Erweiterungen der FileUploadBean:

```
public List<FileUpload> getPageItems() {
	return this.pageItems;
}

public List<FileUpload> getFoundPageItems() {
	return this.foundPageItems;
}

public long getCount() {
	return (!foundPageItems.isEmpty())? foundPageItems.size() : this.count;
}


private String searchText;

public String getSearchText() {
	return searchText;
}

public void setSearchText(String searchText) {
	this.searchText = searchText;
}


public void searchForText() {
	paginate();
	Logger.getAnonymousLogger().info("searchText: " + searchText);
	foundPageItems.clear();
	if (searchText == null || searchText.trim().isEmpty()) {
		foundPageItems.addAll(pageItems);
		return;
	}
	for (FileUpload oneUpload : pageItems) {
		byte[] cv = oneUpload.getCv();
		byte[] coverLetter = oneUpload.getCoverLetter();
		byte[] testimonial1 = oneUpload.getTestimonial1();
		byte[] testimonial2 = oneUpload.getTestimonial2();
		byte[] testimonial3 = oneUpload.getTestimonial3();
		byte[] testimonial4 = oneUpload.getTestimonial4();
		byte[] testimonial5 = oneUpload.getTestimonial5();

		try {
			String extractCoverLetter = "";
			if (coverLetter != null) {
				String textCoverLetter = parsePDF(coverLetter);
				extractCoverLetter = extractContent(textCoverLetter, "Sehr");
```

```
		String content = replaceLineBreaks(extractCoverLetter);
		if (content.indexOf(searchText) >= 0) {
			foundPageItems.add(oneUpload);
			continue;
		}
}
String extractCv = "";
if (cv != null) {
		String textCv = parsePDF(cv);
		extractCv = extractContent(textCv, new String[] {"Lebenslauf","CV","C.V."} );
		String content = replaceLineBreaks(extractCv);
		if (content.indexOf(searchText) >= 0) {
			foundPageItems.add(oneUpload);
			continue;
		}
}
String extractTestimonial1 = "";
if (testimonial1 != null) {
		String textTestimonial1 = parsePDF(testimonial1);
		extractTestimonial1 = extractContent(textTestimonial1, "eugnis");
		String content = replaceLineBreaks(extractTestimonial1);
		if (content.indexOf(searchText) >= 0) {
			foundPageItems.add(oneUpload);
			continue;
		}
}
String extractTestimonial2 = "";
if (testimonial2 != null) {
		String textTestimonial2 = parsePDF(testimonial2);
		extractTestimonial2 = extractContent(textTestimonial2, "eugnis");
		String content = replaceLineBreaks(extractTestimonial2);
		if (content.indexOf(searchText) >= 0) {
			foundPageItems.add(oneUpload);
			continue;
		}
}
String extractTestimonial3 = "";
if (testimonial3 != null) {
		String textTestimonial3 = parsePDF(testimonial3);
		extractTestimonial3 = extractContent(textTestimonial3, "eugnis");
		String content = replaceLineBreaks(extractTestimonial3);
		if (content.indexOf(searchText) >= 0) {
			foundPageItems.add(oneUpload);
			continue;
		}
}
String extractTestimonial4 = "";
if (testimonial4 != null) {
		String textTestimonial4 = parsePDF(testimonial4);
```

```
                                        extractTestimonial4 = extractContent(textTestimonial4, "eugnis");
                                        String content = replaceLineBreaks(extractTestimonial4);
                                        if (content.indexOf(searchText) >= 0) {
                                                foundPageItems.add(oneUpload);
                                                continue;
                                        }
                                }
                                String extractTestimonial5 = "";
                                if (testimonial5 != null) {
                                        String textTestimonial5 = parsePDF(testimonial5);
                                        extractTestimonial5 = extractContent(textTestimonial5, "eugnis");
                                        String content = replaceLineBreaks(extractTestimonial5);
                                        if (content.indexOf(searchText) >= 0) {
                                                foundPageItems.add(oneUpload);
                                                continue;
                                        }
                                }
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                }
        }

        private String extractContent(String txt, String[] starts) {
                String result = "";
                for (String s : starts) {
                        result = extractContent(txt, s);
                        if (result != null && !result.isEmpty()) {
                                break;
                        }
                }
        return result;
        }

        private String extractContent(String txt, String start) {
                if (txt == null || txt.isEmpty()) {
                        return txt;
                }
                String result = "";
                String[] split1 = txt.split(start);
                if (split1 != null && split1.length > 1 && split1[1] != null) {
                        result = split1[1];
                } else if (split1 != null && split1.length == 1) {
                        result = split1[0];
                }
                return result;
        }
```

```
private String parsePDF(byte[] pdfIn) throws IOException {
        PdfReader pdfReader = new PdfReader(pdfIn);
        StringBuilder builder = new StringBuilder();
        for (int counter = 1; counter <= pdfReader.getNumberOfPages(); ++counter) {
            TextExtractionStrategy tExtractStrategy = new SimpleTextExtractionStrategy();
            builder.append(PdfTextExtractor.getTextFromPage(pdfReader, counter, tExtractStrategy));
        }
        pdfReader.close();
        return builder.toString();
}

private String replaceLineBreaks(String input) {
        String result = "";
        String[] lines = input.split("\n");
        for (String line : lines) {
            if (result.isEmpty()) {
                result = line;
            } else {
                result += ", " + line;
            }
        }
        return result;
}
```