

Compte rendu d'activités

Réalisé lors du stage de 1^{er} année à la DGFIP

Environnement de travail :

Dans le cadre de ce stage, j'ai intégré l'équipe PCOM (Portail Collaboratif et Outils de Mobilité) pour créer un petit programme en Java qui permettrait de convertir des logs d'erreurs Tomcat en graphique Excel. Mes documents étaient hébergés sur un espace dédié dans NOVAE, un espace collaboratif. On m'a confié un ordinateur avec un OS Ubuntu, et comme éditeur de code Eclipse.

De plus, je pouvais communiquer régulièrement avec le reste de l'équipe via e-mails ou en face à face.

Missions réalisées

En tant que stagiaire, on m'a demandé de réaliser une première version du programme qui convertirait les logs catalina.out d'une seule journée en un fichier Excel triant la criticité des messages. J'ai pour cela utilisé la librairie ApachePOI qui permet de créer des fichiers Excel. Le programme contenait alors trois fichiers : Main, Excel.java ainsi que LogEntry.java, la classe contenant les données du log.

Extrait du fichier Main

```
List<LogEntry> logs = new ArrayList<>();

try (FileReader fileReader = new FileReader(path);
    BufferedReader reader = new BufferedReader(fileReader)) {
    String line = reader.readLine();
    int numE = 0;
    int numW = 1;
    int numI = 1;

    while (line != null) {

        String host = extractFromLog(line, "host\\:\\\"", 12);
        String date = extractFromLog(line, "@timestamp\\:\\\"", 24);

        if (line.contains("\\\"level\\\":\\\"INFO\\\"")) {
            logs.add(new LogEntry(line, "INFO", numI, numW, numE, host, date));
            numI++;
        } else if (line.contains("WARNING") || line.contains("WARN") || line.contains("\\\"level\\\":\\\"WARN\\\"")) {
            logs.add(new LogEntry(line, "Warning", numI, numW, numE, host, date));
            numW++;
        } else if (line.contains("ERROR") || line.contains("error")) {
            logs.add(new LogEntry(line, "Error", numI, numW, numE, host, date));
            numE++;
        }
        line = reader.readLine();
    }

} catch (IOException e) {
    e.printStackTrace();
}

logs.sort(Comparator.comparing(LogEntry::getSeverity, (s1, s2) -> {
    if (s1.equals("Warning") && !s2.equals("Warning")) {
```

Extrait du fichier LogEntry

```
public class LogEntry {  
  
    public String msg;  
    public String severity;  
    public int occurrenceI;  
    public int occurrenceE;  
    public int occurrenceW;  
    public String host;  
    public Date date;  
  
    public LogEntry(String msg, String severity, int numI, int numW, int numE, String host, String date) {  
        this.msg = msg;  
        this.severity = severity;  
        this.occurrenceI = numI;  
        this.occurrenceW = numW;  
        this.occurrenceE = numE;  
        this.host = host;  
        SimpleDateFormat nDate = new SimpleDateFormat("yyyy-MM-DD'T'HH:mm:ss.SSS'Z'");  
        try {  
            this.date = nDate.parse(date);  
        } catch (ParseException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Extrait du fichier Excel.java

```
Workbook wb = new XSSFWorkbook();  
Sheet sheet = wb.createSheet("Logs");  
  
// Header  
Row headerRow = sheet.createRow(0);  
CellStyle cellStyle = wb.createCellStyle();  
cellStyle.setAlignment(HorizontalAlignment.CENTER);  
Cell headerCell1 = headerRow.createCell(0);  
headerCell1.setCellStyle(cellStyle);  
headerCell1.setCellValue("Criticit ");  
Cell headerCell2 = headerRow.createCell(1);  
headerCell2.setCellStyle(cellStyle);  
headerCell2.setCellValue("Date");  
Cell headerCell3 = headerRow.createCell(2);  
headerCell3.setCellStyle(cellStyle);  
headerCell3.setCellValue("Message");  
Cell headerCell4 = headerRow.createCell(3);  
headerCell4.setCellStyle(cellStyle);  
headerCell4.setCellValue("Host");  
Cell headerCell5 = headerRow.createCell(4);  
headerCell5.setCellValue("Occurrence");  
headerCell5.setCellStyle(cellStyle);  
  
sheet.setColumnWidth(2, 5000);  
CellStyle style = wb.createCellStyle();  
style.setWrapText(true);  
CellStyle dateStyle = wb.createCellStyle();  
CreationHelper createHelper = wb.getCreationHelper();  
short format = createHelper.createDataFormat().getFormat("dd/mm/yy hh:mm:ss.SSS");  
dateStyle.setDataFormat(format);
```

Extrait du fichier Excel final

Criticité	Date	Message	Host	Occurrence		
				Error: 0	Warning: 1026	Info: 8598
Warning		WARNING: An illegal re				
Warning		WARNING: Illegal reflec				
Warning		WARNING: Please cons				
Warning		WARNING: Use --illegal				
Warning		WARNING: All illegal ac				
Warning		2023-06-14 12:40:43.06				
Warning		log4j: WARN Continuable				
Warning		log4j: WARN The conten				
Warning		2023-06-14 12:41:33.17				
Warning	14/01/23 10:42:58:5858	["@timestamp"."2023-0	e517d8f5cf93			
Warning	14/01/23 10:42:59:5959	["source_host"."e517d8f	e517d8f5cf93			
Warning	14/01/23 10:43:02:022	["@timestamp"."2023-0	e517d8f5cf93			
Warning	14/01/23 10:43:02:022	["@timestamp"."2023-0	e517d8f5cf93			
Warning	14/01/23 10:43:02:022	["@timestamp"."2023-0	e517d8f5cf93			
Warning	14/01/23 10:43:03:033	["@timestamp"."2023-0	e517d8f5cf93			
Warning	14/01/23 10:43:03:033	["@timestamp"."2023-0	e517d8f5cf93			
Warning	14/01/23 10:43:03:033	["@timestamp"."2023-0	e517d8f5cf93			
Warning	14/01/23 10:43:03:033	["@timestamp"."2023-0	e517d8f5cf93			
Warning	14/01/23 10:43:03:033	["@timestamp"."2023-0	e517d8f5cf93			
Warning	14/01/23 10:43:03:033	["@timestamp"."2023-0	e517d8f5cf93			

Pour sa deuxième itération, le programme devait pouvoir envoyer ce fichier Excel par mail en choisissant le destinataire. J'ai donc utilisé la librairie Javax.mail pour cela, dans le nouveau fichier SendMail.java. À ce moment-là, le fichier Excel.java demandait les identifiants pour l'envoi par mail.

Extrait du fichier Main

```

    } else if (line.contains("ERROR") || line.contains("error")) {
        logs.add(new LogEntry(line, "Error", numI, numW, numE, host, date));
        numE++;
    }
    line = reader.readLine();
}

} catch (IOException e) {
    e.printStackTrace();
}

// Comparing the severity between warning and information
logs.sort(Comparator.comparing(LogEntry::getSeverity, (s1, s2) -> {
    if (s1.equals("Warning") && !s2.equals("Warning")) {
        return -1;
    } else if (!s1.equals("Warning") && s2.equals("Warning")) {
        return 1;
    } else {
        return 0;
    }
}));

// Generate Excel file
String filePath = "logToExcel.xlsx";
int reponse = JOptionPane.showConfirmDialog(null, "Send file by email?", null, JOptionPane.YES_NO_OPTION);
if (reponse == 0) {
    Excel.createExcelFile(logs, filePath, false, path);
} else {
    Excel.createExcelFile(logs, filePath, true, path);
}
}

```

Extrait du fichier Excel.java

```
/*
 * Display a password field for the email address
 *
 * @return The password field
 */
private static String askCredentials() {
    JPanel panel = new JPanel();
    final JPasswordField passwordField = new JPasswordField(20);
    panel.add(new JLabel("Enter your email password:"));
    panel.add(passwordField);
    JOptionPane pane = new JOptionPane(panel, JOptionPane.QUESTION_MESSAGE, JOptionPane.OK_CANCEL_OPTION) {
        private static final long serialVersionUID = 1L;
        @Override
        public void selectInitialValue() {
            passwordField.requestFocusInWindow();
        }
    };
    pane.createDialog(null, "Password Required").setVisible(true);
    return passwordField.getPassword().length == 0 ? null : new String(passwordField.getPassword());
}

/**
 * Show a message dialogue informing of the creation of the Excel file and opening it if desktop environment supported
 *
 * @param path Path of the Excel file to create
 */
private static void showFileCreatedDialog(String path) {
    Path file = FileSystems.getDefault().getPath(path);
    String msg = "File " + file.getFileName() + " create";
    JOptionPane.showMessageDialog(null, msg);
}
```

Extrait du fichier SendMail

```
//compose the message
try {
    MimeMessage message = new MimeMessage(session);
    message.setFrom(new InternetAddress(from));
    message.setRecipient(Message.RecipientType.TO, new InternetAddress(to));
    message.setSubject("logToExcel result file");

    // Body of the message
    BodyPart messageBodyPart = new MimeBodyPart();
    messageBodyPart.setText(msg);

    // Attachments
    MimeBodyPart attachementPart = new MimeBodyPart();
    attachementPart.attachFile(new File("logToExcel.xlsx"));

    // Creation of the multipart content
    Multipart multipart = new MimeMultipart();
    multipart.addBodyPart(messageBodyPart);
    multipart.addBodyPart(attachementPart);

    // Application of the multipart content
    message.setContent(multipart);

    // Send message
    Transport.send(message);

    // Display a message to inform of sending
    String info = "Mail send succesfully";
    JOptionPane.showMessageDialog(null, info);

    // Terminate the program
    System.exit(0);
}
```


Une deuxième version fut produite afin de permettre l'envoi à plusieurs destinataires.

Extrait du fichier Excel.java

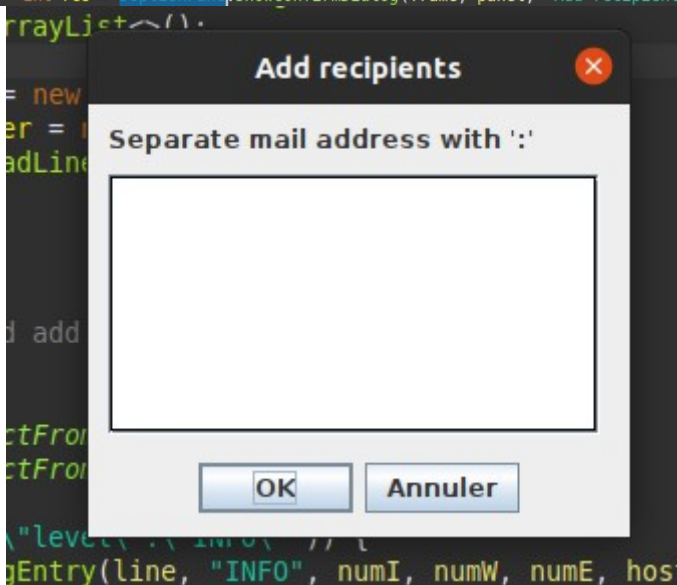
```
* Display a window with a text area for implementing the recipients list
*
* @param recipients List of recipients for the mailing
* @param username Username for the case the user make a mistake
* @param password Password for the case the user make a mistake
* @param path Path of the Excel file for the case the user make a mistake
*/
private static void more(List<String> recipients, List<String> username, List<String> password, String path) {

    // Creating a JFrame for a JPanel with a layout of type springLayout
    JFrame frame = new JFrame();
    final SpringLayout layout = new SpringLayout();
    final JPanel panel = new JPanel(layout);
    panel.setPreferredSize(new Dimension(250, 160));

    // Adding a label and a textArea for the recipients to add
    JLabel lblSeparator = new JLabel("Separate mail address with \':\'");
    panel.add(lblSeparator);
    JTextArea txtRecipients = new JTextArea();
    txtRecipients.setBorder(BorderFactory.createLineBorder(Color.black));
    txtRecipients.setLineWrap(true);
    txtRecipients.setWrapStyleWord(true);
    JScrollPane scrollPane = new JScrollPane(txtRecipients); // Adding a scroll in case of a long list
    scrollPane.setPreferredSize(new Dimension(245, 130));
    panel.add(scrollPane);

    // Setting constraints for the layout
    layout.putConstraint(SpringLayout.WEST, lblSeparator, 0, SpringLayout.WEST, panel);
    layout.putConstraint(SpringLayout.WEST, txtRecipients, 25, SpringLayout.EAST, lblSeparator);
    layout.putConstraint(SpringLayout.NORTH, scrollPane, 10, SpringLayout.SOUTH, lblSeparator);

    // Adding recipients to the list
    int res = JOptionPane.showConfirmDialog(frame, panel, "Add recipients", JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE)
```



Pour la troisième version, il m'a été demandé de générer des graphiques Excel sur plusieurs jours avec un graphique principal regroupant tous les jours du fichier catalina.out. Pour cela, des modifications furent apportées au niveau du Main, qui gère dorénavant la demande d'envoi par mail, et le fichier Excel.java fut alourdi pour générer les graphiques.

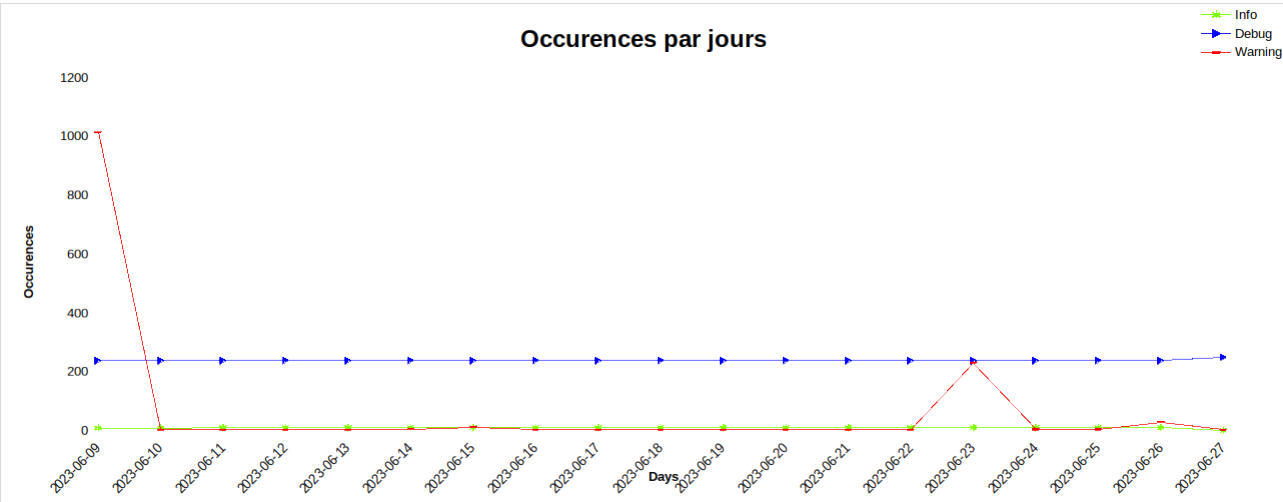
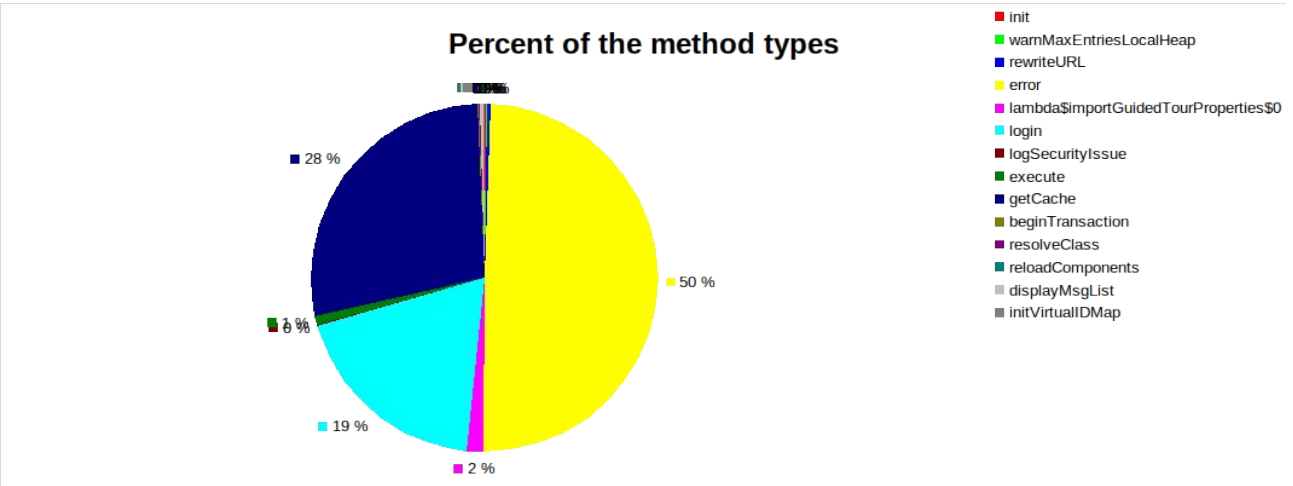
Extrait du fichier Main.java

```
/**
 * Extract a part of a line log and return it
 *
 * @param line      Line of log
 * @param separator Separator from when to start extracting
 * @param count     Number of characters after the separator
 * @return The line of log with only the information wanted or a space character
 */
private static String extractFromLog(String line, String separator) {

    String reg = separator + "\\:" + "(" + ".*?" + ")";
    Pattern p = Pattern.compile(reg);
    Matcher m = p.matcher(line);
    if (m.find()) {
        line = m.group(0);
        line = line.substring(separator.length() + 3, line.length() - 1);
        return line;
    } else {
        return " ";
    }
}

/**
 * Divide a list of LogEntry into several smaller list regroup by date and add
 * them to a list of list of LogEntry
 *
 * @param logs The main list to divide
 * @return The list of list
 */
private static List<List<LogEntry>> groupLogsByDay(List<LogEntry> logs) {
    List<List<LogEntry>> logsByDay = new ArrayList<>();
```

Extrait des graphiques



Extrait du fichier Excel.java

```
// Create a legend
XDDFChartLegend legend = chart.getOrAddLegend();
legend.setPosition(LegendPosition.TOP_RIGHT);

// Set the data sources from cell of the sheet
XDDFDataSource<String> categories = XDDFDataSourcesFactory.fromStringCellRange((XSSFSheet) sheet1,
    new CellRangeAddress(4, 4 + method.size() - 1, 7, 7));
XDDFNumericalDataSource<Double> values = XDDFDataSourcesFactory.fromNumericCellRange((XSSFSheet) sheet1,
    new CellRangeAddress(4, 4 + method.size() - 1, 8, 8));

// Create a chart of type pie
XDDFChartData data = chart.createData(ChartTypes.PIE, null, null);
data.setVaryColors(true);
XDDFChartData.Series series = data.addSeries(categories, values);
chart.plot(data);

// Create the DLbls object if it doesn't exist
if (!chart.getCTChart().getPlotArea().getPieChartArray(0).getSerArray(0).isSetDLbls()) {
    chart.getCTChart().getPlotArea().getPieChartArray(0).getSerArray(0).addNewDLbls();
}

CTDLbls dLbls = chart.getCTChart().getPlotArea().getPieChartArray(0).getSerArray(0).getDLbls();

dLbls.addNewDLblPos().setVal(org.openxmlformats.schemas.drawingml.x2006.chart.STDLblPos.OUT_END);
dLbls.addNewShowLegendKey().setVal(true);
dLbls.addNewShowPercent().setVal(true);
dLbls.addNewShowCatName().setVal(false);
dLbls.addNewShowLeaderLines().setVal(false);
dLbls.addNewShowVal().setVal(false);
dLbls.addNewShowSerName().setVal(false);

// Do not auto delete the title; is necessary for showing title in LibreOffice
```

Une quatrième version m’a été demandée. Je devais alors générer depuis le fichier Excel une page HTML et l’envoyer par mail. Mais par manque de connaissances et de temps, je n’y suis pas arrivé.