Client Name : KPMG

Project Name : Procurement and parsing of HTML, XML, PDF documents.

1. We are using **JSOUP API, PDF box, Adobe API** library

Controller Name : SingaporeProcrementsAndParsingController

@RestContoller

@RequestMapping("procureAndParse")

class SingaporeProcrementsAndParsingController{

@Getmapping("Procure")

public void procure(){

}

}

Production Log check :

**1. kubectl :**

**i. az login**

**ii. az ask get - credential**

**iii. kubectl get pods**

**iv. kubectl logs -f rcm-source-us-deployments(Pod name**)

Dev Team using Kubernetes :

**It is mainly used to automate the deployment, scaling, and operations of the container-based applications across the cluster of nodes. It is also designed for managing the services of containerized apps using different methods which provide the scalability, predictability, and high availability.**

We are using Azure for data storage.

Database table Name :

**1. job\_metadata**

**2. Sourcing\_details**

**3. Sourcing\_status**

**4. Procurement Status**

**5. Parsing status.**

For tickets We are using JIRA or azure.

How to Download files :

**import** java.io.BufferedInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.net.URL;

**import** java.nio.channels.Channels;

**import** java.nio.channels.ReadableByteChannel;

**public** **void** downloadUsingStream(String urlStr, String file) **throws** IOException {

URL url = **new** URL(urlStr);

BufferedInputStream bis = **new** BufferedInputStream(url.openStream());

FileOutputStream fis = **new** FileOutputStream(file);

**byte**[] buffer = **new** **byte**[1024];

**int** count = 0;

**while** ((count = bis.read(buffer, 0, 1024)) != -1) {

fis.write(buffer, 0, count);

}

fis.close();

bis.close();

}

JSOUP API.

 Document doc = Jsoup.parse(new File("e:\\register.html"),"utf-8");

String title = doc.title();

 Document doc = Jsoup.connect("http://www.javatpoint.com").get();

     Elements links = doc.select("a[href]");

PDF BOX :

 File file = **new** File("/eclipse-workspace/blank.pdf");

PDDocument doc = PDDocument.load(file);

  //Instantiate PDFTextStripper class

PDFTextStripper pdfStripper = **new** PDFTextStripper();

 //Retrieving text from PDF document

String text = pdfStripper.getText(doc);