ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Профессор департамента программной инженерии факультета компьютерных наук, к.т.н

_____ Е. М. Гринкруг
«_____» _____ 2020 г.

УТВЕРЖДАЮ
Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук

_____ В. В. Шилов
«_____» _____ 2020 г.

Инв. № подл | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата

СИСТЕМА УПРАВЛЕНИЯ ЗАДАНИЯМИ ПО АВТОМАТИЧЕСКОМУ СБОРУ ДАННЫХ ИЗ СЕТИ ИНТЕРНЕТ

Руководство оператора

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.13-01 34 01-1

Исполнитель: студент группы БПИ 174
_____ Д. Ю. Редникина
«_____» _____ 2020 г.

Москва 2020

УТВЕРЖДЕН
RU.17701729.04.13-01 34 01-1

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СИСТЕМА УПРАВЛЕНИЯ ЗАДАНИЯМИ ПО АВТОМАТИЧЕСКОМУ СБОРУ ДАННЫХ ИЗ СЕТИ ИНТЕРНЕТ

Руководство оператора

RU.17701729.04.13-01 34 01-1

Листов 11

Москва 2020

Содержание

1 Назначение программы

1.1 Функциональное назначение

Система будет применяться как средство управления проектами по созданию, редактированию и запуску веб краулеров для сбора данных в сети интернет. Продукт позволит следить за запусками в режиме реального времени, а также создавать периодические запуски по расписанию.

1.2 Эксплуатационное назначение

Программа будет использоваться как инструмент для самостоятельной или совместной работы над проектами для запуска, управления сбора данных с помощью веб краулеров в сети интернет.

Таким образом, программный продукт позволит создавать, запускать образы пауков (см. 3.2) для сбора, управления, логирования и дальнейшего экспорта данных в целях сбора, изучения и мониторинга данных (см. 3.2).

1.3 Состав выполняемых функций

Следующие требования зафиксированы в документе «Система управления заданиями по автоматическому сбору данных из сети Интернет. Техническое задание» к составу выполняемых функций:

1. **Авторизация**
   Чтобы использовать сервис, клиентская программа должна иметь возможность авторизоваться в системе с помощью REST API
   (a) Для регистрации пользователю нужно указать следующие данные
      i. Почта - уникальна для каждого зарегистрированного пользователя;
      ii. Имя - длина больше 1 символ;
      iii. Логин - длина больше 2 символов;
      iv. Пароль - длина больше 2 символов;
   (b) Для авторизации пользователя в системе должны быть указаны следующие данные
      i. Почта;
      ii. Пароль;
2. **Проекты**
   Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности
   (a) Создание проекта со следующей информацией
      i. Имя проекта;
      ii. Описание проекта - опциональное поле;
   (b) Обновление метаданных о проекте (редактирование) могут быть обновлены только участником с минимальным уровнем дотупа `ReadAndWrite`. Следующие данные могут быть обновлены:

i. Имя проекта;
ii. Описание проекта;
iii. Настройки проекта для запуска краулеров;
iv. Аргументы для запуска краулеров проекта;

(c) Обновление `egg` файла проекта (редактирование) – минимальный уровень доступа участника, обновляющий данные о проекте `ReadAndWrite`.

(d) Удаление данных о проекте. Удалить проект может только владелец `Owner`.

(e) Просмотр списка проектов (с пагинацией), к которым у пользователя есть как минимум `ReadOnly` доступ.

3. **Участники проектов**
Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

(a) Просмотр информации об участниках проекта;
i. Имя, почта, логин участника;
ii. Статус участника в проекте (`ReadOnly`, `ReadAndWrite` или `Owner`);

(b) Обновление статуса участника проекта. Это действие совершать может только владелец проекта;

(c) Удаление участника из проекта. Данное действие может совершать только владелец проекта;

(d) Добавление нового участника с указанными правами на редактирование. Данное действие может совершать только владелец проекта;

4. **Краулеры**
Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

(a) Просмотр списка краулеров проекта;

(b) Редактирование информации о краулере для последующих запусков. Следующая информация может быть изменена
i. Настройки краулера для запуска;
ii. Аргументы для запуска;

5. **Запуски краулеров**
Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

(a) Просмотр списка запусков в определенном статусе (`Pending`, `Running` или `Finished`) с пагинацией, совершенных в проектах, к которым у пользователя есть как минимум `ReadOnly` доступ;

(b) Редактирование запуска - остановка запуска, перевод его в состояние `Finished`. Операция может быть применена только к запускам в состоянии `Running` или `Pending`;

(c) Удаление запуска - удаление всех данных о запуске из базы данных. Операция может быть применена только к запускам в состоянии `Finished`;

(d) Создание запуска со следующей информацией

i. Краулер, с которым происходит запуск;
ii. Настройки запуска – это могут быть как и предопределенные настроки на `scrapyd` [1], так и собственные настройки;
iii. Аргументы запуска – аргументы для запуска краулера, которые передаются через командную строку;
iv. Описание запуска;

6. **Периодические запуски**
Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

(a) Просмотр списка периодических запусков с пагинацией;
(b) Редактирование следующей информации о периодическом запуске
i. Настройки будущих запусков – это могут быть как и предопределенные настроки на `scrapyd`, так и собственные настройки;
ii. Аргументы будущих запусков – аргументы для запуска краулера, которые передаются через командную строку;
iii. Краулер, с помощью которого будет совершен запуск;
iv. cron-expression расписания запуска;
(c) Удаление периодического запуска;
(d) Отмена последующих запусков - перевод периодической задачи в состояние `Disabled`;
(e) Возобновление запусков - перевод периодической задачи в состояние `Enabled`;
(f) Создание периодического запуска со следующими данными
i. Название;
ii. Описание – опциональное;
iii. Краулер;
iv. Приоритетность, влияющая на очередь запусков (`Low`, `Normal` или `High`);
v. Статус (`Enabled` или `Disabled`);
vi. Настройки будущих запусков – это могут быть как и предопределенные настроки на `scrapyd`, так и собственные настройки;
vii. Аргументы будущих запусков – аргументы для запуска краулера, которые передаются через командную строку;

[1] `http://doc.scrapy.org/en/latest/topics/settings.html`

2 Условия выполнения программы

2.1 Минимальный состав аппаратных средств

Минимальный состав технических конмпонент, необходимых для нормального функционирования программы:

1. Компьютер оснащенный процессором Intel Core i5 с тактовой частотой 2,3 ГГц;
2. 16 Гб ОЗУ;
3. Жесткий диск с объемом свободной памяти более чем 50 ГБ;
4. Клавиатура и мышь;
5. Доступ в интернет.

2.2 Минимальный состав программных средств

Для нормального функционирования программы требуется компьютер, оснащенный следующими программными компонентами:

1. macOS 10.15.2;
2. scrapyd [1];
3. Scala 2.12.6;
4. Play-framework 2.6.13;
5. PostgreSQL 11 [5];

2.3 Требования к персоналу

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 1 штатной единицы со следующими навыками:

1. Базовые навыки администрирования Unix [6] систем;
2. Базовые навыки администрирования базы данных PostgreSQL [5];
3. Базовые навыки работы с sbt [7].

3 Выполнение программы

3.1 Подготовка проекта

В комплект поставки программы входит техническая документация, приложение (исполняемые файлы, примеры запросов и прочие необходимые для работы программы файлы) и презентацию проекта.

3.1.1 Запуск scrapyd

В директории проекта (или в любой другой директории на компьютере) надо создать пустую папку с произвольным названием.

В пустой папке надо запустить предустановленный сервер командой scrapyd (см. листинг 1).

```
> mkdir scrapyd_server
> cd scrapyd_server
> ls

> scrapyd
```

Листинг 1 — Запуск scrapyd

3.1.2 Создание таблиц

Для корректной работы сервера необходимо запустить и проинициализировать базу данных PostgreSQL [5].

Для этого в директории проекта необходимо запустить следующую комманду из листинга ниже 2:

```
> sbt "runMain models.common.DBCreator"
```

Листинг 2 — Создание и инициализация таблиц

3.2 Запуск программы

Для запуска сервера необходимо выполнить команду, находясь в директории проекта, из листинга 3.

```
> sbt run
```

Листинг 3 — Запуск сервера

ПРИЛОЖЕНИЕ А

Используемые понятия и определения

Web scraping – это сбор данных с различных интернет-ресурсов. Общий принцип его работы можно объяснить следующим образом: некий автоматизированный код выполняет GET-запросы на целевой сайт и получая ответ, парсит HTML-документ, ищет данные и преобразует их в заданный формат.

Проект – сущность для объединения и предоставления доступа к запускам/краулерам/периодическим задачам.

Веб краулер – программа, являющаяся составной частью поисковой системы и предназначенная для перебора страниц Интернета с целью занесения информации о них в базу данных поисковика. Неотъемлемая часть проекта. Именно с помощью пауков пользователь может “краулить” сайты для сбора необходимой информации.

Запуск – единоразовый запуск краулера с настройками и аргументами, указанными для этого запуска.

Периодический запуск – запуск с множеством настроек, повторяющийся в определенные периоды времени (запуски по cron-expression).

Список источников

[1] Github scrapyd/scrapyd [Электронный ресурс] URL: `https://github.com/scrapy/scrapyd` (Дата обращения: 16.04.2020, режим доступа: свободный)

[2] Единая система программной документации – М.: ИПК, Издательство стандартов, 2000, 125 стр.

[3] ScalaTest+Play [Электронный ресурс] URL:`http://www.scalatest.org/plus`(Дата обращения: 16.04.2020, режим доступа: свободный)

[4] Testing - silhouette [Электронный ресурс] URL:`https://www.silhouette.rocks/docs/testing` (Дата обращения: 16.04.2020, режим доступа: свободный)

[5] Postgresql [Электронный ресурс] URL:`https://www.postgresql.org` (Дата обращения: 16.04.2020, режим доступа: свободный)

[6] Unix [Электронный ресурс] URL: `https://en.wikipedia.org/wiki/Unix` (дата обращения: 10.10.2018).

[7] SBT [Электронный ресурс] URL: https://www.scala-sbt.org (Дата обращения: 16.04.2020, режим доступа: свободный)

Лист регистрации изменений									
Изм.	Номера листов				Всего листов в документе	№ документа	Входящий № сопроводит. докум. и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					