

ДОДАТОК А

Перелік джерел посилання за науковими напрямками керівника та науковців
кафедри програмної інженерії

2. Pronina, D., Kyrychenko, I. Comparison of Redux and React Hooks Methods in Terms of Performance CEUR Workshop Proceedings, 2022, 3171, pp. 791–800
6. Sakharov, I., Smelyakov, K., Bohomolov, O. Research and Development of Information Technology for Determining Shoe Size by Image2023 IEEE Open Conference of Electrical, Electronic and Information Sciences, eStream 2023 - Proceedings, 2023
8. Gruzdo, I., Kyrychenko, I., Tereshchenko, G., Shanidze, N. Metrics applicable for evaluating software at the design stage // 5th International Conference on Computational Linguistics and In-telligent Systems (COLINS-2021), Kharkiv, Ukraine, April 22-23, 2021. – CEUR Workshop Proceedings, 2021, 2870, Volume I, pp. 916-936

ДОДАТОК Б

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016324318

Дата перевірки:
05.06.2024 17:11:07 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
05.06.2024 17:15:26 EEST

ID користувача:
100012353

Назва документа: 2024_M_ПІ_ІПЗм-22-4_Кондратьєв_М_А

Кількість сторінок: 35 Кількість слів: 6082 Кількість символів: 45384 Розмір файлу: 533.49 KB ID файлу: 1016123007

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

2.8%
Схожість

Найбільша схожість: 0.82% з джерелом з Бібліотеки (ID файлу: 1016108283)

1.15% Джерела з Інтернету

74

Сторінка 37

2.2% Джерела з Бібліотеки

40

Сторінка 37

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

4

Підозріле форматування

6
сторінок

ДОДАТОК В

Слайди презентації

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Дослідження методів обміну даними між платформою Android та серверною частиною

Керівник: доц. Голян Віра Володимирівна
Виконав: ст. гр. ІПЗм-22-4 Кондратьєв Микита Андрійович

1

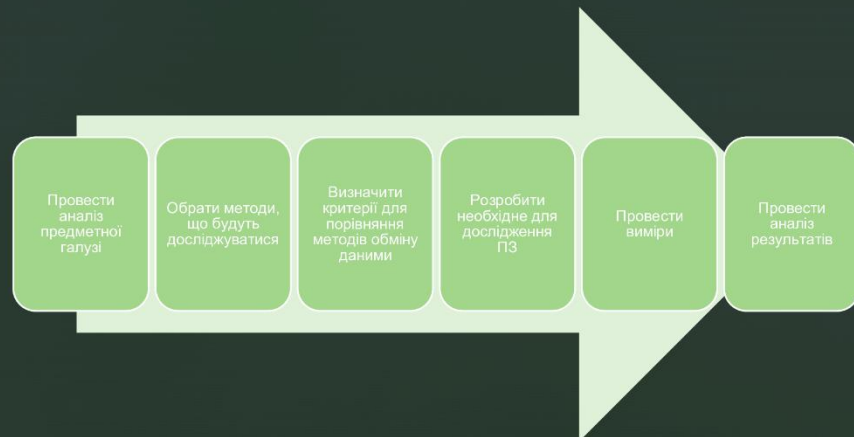
Мета роботи

Об'єктом дослідження є сучасні способи обміну даними між платформою Android та серверною частиною.

Мета роботи – аналіз та порівняння різних методів обміну даними між платформою Android та серверною частиною для підвищення ефективності процесу їх взаємодії.

2

Планування дослідження



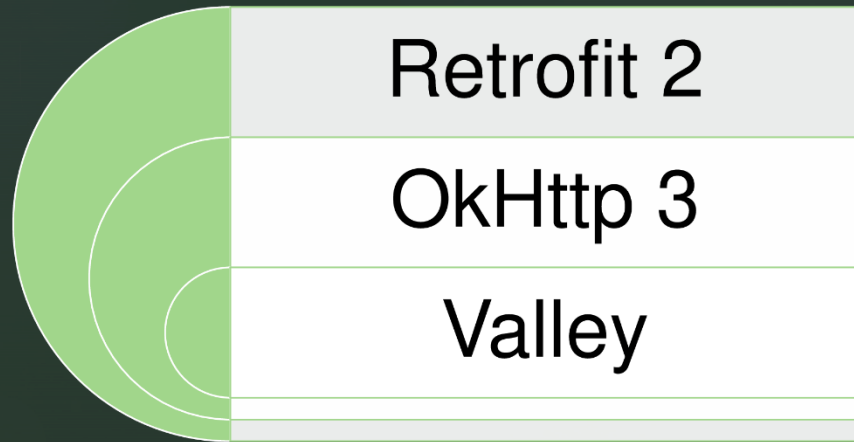
3

Аналіз предметної галузі

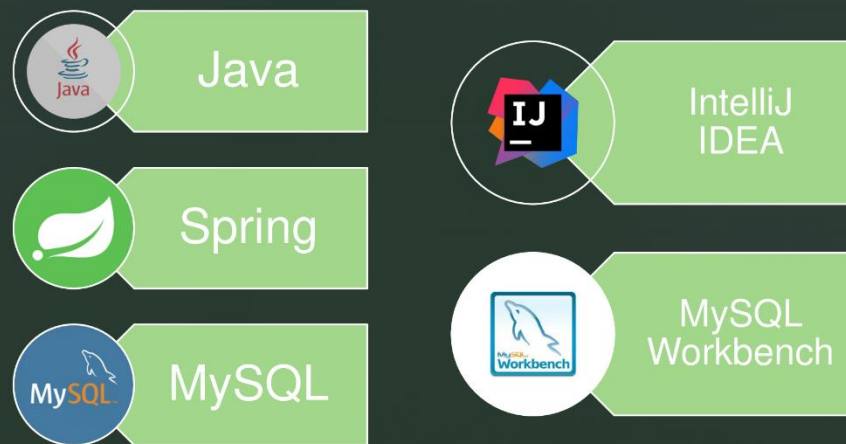
- На сьогоднішній день клієнт-серверна архітектура є одною з найпопулярніших архітектур для створення програмного забезпечення, а платформа Android однією з найпопулярніших платформ для користувачів
- Способів обміну даними між платформою Android та серверною частиною, та їх реалізацій існує досить багато
- Кожен з них має свої переваги та недоліки

4

- У дослідженні приймають участь:



- Програмна реалізація серверу



Програмна реалізація серверу

Приклад GET запиту

```
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class ExperimentGetController {  
  
    @GetMapping("/experiment/getTest")  
    public TestData getTestData() {  
        ExperimentData.setTimeOfGetRequestReceive();  
  
        TestData data = DataGenerator.generateTestData();  
  
        ExperimentData.setTimeOfGetResponseSent();  
        return data;  
    }  
}
```

7

Програмна реалізація Android-частини



Kotlin



Gson



Android Studio

8

Вибір Android API

Критерії загорткової моделі	Доступність для пристроїв					Корисність
	Зворотня сумісність	Можливості для розробки	Швидкодія роботи	Зручність розробки		
Коефіцієнти загорткової моделі	2/15	1/5	1/3	4/15	1/15	
API 16	1	0	0	0	0	0,133
API 21	0,94	0,25	0,33	0,33	0,17	0,385
API 23	0,71	0,5	0,67	0,67	0,5	0,63
API 26	0	1	1	1	1	0,867

9

Реалізація GET запиту для Retrofit

```
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory

object RetrofitClient {
    private const val BASE_URL = ""

    val instance: Retrofit by lazy {
        Retrofit.Builder()
            .baseUrl(BASE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .build()
    }
}
```

```
experimenData.collectDataBeforeStart()

val apiService = RetrofitClient.instance.create(ApiService::class.java)
apiService.getTestData().enqueue(object : Callback<TestData> {
    override fun onResponse(call: Call<TestData>, response: Response<TestData>) {
        if (response.isSuccessful) {
            testData = response.body()
            experimenData.collectDataIfSuccess()
        } else {
            experimenData.collectDataIfFail()
        }
    }

    override fun onFailure(call: Call<TestData>, t: Throwable) {
        experimenData.collectDataIfFail()
    }
})
```

10

Реалізація GET запиту для OkHttp

```

experimenData.collectDataBeforeStart()

val client = OkHttpClientInstance.client
val request = Request.Builder()
    .url("")
    .build()

client.newCall(request).enqueue(object : Callback {
    override fun onFailure(call: Call, e: IOException) {
        experimenData.collectDataIfFail()
    }

    override fun onResponse(call: Call, response: Response) {
        if (response.isSuccessful) {
            response.body?.string()?.let { responseBody ->
                testData = Gson().fromJson(responseBody, TestData::class.java)
                runOnUiThread {
                    experimenData.collectDataIfSuccess()
                }
            }
        } else {
            experimenData.collectDataIfFail()
        }
    }
})

```

```

import okhttp3.OkHttpClient
import okhttp3.Request

object OkHttpClientInstance {
    val client: OkHttpClient by lazy {
        OkHttpClient.Builder().build()
    }
}

```

11

Реалізація GET запиту для Volley

```

import android.content.Context
import com.android.volley.Request
import com.android.volley.RequestQueue
import com.android.volley.toolbox.Volley

class VolleySingleton constructor(context: Context) {
    companion object {
        @Volatile
        private var INSTANCE: VolleySingleton? = null
        fun getInstance(context: Context) =
            INSTANCE ?: synchronized(this) {
                INSTANCE ?: VolleySingleton(context).also {
                    INSTANCE = it
                }
            }
    }

    val requestQueue: RequestQueue by lazy {
        Volley.newRequestQueue(context.applicationContext)
    }

    fun <T> addToRequestQueue(req: Request<T>) {
        requestQueue.add(req)
    }
}

```

12

Реалізація GET запиту для Volley

```

experimenData.collectDataBeforeStart()
val url = ""

val stringRequest = StringRequest(
    Request.Method.GET, url,
    Response.Listener<String> { response ->
        testData = Gson().fromJson(response, TestData::class.java)
        experimenData.collectDataIfSuccess()
    },
    Response.ErrorListener { error ->
        experimenData.collectDataIfFail()
    })

VolleySingleton.getInstance(this).addToRequestQueue(stringRequest)

```

13

Результати вимірювання часу взаємодії між сервером та платформою Android

Бібліотеки	Час взаємодії між сервером та платформою Android (мс)	
	GET запит	POST запит
Retrofit	150	200
OkHttp	140	190
Valley	130	180

14

Результати вимірювання використання ресурсів оперативної пам'яті та процесору

Бібліотеки	Використання ресурсів	
	Використання пам'яті (МВ)	Використання CPU (%)
Retrofit	50	15
OkHttp	45	12
Valley	40	10

15

Результати вимірювання відсотку успішних запитів

Бібліотеки	Відсоток успішних запитів	
	GET запит	POST запит
Retrofit	98%	97%
OkHttp	97%	96%
Valley	99%	98%

16

Аналіз результатів

- Підводячи підсумок, Valley має більш кращі показники за свої аналоги, однак її налаштування займає більше часу, бо є більш складною задачею, у порівнянні з іншими. Тому вибір методу обміну даними між платформою Android та серверною частиною, залежить від цілей проекту.
- Якщо складність програмної реалізації проекту не так важлива як його якість, то вибір краще зробити у сторону Valley.
- Якщо ж ні, то краще буде обрати Retrofit, для більшої надійності обміну даними, або OkHttp, для більш швидкого відгуку та меншого навантаження.
- Однак, варто зазначити, що вказані переваги та недоліки будуть помітні, зазвичай, у великих проектах. У малих проектах, слід віддати перевагу, більш знайомому та простому методу.

17

Апробація

За результатами роботи були створені тези доповіді IX Міжнародна науково-технічна конференції «Поліграфічні, мультимедійні та web-технології» за темою «Оцінка ефективності методів обміну даними між платформою android та серверною частиною»

18

Висновки

У ході кваліфікаційної роботи було зібрано необхідні данні та підготовлено все необхідне для проведення магістерського дослідження сучасних способів обміну даними між платформою Android та серверною частиною.

Було розроблено технічне завдання для магістерського дослідження. Проведено збір теоретичного матеріалу, охоплюючи актуальні підходи та технології в області обміну даними.

Було представлено основні відомості про платформу Android. Надано відомості про структуру серверних систем, які пов'язані з обробкою та обміном даними.

Було представлено основні принципи та інформацію про здійснення обміну даними у контексті платформи Android. Обрано інструменти та технології для виконання дослідження та реалізовано серверну частину та три Android-застосунки.

Було створено та реалізовано план проведення магістерського дослідження. У якості метрик для порівняння методів обміну даними між платформою Android та серверною частиною були обрані відсоток успішних запитів, осяг використовуваної оперативної пам'яті та CPU, та час взаємодії між сервером та платформою Android.

Використовуючи результати проведених експериментів було сформувано рекомендації використання тих чи інших методів для обміну даними.

За результатами роботи було створено тези доповіді за темою «Оцінка ефективності методів обміну даними між платформою Android та серверною частиною», на IX Міжнародну науково-технічну конференцію «Поліграфічні, мультимедійні та веб-технології».

19

Дякую за
увагу!

20

ДОДАТОК Г

Апробація результатів роботи

ОЦІНКА ЕФЕКТИВНОСТІ МЕТОДІВ ОБМІНУ ДАНИМИ МІЖ ПЛАТФОРМОЮ ANDROID ТА СЕРВЕРНОЮ ЧАСТИНОЮ

Кондратьєв М. А., студент, кафедра ПІ ХНУРЕ

Голян В. В., доцент, кафедра ПІ ХНУРЕ

Актуальність та постановка проблеми. В умовах сучасного світу та розвитку технологій, мобільні додатки стають все більш популярними. Зі зростом популярності, мобільні застосунки стають все більш складними. За весь час сильно ускладнилася внутрішня та зовнішня архітектури застосунків, а клієнт-серверна взаємодія стала невід'ємною частиною реалізації. Зі зростом складності програмної реалізації програмних систем із мобільною та серверною частинами, зростає необхідність у правильній оцінці ефективності методів обміну даними між мобільною та серверною частинами. Особливо гостро дане питання стоїть для платформи Android, як для найбільш популярної мобільної платформи у наш час [1].

Основні матеріали дослідження. Процес обміну даними між платформою Android та серверною частиною програмного продукту складається з відправлення запиту із мобільного застосунку на сервер, обробки запиту на стороні серверу та отримання відповіді від серверу на стороні платформи Android [2]. Цей процес необхідний для отримання мобільним застосунком збережених даних від серверу, для відправлення нових даних на збереження, або для оновлення збережених даних. Розглянувши цей процес, можна визначити декілька груп метрик, які можуть показати ефективність методів обміну даними між платформою Android та серверною частиною.

Говорячи про процес обміну даними між застосунком та сервером, першою, та основною групою метрик слід вважати час витрачений на різних етапах процесу обміну даними.

Розглядаючи дану групу, головною метрикою, що може показати ефективність методів, можна побачити загальний час відгуку. Дана метрика вираховується як різниця в часі між початком відправки запиту на серверну частину та завершенням отримання відповіді на стороні платформи Android:

$$TRT = TDR - TSR, \quad (1)$$

де TRT – загальний час відгуку;

TDR – час завершення отримання відповіді;

TSR – час початку відправки запиту.

Однак, дана метрика має суттєвий недолік, який не дозволяє точно оцінити сам метод обміну даними. Даним недоліком є те, що між початком відправки запиту на серверну частину та завершенням отримання відповіді на стороні платформи Android, є період часу, який не залежить від обраного методу обміну, але залежить від складності запиту та ресурсів серверу, а саме час обробки запиту на сервері.

Саме тому, у якості головної метрики для аналізу ефективності слід

розглядати, не загальний час відгуку, а час взаємодії серверу та застосунку. Дана метрика включає в себе лише час відправки запиту із мобільного застосунку до серверу, та час передачі відповіді від серверу до платформи Android [3]. Час взаємодії серверу та застосунку можна визначити, як різницю між загальним часом відгуку та часом обробки запиту на сервері:

$$IT = TRT - SPT, \quad (2)$$

де IT – час взаємодії серверу та мобільного застосунку;

TRT – загальний час відгуку;

SPT – час обробки запиту на сервері.

Також, як метрики, що можуть допомогти при оцінці ефективності методів обміну даними, можна розглянути, об'єм використаних ресурсів, на стороні мобільного застосунку під час обміну даними, максимальний обсяг передачі даних, відсоток успішних запитів, та інші.

Після проведення розрахунків для різних розглядаємих методів, для більш зручної оцінки, можна порівняти та вирахувати виграші та втрати для кожних метрик. Це надасть метрикам більш зручний вигляд для оцінки.

Визначивши виграші та втрати кожної з обраних метрик, можна перейти до розрахунку ефективності методів. Загальна формула для розрахунку ефективності методів обміну даними буде виглядати, як сума добутків виграшів метрик та коефіцієнтів виграшних метрик поділена на суму добутків втрат метрик та коефіцієнтів метрик із втратами.

Для кожного окремого випадку, для якого порівнюються методи обміну даними, можна використати різні метрики. Також, в залежності від поставлених цілей та обмежень, можна змінити розрахунки виграшів та втрат для відповідних метрик. Коефіцієнти метрик виграшів та втрат, також будуть змінюватися відповідно до важливості цілей та обмежень проекту, для якого проводиться порівняння методів.

Висновки. Виявлено, що різні метрики, що приймають участь в оцінці ефективності методів обміну даними між серверною частиною та платформою Android, мають свої переваги та обмеження. Важливим аспектом є підбір відповідних до ситуації метрик для оцінки ефективності. Описано загальний вигляд формули для оцінки ефективності методів обміну даними між серверною частиною та платформою Android.

Список літератури

1. Franceschi H. J. Android App Development. Jones & Bartlett Learning, LLC, 2016. 682 с.
2. Jeong H., Ryoo H., Li K. J. INFACTORY: A RESTFUL API SERVER FOR EASILY CREATING INDOORGML. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2018. XLII-4/W8. С. 77–84. URL: <https://doi.org/10.5194/isprs-archives-xlii-4-w8-77-2018> (дата звернення: 12.12.2023).
3. Kumari S. REST based API. *International Journal of Trend in Scientific Research and Development*. 2017. Volume-1, Issue-4. С. 571–575. URL: <https://doi.org/10.31142/ijtsrd2200> (дата звернення: 12.12.2023).

ДОДАТОК Д

Експертний висновок результатів перевірки кваліфікаційної роботи на
відповідність оформлення вимогам ДСТУ 3008: 2015

Експертний висновок результатів перевірки кваліфікаційної роботи

студент
(посада)

програмної інженерії
(кафедра)

ПЗМ-22-4
(група)

Кондратьєв Микита Андрійович

(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні позначки	
	7.15 Додатки	

зауважень немає

Експерт

(підпис)

Олена ОЛІЙНИК

(прізвище, ініціали)

07.06.2024