So in Pega we know that it has a certificate which is imported into the truststore

It has

* Authentication services that import identity metadata exposed over HTTP URL
* Connectors that access external REST API over HTTPS

And we also have the certificates which belong to **Data-Admin-Security-Certificate**

We have 2 activities to either make the activities or deactivate the certificate

The 2 activities are

1. pxChangeCertificateStatus
2. pxDeleteCertificate

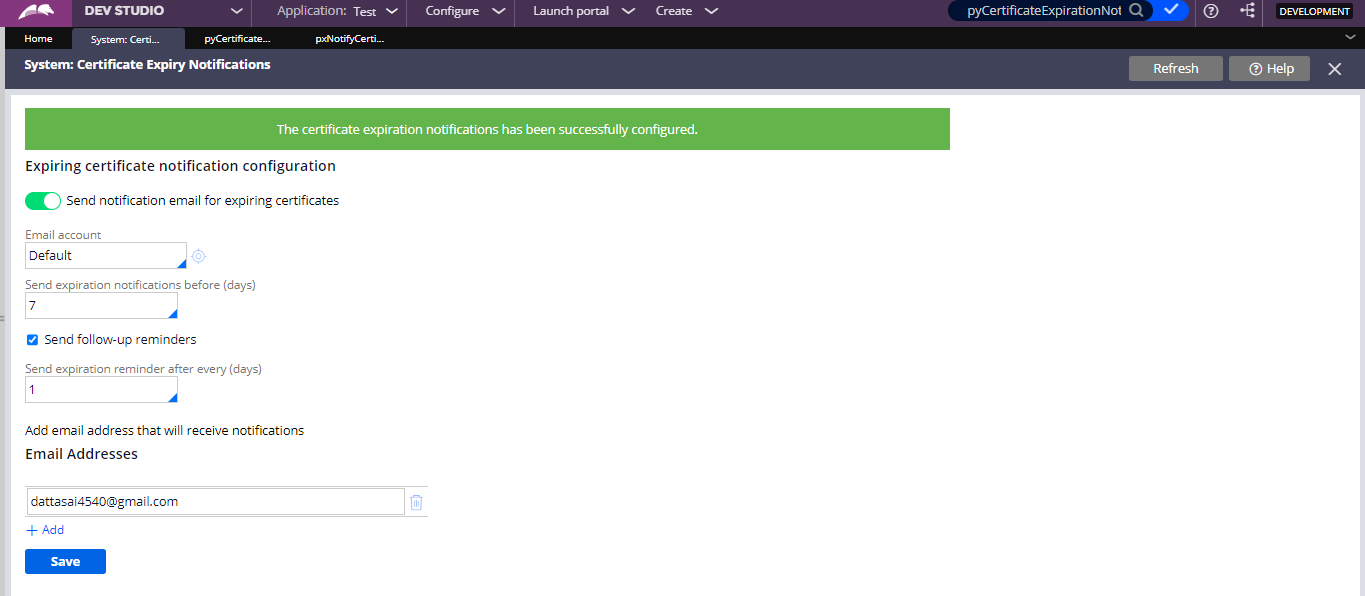
if the certificate is not present in the Pega Platform Truststore. If the certificate is not present then external host is not authenticated and an exceotion is thrown.

Now in order to get hold of these all certificates and their expirations we have Expiring Certificate Notifications if any changes happen in the certificate.

In Pega Dev Studio

Navigate to **Configure>System>Expiring Certificate Notifications**

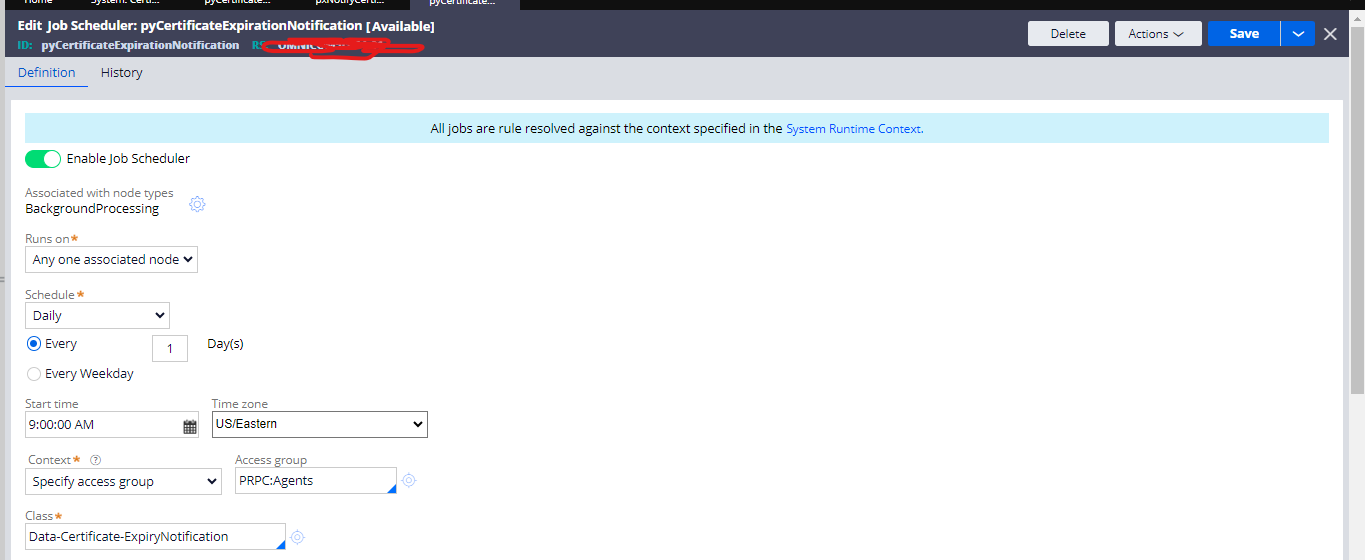
Then



Now here you can configure an email to which it has to send the notifications and also it should be sending the mail to the email account and also after how many expirations days it should send the follow up notifications also will be sent to same email..

We have one job scheduler which we need to enable and by default it is disabled..

Job Scheduler- **pyCertificateExpirationNotification**



Now saved it to the ruleset and it has one activity and which includes Java to run and send the notifications.

A screenshot of a computer

Description automatically generated

Now lets see the java step what it is trying to do

try {

pega.getCertificateUtils().sendCertificateExpirationNotification(tools);

} catch (PRRuntimeException ex) {

oLog.error(ex);

}

Explanation-

Here's what this code is doing:

It's using a try block to encapsulate a block of code that might throw exceptions.

Inside the try block, there's a method call pega.getCertificateUtils().sendCertificateExpirationNotification(tools);. This seems to be calling a method to send a certificate expiration notification using some certificate utilities provided by the Pega platform.

If an exception of type PRRuntimeException is thrown during the execution of the try block, it will be caught by the corresponding catch block.

Inside the catch block, the caught exception (ex) is logged as an error using oLog.error(ex);. It seems that oLog is an object that provides logging functionality, and it's logging the error message associated with the caught exception.

This structure is a common pattern used to handle exceptions in Java. It ensures that the application can gracefully handle errors and log relevant information when exceptions occur during the execution of the try block.

So this is configuration and explanation of how the Expiring Certifications work

Happy learning!!!