**Create PDF by using Class Mediator**

# Objective:

The objective of this application is to create a PDF file with the information present in the message context (In sequence payload) by class mediator using java code.

# Set up the workspace:

The following software needs to be installed.

* [WSO2 Integration Studio](https://ei.docs.wso2.com/en/latest/micro-integrator/develop/installing-WSO2-Integration-Studio/)
* Download this Java library which deals with PDFs [itextpdf-5.5.13.2.jar](https://bit.ly/3v5ZumI)

# Develop the integration artifacts:

## Create an Integration Project

An Integration project is a maven multi module project, which will contain all the required modules for the integration solution.

1. Open **WSO2 Integration Studio**.
2. Click **New Integration Project** in the **Getting Started** tab as shown below.

Calendar

Description automatically generated

This will open the **New Integration Project** dialog box.

Graphical user interface, text, application

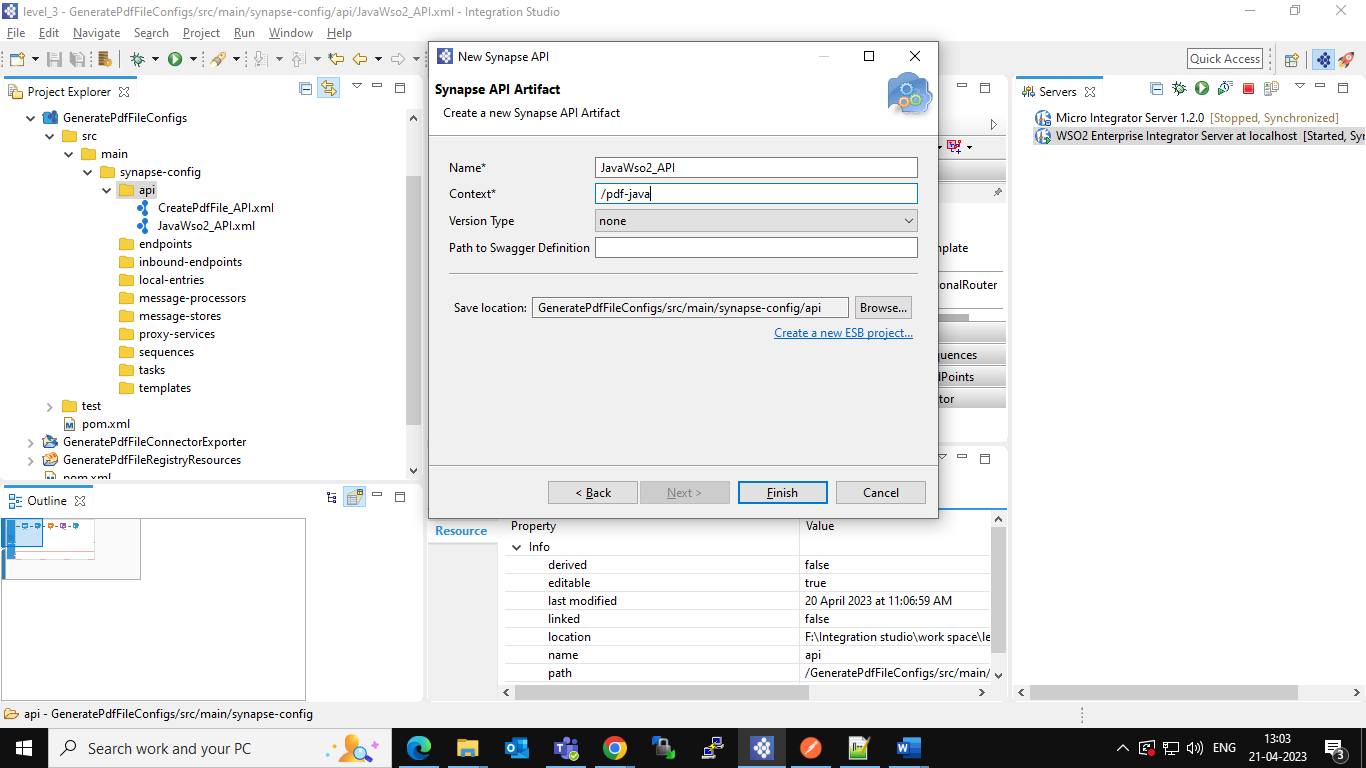
Description automatically generated

1. Enter **GeneratePdfFile** as the project name and select the following check boxes to create the required modules.
   * **Create ESB Configs**
   * **Create Composite Exporter**
2. Click **Finish**.

You will now see the projects listed in the **Project Explorer**.

#### Create a REST API.

1. In the Project Explorer, right-click **GeneratePdfFileConfigs** in the project explorer and click **New -> REST API**.
2. Ensure **Create A New API Artifact** is selected and click **Next**.
3. Enter the details given below to create a new REST API.



Make sure that click on POST method in **API Resource**.

Graphical user interface, text, application

Description automatically generated

**Source code of JavaWso2\_API:**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<api context=*"/pdf-java"* name=*"JavaWso2\_API"* xmlns=*"http://ws.apache.org/ns/synapse"*>

<resource methods=*"POST"*>

<inSequence>

<property expression=*"$body/\*"* name=*"message"* scope=*"default"* type=*"STRING"*/>

<log level=*"custom"*>

<property expression=*"$ctx:message"* name=*"message log"*/>

</log>

<class name=*"com.knot.ConvertTextToPdf"*/>

<payloadFactory media-type=*"text"*>

<format>$1</format>

<args>

<arg evaluator=*"xml"* expression=*"$ctx:out"*/>

</args>

</payloadFactory>

<respond/>

</inSequence>

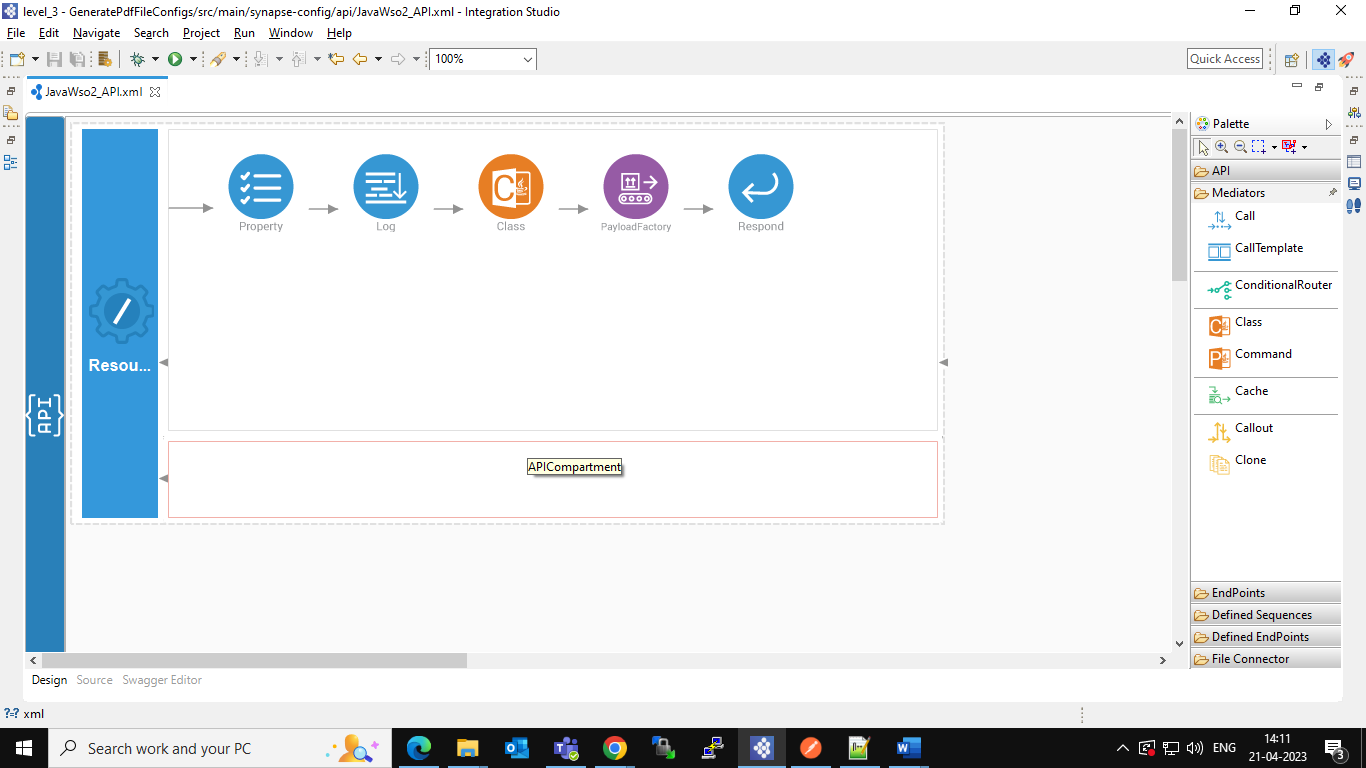
<outSequence/>

<faultSequence/>

</resource>

</api>

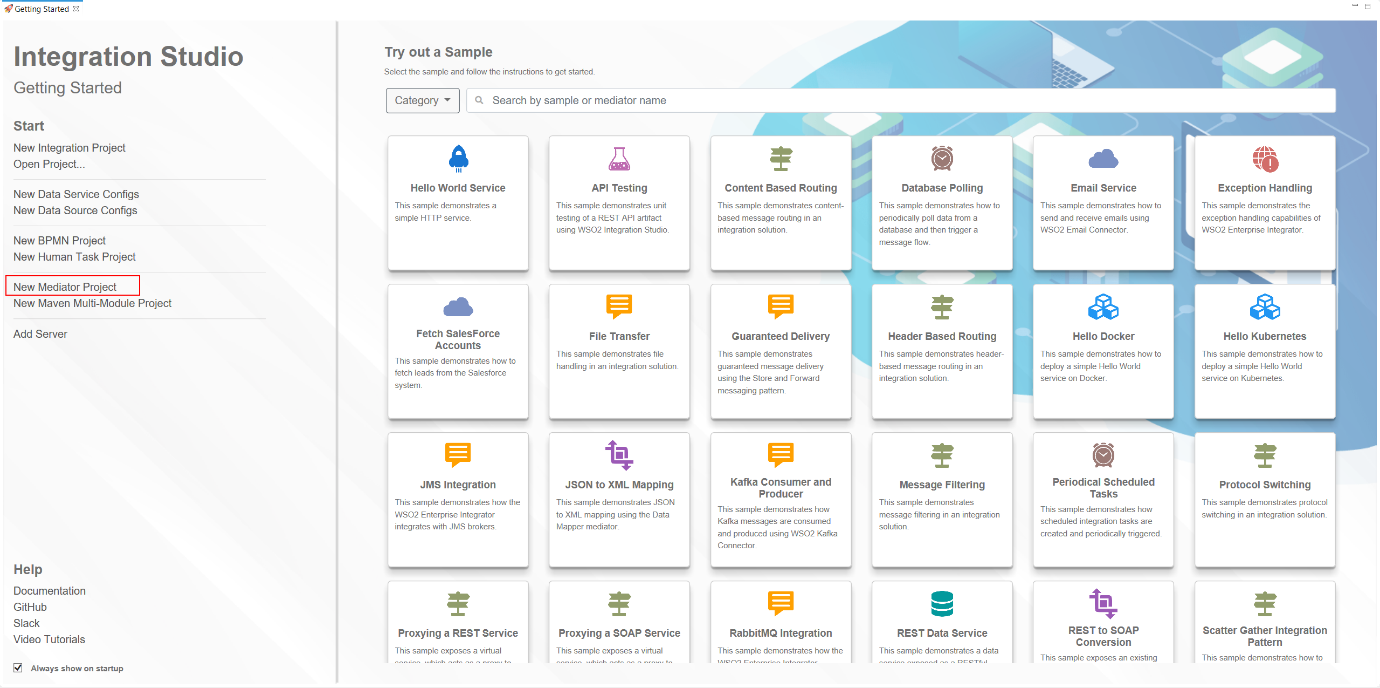
Mediation flow Design looks like below image:



## Creating a Mediator Project

Create this project directory to start creating custom mediator artifacts. You can use these customer mediators when you define the mediation flow in your ESB config project.

1. Open **WSO2 Integration Studio** and click **Miscellaneous → Create Mediator Project** in the **Getting Started** view as shown below.

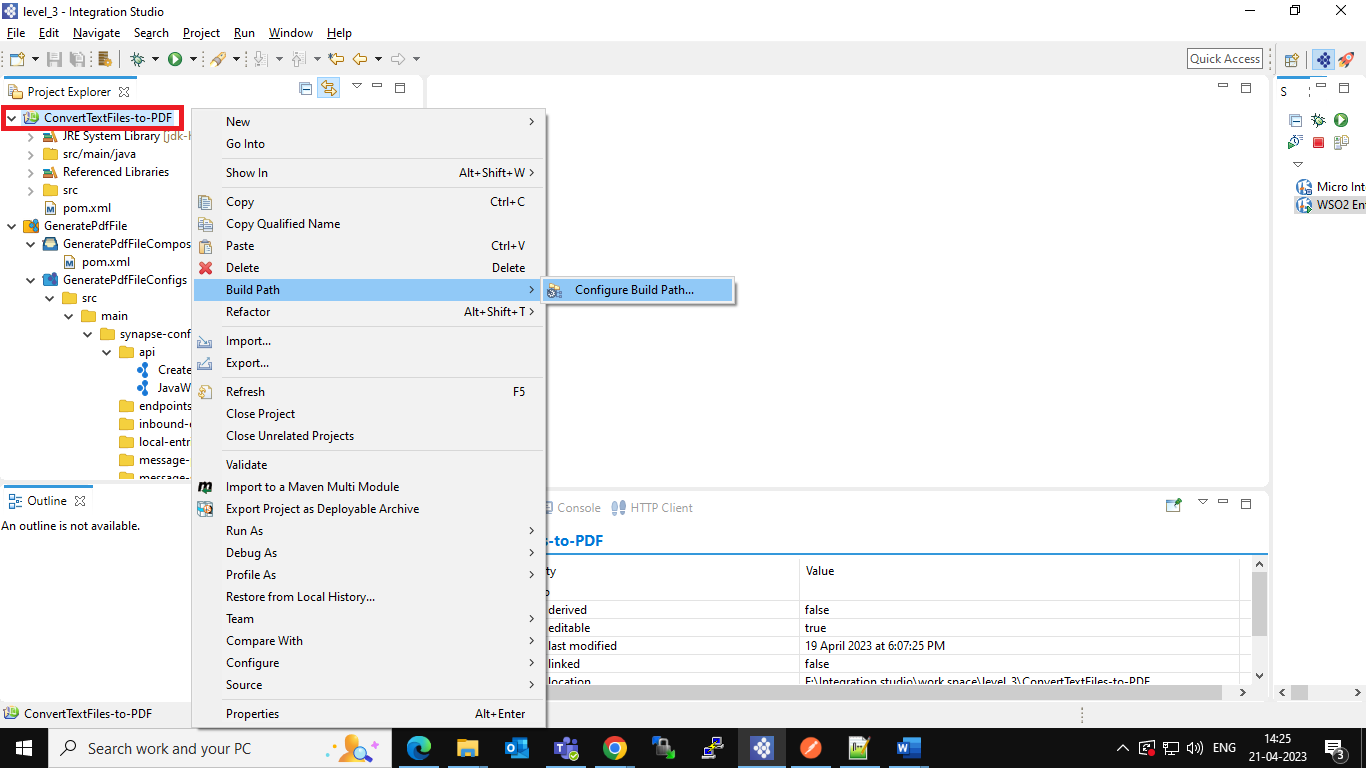


1. In the dialog that opens, select **Create New Mediator** and click **Next**.
2. Enter a project name, package name, and class name.

Graphical user interface, text, application, email

Description automatically generated

1. Add Java library of create PDF [itextpdf-5.5.13.2.jar](https://bit.ly/3v5ZumI) to the mediator project as shown below.
2. Right-click on Mediation project **(ConvertTextFiles-to-PDF)->Build Path->Configure Build Path,** then click on **libraries** in that click on **Add External JARS** go to the file location of downloaded java library and select it.



After adding the library to the libraries, you can see that jar file in the list as shown below then click **Apply and Close.**

A screenshot of a computer

Description automatically generated

* Also add this **JAR** file in the Jdk-home at this path **\jdk-home\jre\lib\ext**

1. Write the Java logic in the created class.

**In the class add the below code:**

package com.knot;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import org.apache.synapse.MessageContext;

import org.apache.synapse.mediators.AbstractMediator;

import com.itextpdf.text.Document;

import com.itextpdf.text.DocumentException;

import com.itextpdf.text.Paragraph;

import com.itextpdf.text.pdf.PdfWriter;

public class ConvertTextToPdf extends AbstractMediator {

public boolean mediate(MessageContext mc) {

String msg = (String) mc.getProperty("message");

mc.setProperty("out", msg);

Document doc = new Document();

try

{

//generate a PDF at the specified location

PdfWriter writer = PdfWriter.getInstance(doc, new FileOutputStream("E:\\EI\\created-csv\\in\\Motivation.pdf"));

System.out.println("PDF created.");

//opens the PDF

doc.open();

//adds paragraph to the PDF file (

doc.add(new Paragraph(msg));

//close the PDF file

doc.close();

//closes the writer

writer.close();

}

catch (DocumentException e)

{

e.printStackTrace();

}

catch (FileNotFoundException e)

{

e.printStackTrace();

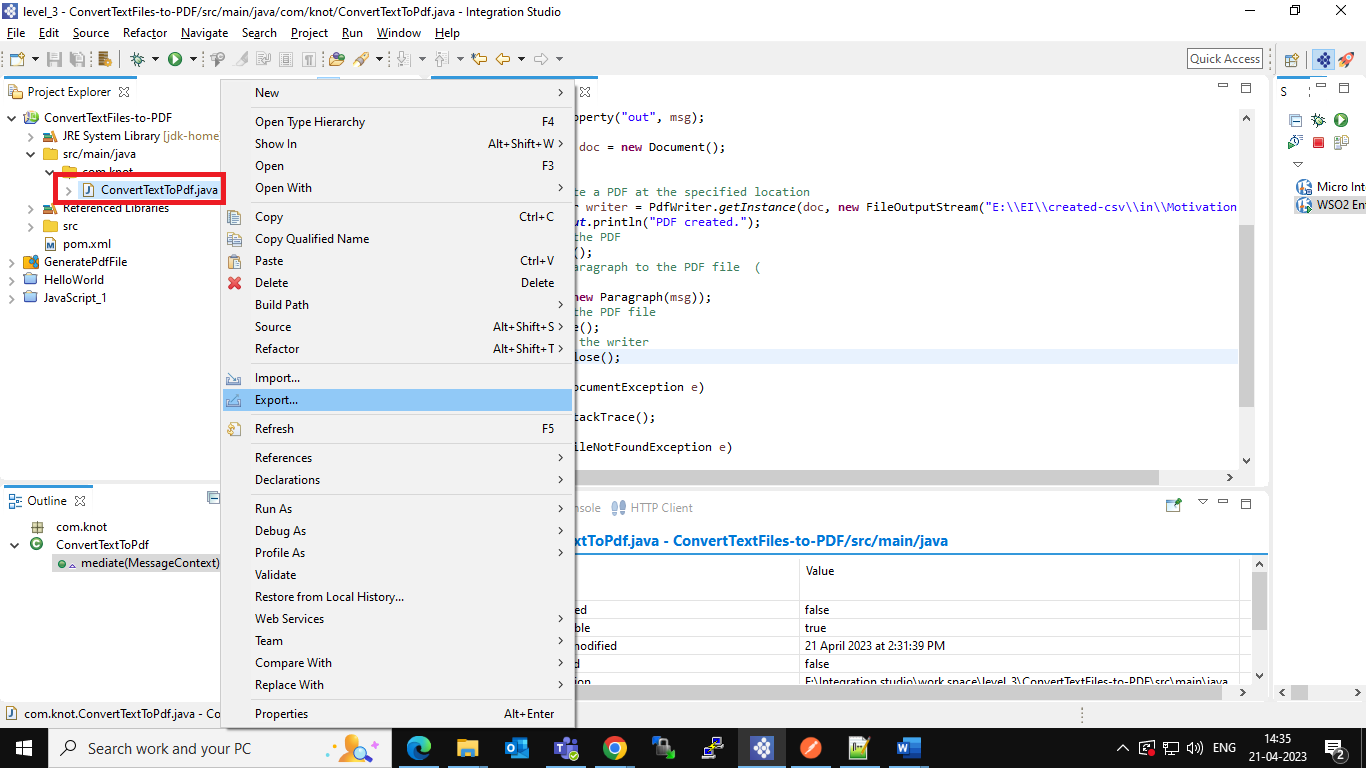
}

return true;

}

}

1. After completion of code, export your project as **JAR** file and save it in your system at any location then click on **Finish.**



A screenshot of a computer

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Now open the Integration project that we created initially (**GeneratePdfFile**), right-click on **GeneratePdfFileCompositeExporter ->Export Project Artifacts and Run.**

A screenshot of a computer

Description automatically generated

1. Add the **JAR** file of mediator project which we are exported in previous step, click on **Configure Embedded Micro Integrator Server** which is shown in below image.

A screenshot of a computer

Description automatically generated

Click on (**+**) add-icon as shown below then browse the JAR file and click **Open.**

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

After adding the mediator project JAR, you can see that in the server as shown below. Then click on **Save.**

A screenshot of a computer

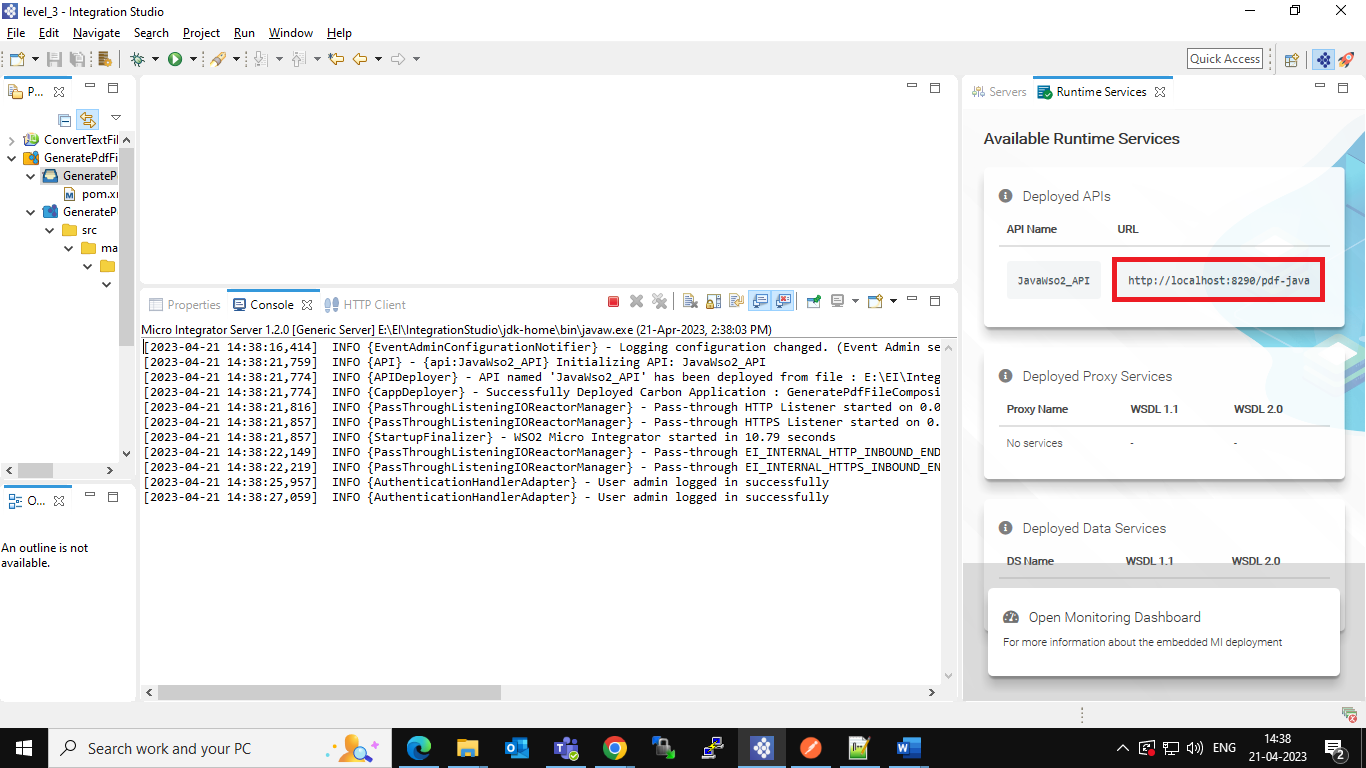
Description automatically generated with medium confidence

1. From the artifacts select **JavaWso2\_API** artifact click on **Finish.**

A screenshot of a computer

Description automatically generated

1. After successful deployment you can see the successful information in the **console** and on the right-side **Runtime Services** tab you can see the API.



1. Open the Postman and copy the URL which is highlighted in the above and paste it in the postman and call this service with **POST** method and use following body.

Graphical user interface, text, application, email

Description automatically generated

1. After successful calling a PDF file is created at the path you have provided in the mediator project which contains the message that you have given as body in the postman.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated