Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа № 2

По дисциплине ЕЯИИС

Тема: «Разработка системы автоматического реферирования документов»

Выполнил:

Студент 4 курса

Группы ИИ-16(2)

Пешко А.С.

Проверил:

Скарубо А.О.

Брест, 2021

**Цель работы:** освоить на практике основные принципы автоматического реферирования документов.

**Ход работы:**

**Код программы:**

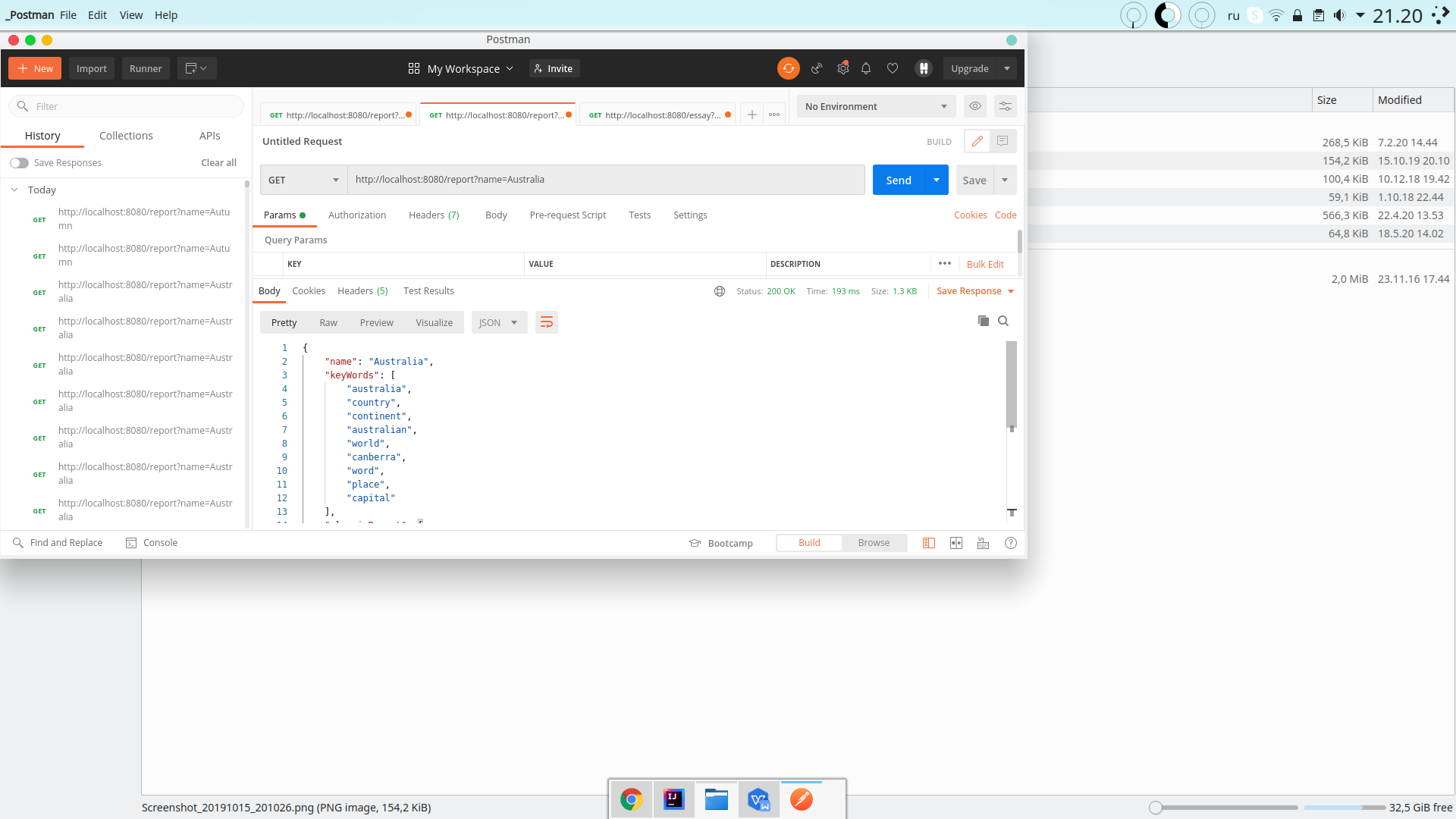
@Service  
public class KeyWordReportServiceImpl {  
  
 Map<String, Integer> countWordsInFile = new HashMap<>();  
  
 public Set<String> getKeyWords(String finalDataFromFile, List<String> paragraphs) {  
  
 List<String> fileWithWords = Arrays.asList(finalDataFromFile  
 .split(ReportServiceImpl.REGEX));  
  
 fileWithWords = fileWithWords.stream()  
 .map(String::toLowerCase)  
 .collect(Collectors.toList());  
  
 int count;  
 for (String word : fileWithWords) {  
 if (countWordsInFile.containsKey(word)) {  
 count = countWordsInFile.get(word) + 1;  
 } else {  
 count = 1;  
 }  
 countWordsInFile.put(word, count);  
 }  
  
 Set<String> sortedKeySet = countWordsInFile.keySet()  
 .stream()  
 .filter(word -> !word.isEmpty() && ReportServiceImpl.isSensibleWord(word))  
 .sorted((e1, e2) -> countWordsInFile.get(e2).compareTo(countWordsInFile.get(e1)))  
 .limit(paragraphs.size())  
 .collect(Collectors.toCollection(LinkedHashSet::new));  
  
 return sortedKeySet;  
 }  
  
 public Map<String, Integer> getCountWordsInFile() {  
 return this.countWordsInFile;  
 }  
}

@Service  
public class ClassicReportServiceImpl {  
  
 public Set<String> getSentences(List<String> paragraphs, String finalDataFromFile, Map<String, Integer> allWordsWithCount) throws IOException {  
  
 Map<String, Integer> sentenceList = new HashMap<>();  
 List<String> sentences = Arrays.stream(finalDataFromFile.split("[.!?]"))  
 .filter(sentence -> !sentence.isEmpty())  
 .map(sentence -> sentence = sentence.trim())  
 .collect(Collectors.toList());  
  
 for (String s : sentences) {  
 int score = 0;  
 List<String> wordsInSentence = Arrays.stream(s.split(ReportServiceImpl.REGEX))  
 .filter(word -> !word.isEmpty() && ReportServiceImpl.isSensibleWord(word))  
 .map(String::toLowerCase)  
 .collect(Collectors.toList());  
 for (String wordInSentence : wordsInSentence) {  
 score += allWordsWithCount.get(wordInSentence);  
 }  
 sentenceList.put(s, score);  
 }  
  
 Set<String> finalSen = new LinkedHashSet<>();  
 for (String par : paragraphs) {  
 int maxSenScore = 0;  
 String sen= null;  
 for (Map.Entry<String, Integer> entry : sentenceList.entrySet()) {  
 if (par.contains(entry.getKey()) &&  
 entry.getValue() > maxSenScore) {  
 maxSenScore = entry.getValue();  
 sen = entry.getKey();  
 }  
 }  
 finalSen.add(sen);  
 }  
  
 return finalSen;  
 }  
}  
  
@Service  
@RequiredArgsConstructor  
public class ReportServiceImpl {  
  
 public static final String REGEX = "[ ,.;:\"'»\\-«()—!\\[\\]\n]+";  
  
 private final ClassicReportServiceImpl classicReportService;  
 private final KeyWordReportServiceImpl keyWordReportService;  
  
 public Report getReport(String reportName) throws IOException {  
  
 FileReaderServiceImpl fileReaderService = new FileReaderServiceImpl(reportName);  
  
 String dataFromFile = fileReaderService.readDataFromFile();  
 List<String> paragraphs = fileReaderService.readFileByParagraphs();  
  
 Set<String> keyWords = keyWordReportService.getKeyWords(dataFromFile, paragraphs);  
 Map<String, Integer> countWords = keyWordReportService.getCountWordsInFile();  
 Set<String> sentences = classicReportService.getSentences(paragraphs, dataFromFile, countWords);  
  
 return Report.builder()  
 .name(reportName)  
 .keyWords(keyWords)  
 .classicReport(sentences)  
 .build();  
 }  
 }

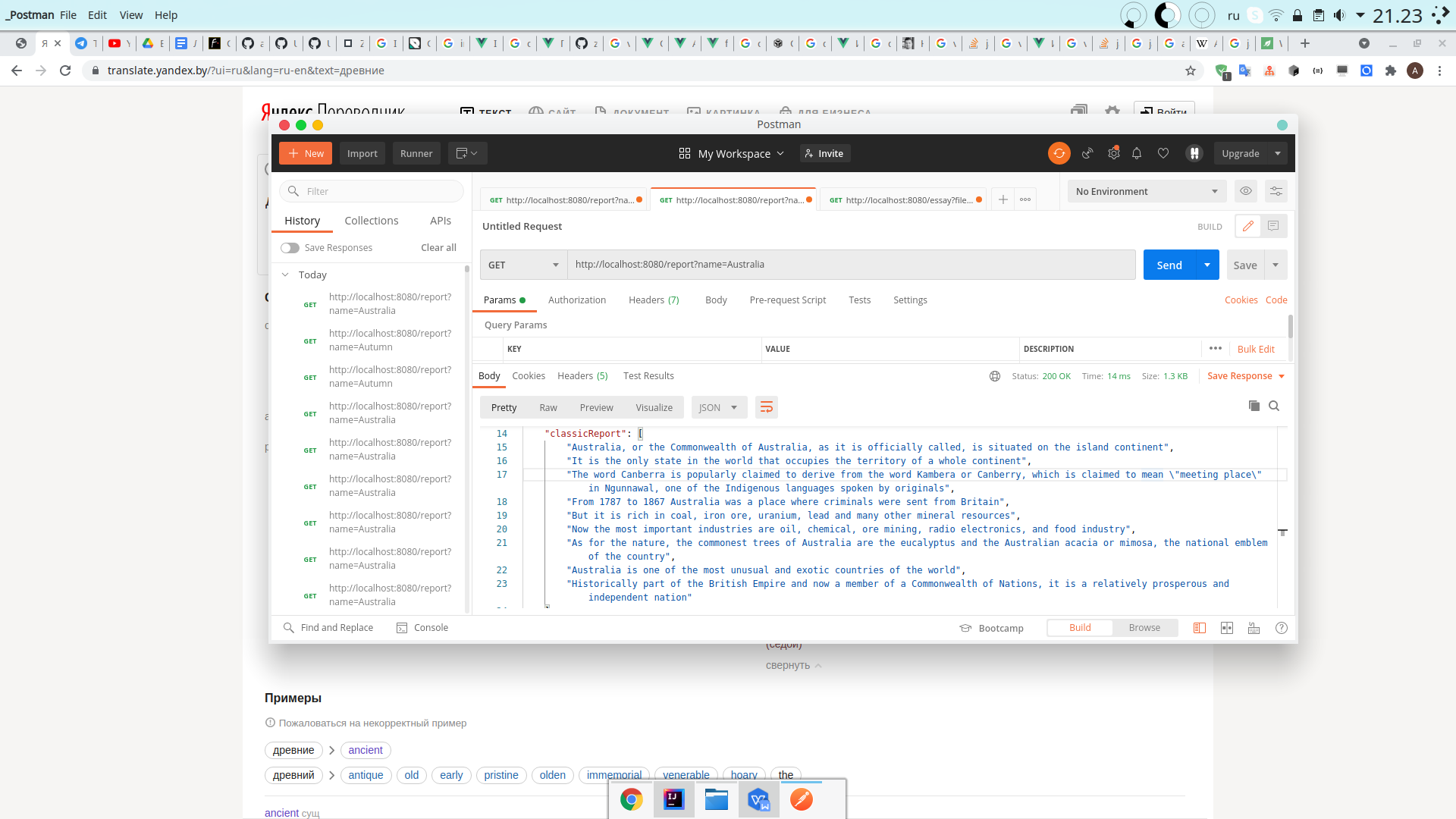
**Тестирование:**

Передаем в качестве входного параметра имя реферата. В данном случае: ‘Australia’

Реферат в виде ключевых слов:



Классический реферат:



**Вывод:** освоили на практике основные принципы автоматического реферирования документов.