|  |  |
| --- | --- |
|

|  |
| --- |
| МИНОБРНАУКИ РОССИИФедеральное государственное бюджетное образовательное учреждениевысшего образования**«МИРЭА – Российский технологический университет»****РТУ МИРЭА** |

 |

Институт Информационных Технологий

Кафедра корпоративных информационных систем (КИС)

**ОТЧЕТ**

по Лабораторной работе №5

по дисциплине

«Разработка программного обеспечения для корпоративных информационных систем»

Выполнил студент группы ИКБО-08-18 Валяев Д.А.

Принял ассистент Масленников В.В.

Выполнено «\_\_\_» \_\_\_\_\_\_\_\_\_2021 г.

Зачтено «\_\_\_» \_\_\_\_\_\_\_\_\_2021 г.

**Задание**

Реализовать настраиваемый модуль экспорта выбранных пользователем данных в промежуточный формат хранения и обмена.

Пользователь должен иметь возможность в некоторой степени выбирать состав и объем данных, а также формат, в который будет производиться экспорт. Для ЛР №3 должна быть возможность выбора ветки/набора веток, количества сообщений.

Экспорт должен осуществляться в форматы CSV (comma-separated values), XML, JSON. Использовать встроенные в фреймворк, на котором построен ресурс, функции экспорта нельзя. Взаимодействие должно ограничиваться уровнем базы данных (использовать копию фреймворка с другими настройками, независимую от основной — можно). Механизм экспорта должен предусматривать некоторую гибкость конфигурации — в частности, иметь возможность указывать состав и названия полей и таблиц в БД, откуда брать информацию для экспорта, т.е. быть универсальным и иметь возможность извлекать подобную информацию из любой базы данных с аналогичной или совместимой архитектурой.

Дополнительные задания (одно задание на выбор):

Реализовать экспорт в форматы Office Open XML (docx, xlsx)

**Содержание работы**

1. Реализация основного функционала
	1. Создание страницы экспорта
	2. Экспорт в формат csv
	3. Экспорт в формат xml
	4. Экспорт в формат json
2. Реализация дополнительного функционала
	1. Экспорт в формат docx
	2. Экспорт в формат xlsx

**Выполнение работы**

В качестве основы была выбрана лабораторная работа №3 с использованием копии фреймворка для экспорта.

1. **Основной функционал**

1.1 Для начала необходимо создать страницу экспорта для выбора веток, количества сообщений и выходного формата. При переходе на страницу экспорта, на серверной части выбираются все существующие ветки и генерируется страница.

Рис 1.1 – получение веток



Рис 1.2 – страница экспорта

Для обращения к серверной части был написан скрипт на языке программирования javascript.



Рис 1.3 – скрипт для обращения к серверной части

 1.2 Для экспорта сообщений в csv формате, выберем обе ветки, первые 3 сообщения и формат экспорта csv. После нажатия на кнопку «Экспортировать» javascript отправляет запрос на серверную часть, где происходит его обработка.



Рис 1.4 – генерация export.csv



Рис 1.5 – экспорт в формате csv

 При получении запроса начинает формироваться запрос к базе данных. Для начала выбираются поля и ветка, далее происходит определение какие сообщения должны выбираться. Если выбираются первые сообщения, тогда используем LIMIT, в случае с выбором последних сообщений, происходит проверка количества сообщений, если количество сообщений больше запрашиваемого, то используем OFFSET, иначе возвращаются все сообщения. Если требуется выбор сообщений от определенной даты, тогда из текущей даты вычитается требуемое количество дней, для последующего выбора из базы данных записей, где дата больше запрошенной. Последним шагом происходит проверка выбранного типа выходных данный, в зависимости от выбора вызывается определенная функция.



Рис 1.6 – обработка запроса



Рис 1.7 – результат запроса

 1.3 Для экспорта сообщений в xml формате, выберем только вторую ветку, последние 3 сообщения и формат экспорта xml. После нажатия на кнопку «Экспортировать» javascript отправляет запрос на серверную часть, где происходит его обработка. Ответом должен стать файл «export.xml» с двумя сообщениями, потому что во второй ветке их всего 2.



Рис 1.7 – экспорт в формате xml



Рис 1.8 – генерация export.xml



Рис 1.9 – результат запроса

1.4 Для экспорта сообщений в json формате, выберем обе ветки, период сообщений за последние 9 дней и формат экспорта json. После нажатия на кнопку «Экспортировать» javascript отправляет запрос на серверную часть, где происходит его обработка. Ответом должен стать файл «export.json» с одним сообщением, потому что за этот период времени было лишь одно сообщение.



Рис 1.10 – генерация export.json



Рис 1.11 – экспорт в формате json



Рис 1.12 – результат запроса

1. **Дополнительный функционал**

Кроме основного функционала были реализованы дополнительные функции, такие как:

* 1. Экспорт в формате docx. Для получения сообщений в docx формате, выберем обе ветки, период сообщений за последний год и формат экспорта docx. После нажатия на кнопку «Экспортировать» javascript отправляет запрос на серверную часть, где происходит его обработка.



Рис 2.1 – экспорт в формате docx



Рис 2.2 – генерация export.docx



Рис 2.3 – результат запроса

2.2 Экспорт в формате xlsx. Для получения сообщений в xlsx формате, выберем обе ветки, первые 5 сообщений и формат экспорта xlsx. После нажатия на кнопку «Экспортировать» javascript отправляет запрос на серверную часть, где происходит его обработка.



Рис 2.4 – экспорт в формате xlsx



Рис 2.5 – генерация export.xlsx



Рис 2.6 – результат запроса

**Вывод**

 В процессе выполнения данной лабораторной работы были приобретены навыки работы с экспортом данных в разных форматах, генерации файлов для их последующей загрузки.