

Abstract geometric lines forming various polygons and shapes, primarily in the upper left quadrant of the page.

ПРОЕКТ «ОБЛАЧНОЕ ФАЙЛОВОЕ ХРАНИЛИЩЕ»

Иван Мальцев



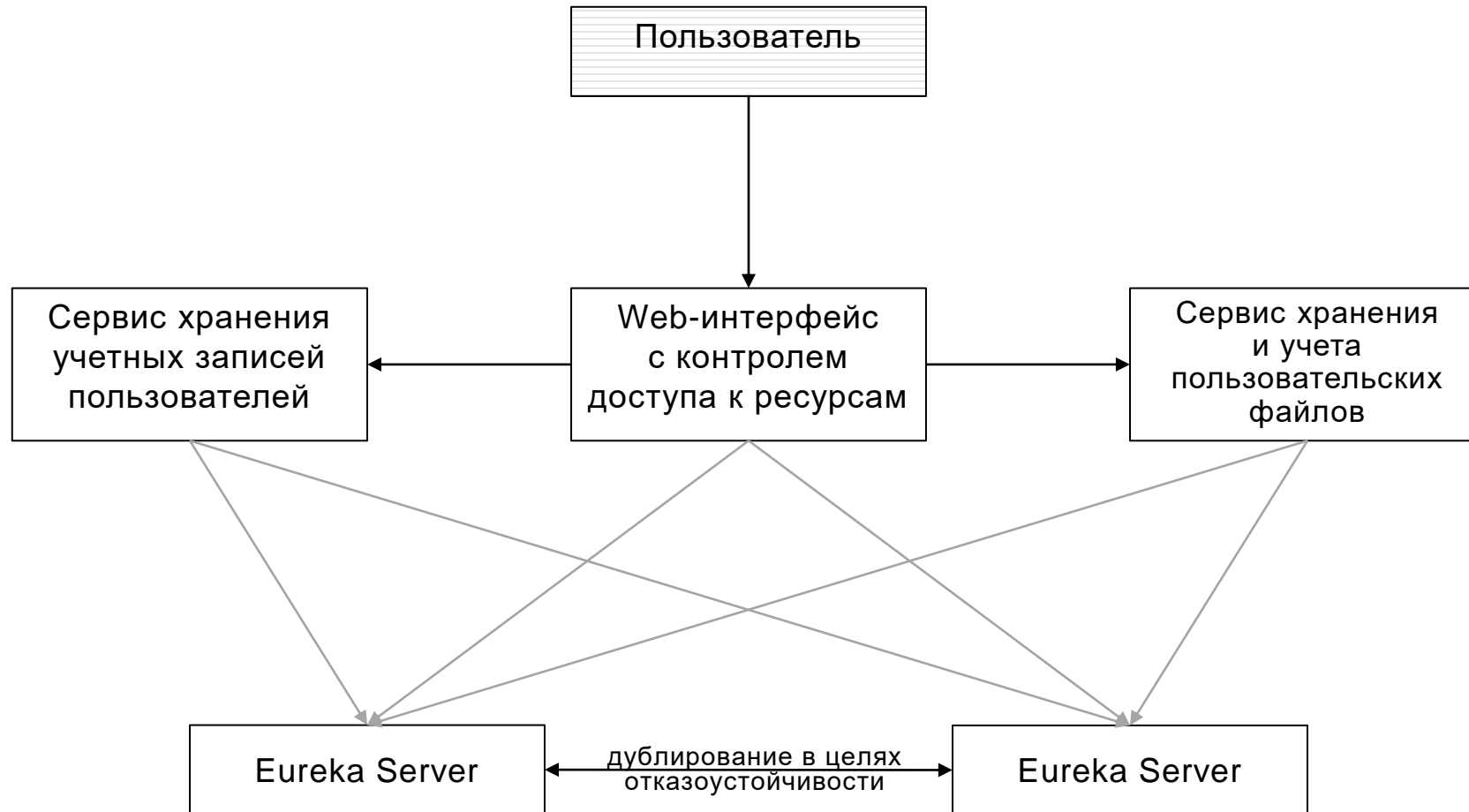
ОБЗОР ПРОЕКТА

Объектом разработки является многопользовательское облачное файловое хранилище.

Цель работы: разработать файловый сервер с функционалом хранения личных файлов пользователей на удаленном сервере, с возможностью доступа к ним (скачивания и загрузки новых) через Интернет. Доступ пользователей и разграничение хранимых на сервере файлов должен осуществляться с помощью учетных записей, аутентифицируемых посредством пары логин-пароль. Также, с целью расширения пользовательской аудитории, следует реализовать возможность создания новых пользователей.

Разработка проведена на языке Java с использованием фреймворка Spring Boot. С целью увеличения отказоустойчивости, а также для упрощения дальнейшего развития функционала в разработке применена микросервисная архитектура под управлением сервера обнаружения Eureka.

СХЕМА СЕРВИСОВ ПРОЕКТА



СЕРВИС ХРАНЕНИЯ ФАЙЛОВ

Хранит файлы во внутреннем хранилище, предварительно переименовав их в собственном формате именования. Информация об оригинальном имени файла, его размере, дате создания и идентификаторе владельца хранится в собственной базе данных. По запросу от web-интерфейса сервис отдает файл для скачивания (снабдив сведениями об исходном имени), отдает информацию о файле с целью её редактирования, а также принимает новый файл для размещения в хранилище.

```
@RestController
@RequestMapping("/api/files")
@RequiredArgsConstructor
public class FileEntityController {

    private final FileEntityService fileEntityService;

    @GetMapping
    public FileEntityPageDto getPageOfFilesList(@RequestParam(required = false) Long
ownerId, @PageableDefault(value = 15) Pageable pageable) {
        return fileEntityService.getPageWithFoundByOwnerId(ownerId, pageable);
    }

    @PostMapping
    @ResponseStatus(HttpStatus.CREATED)
    public FileEntityDto newFile(@RequestParam Long ownerId, @RequestPart MultipartFile
newFile) {
        System.out.println(ownerId);
        return fileEntityService.saveNewFile(newFile, ownerId);
    }

    @PutMapping
    public FileEntityDto editFile(@RequestBody FileEntityDto updatedEntity) {
        return fileEntityService.editFileEntity(updatedEntity);
    }

    @GetMapping("/{id}")
    public FileEntityDto getFileEntity(@PathVariable BigInteger id) {
        return fileEntityService.getFileEntityById(id);
    }

    @GetMapping("/download/{id}")
    public ResponseEntity<Resource> downloadFile(@PathVariable BigInteger id) {
        FileEntityDto fileEntity = fileEntityService.getFileEntityById(id);
        Resource resource = fileEntityService.getFileById(id);
        try {
            String encodedName = URLEncoder.encode(fileEntity.getName(),
                StandardCharsets.UTF_8.toString()).replace("+", "%20");
            return ResponseEntity.ok()
                .header(HttpHeaders.CONTENT_DISPOSITION,
                    "attachment; filename=\"" + encodedName + "\"")
                .body(resource);
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
            return ResponseEntity.notFound().build();
        }
    }

    @DeleteMapping("/{id}")
    public void deleteFile(@PathVariable BigInteger id) {
        fileEntityService.deleteFileAndEntityById(id);
    }
}
```

СЕРВИС ХРАНЕНИЯ УЧЕТНЫХ ЗАПИСЕЙ

Хранит в базе данных сведения о зарегистрированных пользователях, их логины, пароли (зашифрованные с помощью хеш-функции Bcrypt), а также назначенные им роли (пользователь, администратор). По запросу от web-интерфейса сервис отдает сведения о пользователе с целью использования их при определении прав доступа к ресурсам, а также принимает сведения о новом пользователе при регистрации.

```
@RestController
@RequiredArgsConstructor
@RequestMapping("/api/accounts")
public class AccountController {

    private final AccountService accountService;

    @GetMapping("/{login}")
    public Optional<AccountDto> getAccount(@PathVariable(name = "login") String login) {
        return accountService.getByLogin(login);
    }

    @PostMapping
    public AccountDto createAccount(@RequestBody AccountDto newAccount) {
        return accountService.saveNew(newAccount);
    }

    @PutMapping
    public AccountDto editAccount(@RequestBody AccountDto account) {
        return accountService.edit(account);
    }































    @DeleteMapping("/{id}")
    public void deleteAccount(@PathVariable(name = "id") Long id) {
        accountService.deleteById(id);
    }
}
```


WEB-ИНТЕРФЕЙС

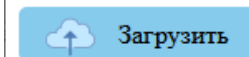
Реализует взаимодействие пользователя с проектом, осуществляет контроль доступа к ресурсам, предоставляет пользовательский интерфейс для доступа к файлам. Интерфейс реализован в виде таблицы со строками, представляющими собой имена файлов и возможные действия с ними. Также предусмотрено переключение языков русский/английский, возможность переключения пользователей, настраивается количество выведенных файлов на странице.


Ваши файлы:

Текущая страница: 1 из 1

Имя файла	Размер, kB	Дата загрузки	Действия
D_E_2021-10-11_Accident.mp4	790429	2022-05-17 00:15	  
Java Intro.pdf	2975	2022-05-17 01:56	  
Java Lesson 3.pdf	1233	2022-05-17 01:56	  
Java Lesson 4.pdf	1120	2022-05-17 01:56	  
Java Lesson 5_compressed.pdf	674	2022-05-17 01:56	  
Java Lesson 6_updated.pdf	1178	2022-05-17 01:57	  
Java Lesson 7.pdf	1749	2022-05-17 01:57	  
output.txt	7	2022-05-17 01:45	  
sdh.txt	2	2022-05-17 00:16	  
Скриншот сообщения.docx	991	2022-05-17 02:34	  

Показывать файлов на странице: 15 



admin 

[English](#)



ПЛАНИРУЕМАЯ ДОРАБОТКА ФУНКЦИОНАЛА

- хранение файлов в архивированном виде с целью уменьшения использования дискового пространства
- возможность загрузки на сервер и с сервера сразу нескольких файлов
- возможность делиться файлами с другими пользователями, а также предоставлять ссылку для скачивания неавторизованным пользователям
- улучшение взаимодействия между микросервисами с более подробной обработкой возникающих ошибок в целях объяснения пользователю, что же именно пошло не так



СПАСИБО ЗА ВНИМАНИЕ!

Иван Мальцев

ivmcorp@yandex.ru