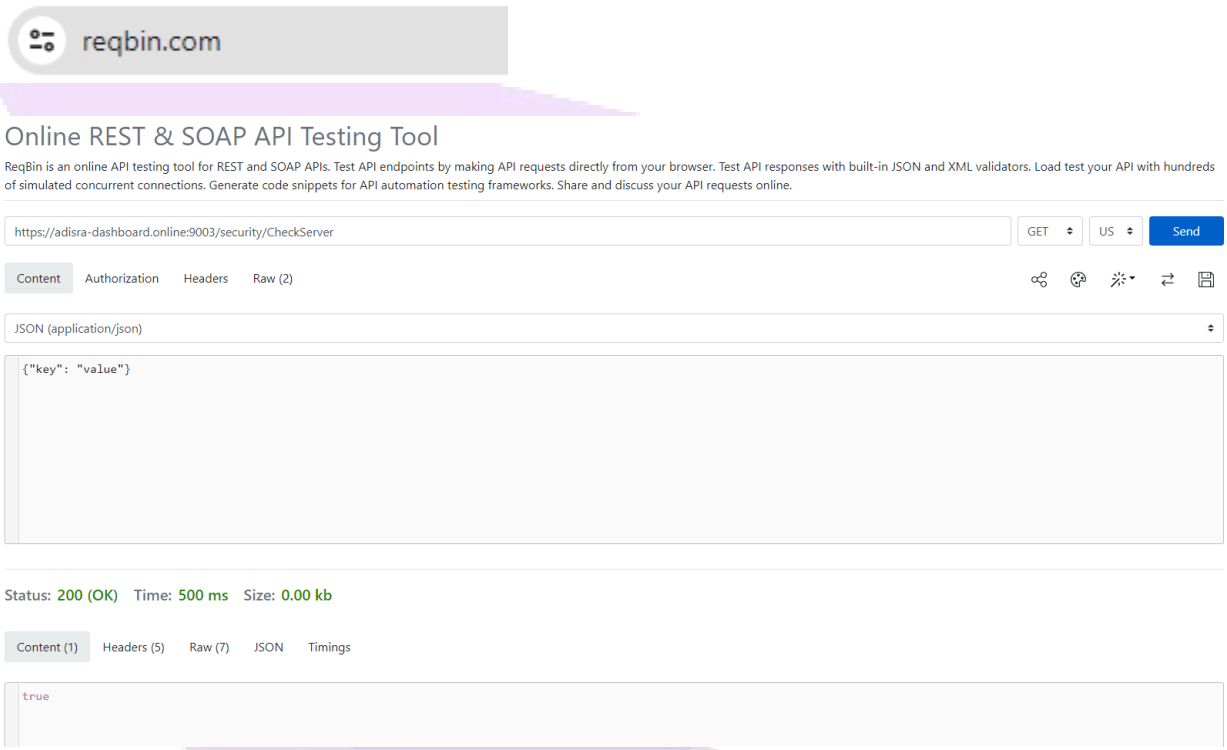
4.3. Checking public ip address

You can use https://www.whatismyip.com/ top find out your public ip address.



4.4. Testing the “CheckServer” service on port 9003

You can use https://reqbin.com/ to test the CheckServer service. It will only succeed if port 9003 is up and running.



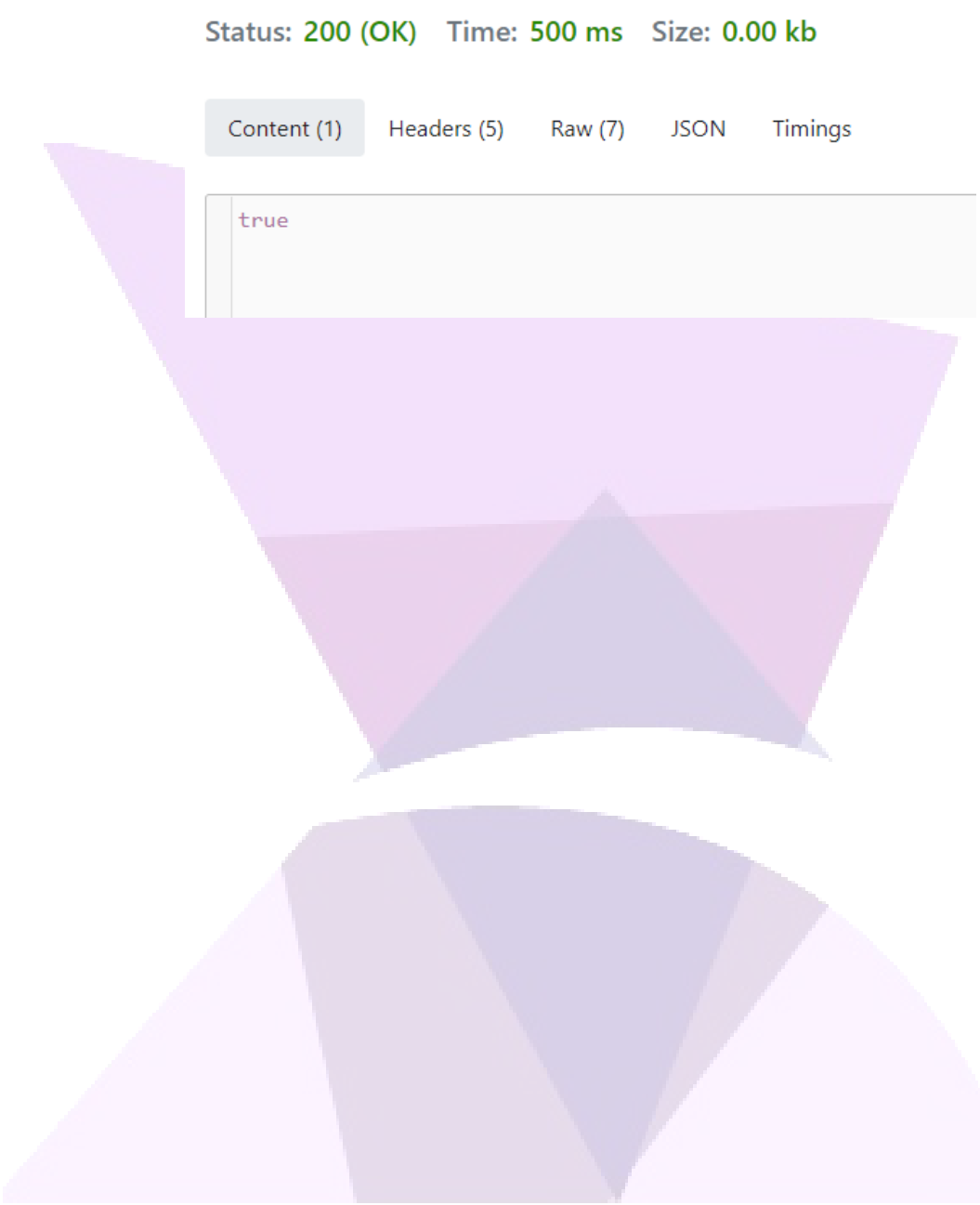
Url (replace with your domain and port):

https://adisra-dashboard.online:9003/security/CheckServer

Type:

GET

Response:


Status: 200 (OK) Time: 500 ms Size: 0.00 kb
Content (1)
Headers (5)
Raw (7)
JSON
Timings
true