

Конструирование программного обеспечения

Сервис уведомлений об изменениях в расписании

Линь Цзэдун 5130904/30106

Определение проблемы

Студенты и преподаватели часто проверяют сайт `ruz.spbstu.ru` вручную, чтобы узнать, были ли внесены изменения в расписание; на платформе отсутствуют функции «добавить в избранное/подписаться» и «уведомление об изменениях», из-за чего легко пропустить обновления. Мы стремимся предоставить сервис, позволяющий пользователям подписаться на расписание занятий/преподавателей/аудиторий, которые они следят, и получать мгновенные уведомления через веб-сайт/Telegram/электронную почту при возникновении изменений.

Выработка требований

1) Персоны

Студент (основной пользователь),
Преподаватель,
Администратор (аудит и метрики доставки)

2) Пользовательские истории (текстом)

US-1 Подписка студента и быстрые оповещения

- Как студент, когда в расписании моей группы меняются время/аудитория/преподаватель/отмена/добавление, я хочу получить сводное уведомление в течение 2 минут (Web/Telegram/Email) с различием «было → стало», чтобы оперативно скорректировать планы.
- Критерии приёмки: создание подписки по `groupId`; появление записи в `/events`; дедупликация/дебаунс ≤ 120 с; уведомление содержит предмет, время, аудиторию, преподавателя и `diff`.

US-2 Отметка “важных” занятий

- Как студент, когда просматриваю подписки, я хочу помечать отдельные занятия как важные, чтобы получать по ним приоритетные уведомления.
- Критерии приёмки: `important=true` в подписке; важные изменения не агрегируются; маркировка [Важное] в уведомлении.

US-3 Мониторинг преподавателя и синхронизация со студентами

- Как преподаватель, когда меняются мои занятия, я хочу сразу узнать и (опционально) разослать уведомление подписавшимся студентам, чтобы избежать недопонимания.
- Критерии приёмки: подписка по `teacherId`; в событии — затронутые группы; в админке — статистика доставки.

3) Оценка масштаба и сроков хранения

- DAU(Ежедневная активность): ~10k. Пиковая одновременная нагрузка 300–500 пользователей.
- Частота опроса: горячие субъекты каждые 30–60 с, холодные — 5–10 мин; ночная сверка.
- Объём уведомлений: до 50k–100k/день в сезон пиков.
- Хранение: события и аудит — ≥ 5 лет; сырые снапшоты — 30–90 дней; подписки/настройки — бессрочно до удаления.

4) Нефункциональные цели

- Концевая задержка детекции ≤ 2 мин;
- P95 чтения (кеш) ≤ 300 мс;
- Доступность 99.5%+/мес;
- Аудит и переотправка для снижения ложных срабатываний.

Разработка архитектуры и детальное проектирование

1) Характер нагрузки на сервис

- Соотношение R/W нагрузки

R/W \approx 20:1 (чтение: /events, /subscriptions; запись: создание/удаление подписок, фиксация событий и аудита).

- Объёмы трафика (API и внешние вызовы)

Пользовательский API: 200–400 RPS в среднем, пик 1–2k RPS (при попадании в кеш).

Одновременные пользователи: ~300–500 при ~10k DAU.

Опрос внешнего источника RUZ:

«Горячие» субъекты (часто читаемые) — каждые 30–60 с

«Холодные» — каждые 5–10 мин

Ночная сверка — фоновая, с пониженными лимитами

Уведомления: **50k–100k/день** в пиковые периоды (начало семестра/сессия), агрегируются дебаунсом.

- Объёмы дисковой системы

“Горячие” данные (30 дней): Postgres 5–10 GB.

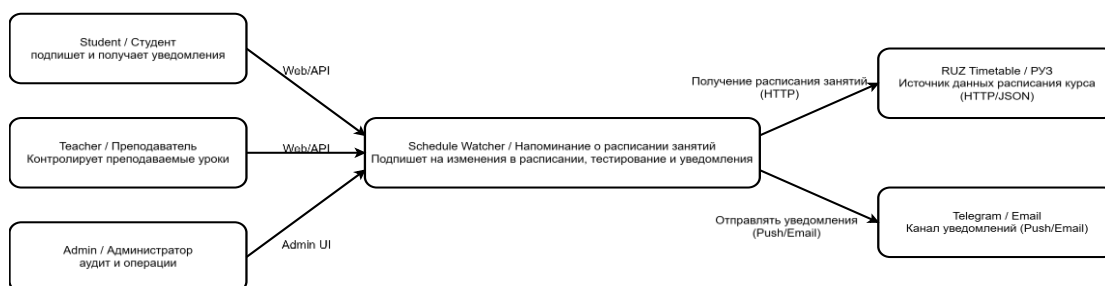
События + аудит: **0.5–1 GB/день** (зависит от числа подписок и «шумности» RUZ).

Сырые снапшоты RUZ: храним **30–90 дней** (старше — архив/удаление).

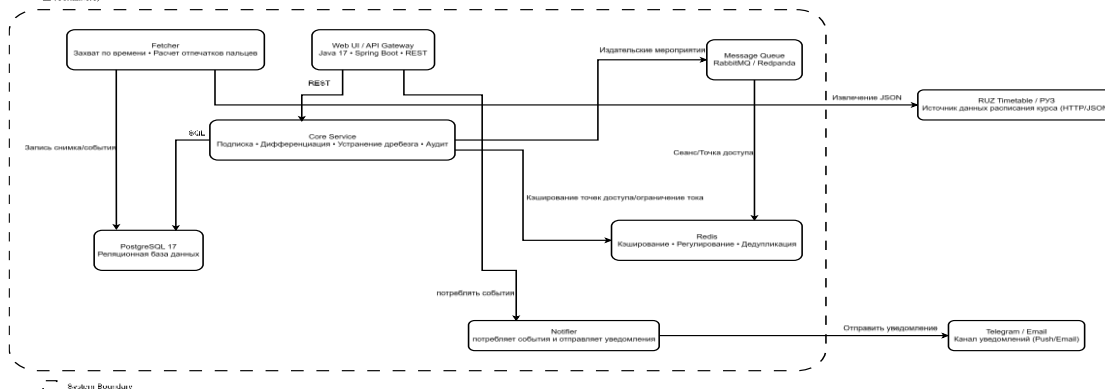
Долгосрочное хранение событий/аудита: **≥ 5 лет** (требования курса).

2)

L1 (System Context)



L2 (Событийный)



- L1 (System Context) — показаны акторы (Студент, Преподаватель, Администратор), наш сервис “Schedule Watcher” и внешние системы (RUZ, Telegram/Email) с направлениями взаимодействий (HTTP/JSON, Push/Email).
 - L2 (Container) — контейнеры внутри системы: Web/API, Core Service, Fetcher, Notifier, PostgreSQL, Redis, Message Queue, а также связи с RUZ и каналами уведомлений.
- 3) Контракты API + Ожидаемые нефункциональные требования на время отклика
- Базовые эндпойнты (v1):

- GET /ping → {"status":"pong"} (P99 ≤ **50 мс**)
- POST /subscriptions — создать подписку:
Тело: { "type":"group|teacher|room", "id":"<ruz-id>", "channels": ["web","telegram","email"], "filters":{"..."} } → 201 + {"subscriptionId":"..."}
- GET /subscriptions?cursor&limit — пагинация списка
- DELETE /subscriptions/{id} → 204 No Content
- GET /events?since=<ts>&limit=<n> — события-изменения (курс, время/аудитория/преподаватель, diff «было→стало»)
- GET /audit?subjectId=<id> (админ) — снапшоты, payload_hash, структурный diff, окно дебаунса, логи доставки
- Ошибки: единый формат { traceId, code, message, details }

НФ-требования (SLO/SLI):

- Чтение (кеш-хит) P95 ≤ 300 мс
- E2E-детекция изменения (fetch → diff → debounce → send) ≤ 2 мин
- Доступность ≥ 99.5% в месяц
- Ограничение частоты на RUZ + экспоненциальные ретрай; идемпотентность операций записи

4) Схему базы данных + почему она выдержит нефункциональные требования

Основные таблицы:

- users(id UUID PK, email UNIQUE, telegram_id, created_at, status)
- subjects(id PK, type ENUM, name, ruz_key UNIQUE)
- subscriptions(id UUID PK, user_id FK, subject_id FK, channels JSONB, filters JSONB, created_at, UNIQUE(user_id,subject_id))
- events(id BIGSERIAL PK, subject_id FK, event_time, diff JSONB, hash, created_at, UNIQUE(subject_id,hash))
- snapshots(id BIGSERIAL PK, subject_id FK, snapshot_at, payload_hash, raw JSONB)
- audit_logs(id BIGSERIAL PK, action, actor, target_id, meta JSONB, created_at)

Индексы и приёмы производительности:

- Покрывающий индекс events(subject_id, event_time DESC) — быстрые «по времени» выборки.
- Уникальность по user_id+subject_id и subject_id+hash — идемпотентность и защита от дубликатов.
- Партиционирование events по времени (неделя/месяц) — дешёвые архив/очистка.
- JSONB-пути на «горячих» полях diff (например, аудитория/время) — фильтрация без полного скана.
- Разделение нагрузки: ведущая БД на запись + один/несколько read-реплик для /events и /subscriptions.
- Redis — кеш «горячих» результатов и ограничение частоты вызовов.

Почему выполняются НФ-цели:

- Быстрые чтения → покрывающие индексы + реплики + кеш.
- Снижение записи → дедупликация событий по hash, дебаунс и пакетная обработка.
- Предсказуемое время ответа → короткие транзакции, ограниченные JSONB-выборки, партии.

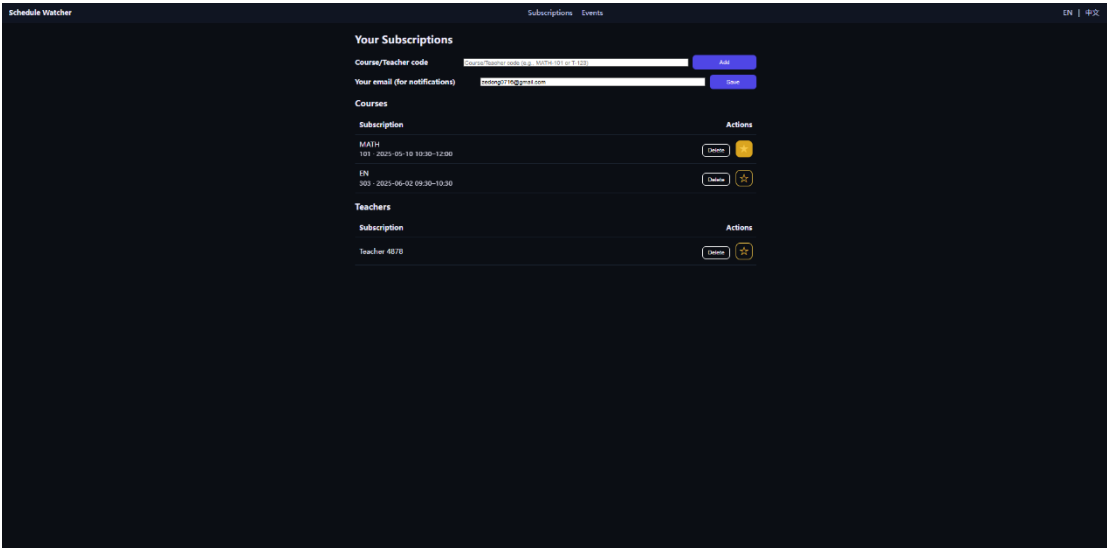
5) Схему масштабирования сервиса при росте нагрузки в 10 раз

- Горизонтальное масштабирование Web/API, Fetcher, Notifier (K8s HPA по RPS/длине очереди/CPU).
- Чтение с реплик в Postgres; автомасштаб пулов соединений; разнос read/write.
- Очереди: партиционирование по subjectId, несколько потребителей в группе; DLQ для «тяжёлых» сообщений.
- Кеширование: Redis TTL 5–15 мин для горячих ключей; кеш-штамп и rate-limit на RUZ.
- Деградация: при недоступности RUZ — показ «последнего валидного снапшота» с явной меткой времени.
- Управление данными: автоархив партиций events, TTL для snapshots, регулярные VACUUM/REINDEX.
- Наблюдаемость: Prometheus/Grafana (RPS, P95, задержка fetch, глубина очереди, delivery-rate), трассировка OTel.
- Безопасные развёртывания: канареечные релизы/rollback, миграции схемы через версии.

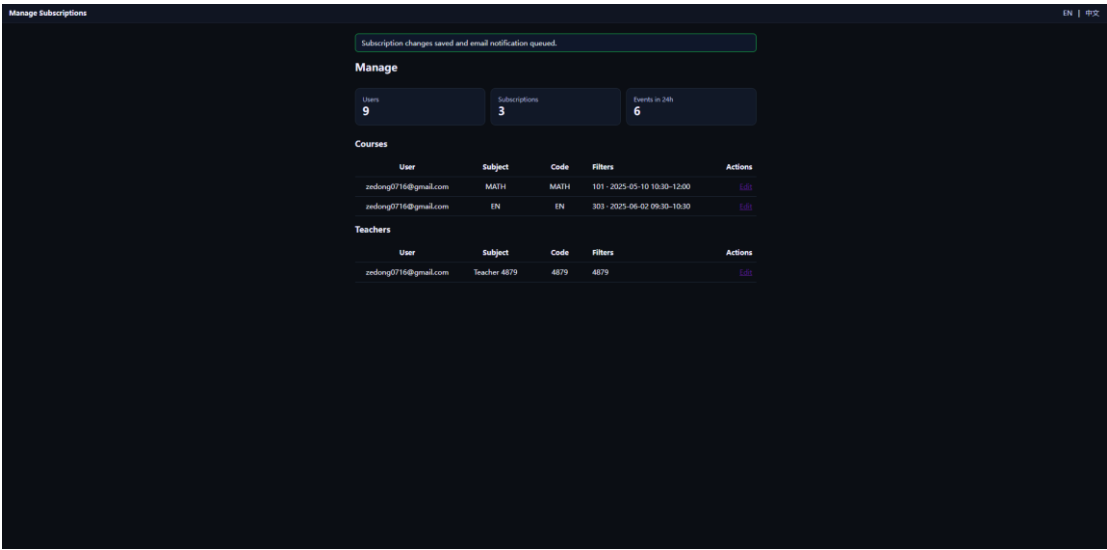
Кодирование и отладка

Конкретный программный код находится в папке Course-project (<https://github.com/Lin-zedong/Course-project/tree/main/Course-project>)

1. Schedule Watcher



2. Manage Subscriptions



3. Mails

Maillog

Connected

Inbox (3)

Delete all messages

Jim

Jim is a chess monkey.
Find out more at CERN.ch

Enrolle Jim

Search

1/3

1/3 of 3

<

>

[getSender(message)] moring0716@gmail.com	[Important] [Schedule Change] MATH	a few seconds ago	387 B
[getSender(message)] moring0716@gmail.com	[Schedule Change] Teacher 4879	2 minutes ago	313 B
[getSender(message)] moring0716@gmail.com	[Schedule Change] Teacher 4879	2 minutes ago	313 B

Maillog

Connected

Inbox (3)

Delete all messages

Jim

Jim is a chess monkey.
Find out more at CERN.ch

Enrolle Jim

Search

←

↻

⤴

⤵

From: real@buck0716137

Subject: [Important] [Schedule Change] MATH

To: zedong0716@gmail.com

Plain text

Source

Schedule changed for 10/28

times: 10:1 → 10:3

Show headers

Maillog

Connected

Inbox (3)

Delete all messages

Jim

Jim is a chess monkey.
Find out more at CERN.ch

Enrolle Jim

Search

←

↻

⤴

⤵

From: real@buck0716137

Subject: [Schedule Change] Teacher 4879

To: zedong0716@gmail.com

Plain text

Source

Schedule changed for Teacher 4879

teacher: 4879 → 4879

Show headers

Unit тестирование

```

Run // schedule-watcher [test]
D:\java\bin\java.exe -Dmaven.multiModuleProjectDirectory=D:\course-project -Djansi.passthrough=true -Dmaven.home=D:\IntelliJ IDEA 2024.1.4\plugins\maven\lib\maven3" *-Dclassworlds.conf=D:\IntelliJ IDEA 2024.1.4\plugins\m
[INFO] Scanning for projects...
[INFO] ----- com.example:schedule-watcher >-----
[INFO] Building Schedule Watcher 1.0.0
[INFO] from pom.xml
[INFO] ----- [ jar ]-----
[INFO]
[INFO] --- jacoco:0.8.11:prepare-agent @ schedule-watcher ---
[INFO] argLine set to: -javaagent:C:\Users\ADMINISRE\AppData\Local\Temp\org\jacoco\org.jacoco.agent\0.8.11\org.jacoco.agent-0.8.11-runtime.jar=destfile=D:\Course-project\target\jacoco.exec,excludes=**/ScheduleWatch
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ schedule-watcher ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 13 resources from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ schedule-watcher ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ schedule-watcher ---
[INFO] skip non existing resourceDirectory D:\Course-project\src\test\resources
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ schedule-watcher ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 11 source files with javac [debug parameters release 17] to target\test-classes
[INFO] 由于在类路径中发现了多个或多个处理器程序，因此启用了
批注处理。考虑运行新的 javac 可能由禁用批注处理。
所幸至少该名称指定了一个处理器程序。(-processor)
请指定处理器链 (-processor-path, --processor-module-path)。
或显式禁用批注处理 (-proc:none, -proc:full)。
可使用 -Xlint:options 隐藏此消息。
可使用 -proc:none 禁用批注处理。
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ schedule-watcher ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit4:junit4-platform-provider
[INFO]

```

```
Run // schedule-watcher [test]
[INFO] .....
[INFO] T E S T S
[INFO] .....
[INFO] Running com.example.schedulewatcher.controller.AdminControllerTest
WARNING: A Java agent has been loaded dynamically (C:\Users\ADMIN\workspace\repository\net\bytebuddy\byte-buddy-agent\1.14.19\byte-buddy-agent-1.14.19.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.288 s -- in com.example.schedulewatcher.controller.AdminControllerTest
[INFO] Running com.example.schedulewatcher.controller.AdminSubscriptionControllerManualDiffTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.033 s -- in com.example.schedulewatcher.controller.AdminSubscriptionControllerManualDiffTest
[INFO] Running com.example.schedulewatcher.controller.EventControllerTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.116 s -- in com.example.schedulewatcher.controller.EventControllerTest
[INFO] Running com.example.schedulewatcher.controller.SubscriptionControllerTest
[INFO] Tests run: 10, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.104 s -- in com.example.schedulewatcher.controller.SubscriptionControllerTest
[INFO] Running com.example.schedulewatcher.controller.WebControllerTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.011 s -- in com.example.schedulewatcher.controller.WebControllerTest
[INFO] Running com.example.schedulewatcher.service.NotificationServiceTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.178 s -- in com.example.schedulewatcher.service.NotificationServiceTest
[INFO] Running com.example.schedulewatcher.service.PushTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.366 s -- in com.example.schedulewatcher.service.PushTest
[INFO] Running com.example.schedulewatcher.service.ScheduleFetchServiceTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.114 s -- in com.example.schedulewatcher.service.ScheduleFetchServiceTest
[INFO] Running com.example.schedulewatcher.service.SubscriptionServiceTest
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.033 s -- in com.example.schedulewatcher.service.SubscriptionServiceTest
[INFO] Running com.example.schedulewatcher.service.UserSessionServiceTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.010 s -- in com.example.schedulewatcher.service.UserSessionServiceTest
[INFO] Running com.example.schedulewatcher.util.ScheduleDiffUtilTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.017 s -- in com.example.schedulewatcher.util.ScheduleDiffUtilTest
[INFO] .....
[INFO] Results:
[INFO] .....
[INFO] Tests run: 43, Failures: 0, Errors: 0, Skipped: 0
```

```

Run // schedule-watcher [test]
[INFO] Running com.example.schedulewatcher.controller.AdminSubscriptionControllerManualDiffTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.063 s -- in com.example.schedulewatcher.controller.AdminSubscriptionControllerManualDiffTest
[INFO] Running com.example.schedulewatcher.controller.EventControllerTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.114 s -- in com.example.schedulewatcher.controller.EventControllerTest
[INFO] Running com.example.schedulewatcher.controller.SubscriptionControllerTest
[INFO] Tests run: 16, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.104 s -- in com.example.schedulewatcher.controller.SubscriptionControllerTest
[INFO] Running com.example.schedulewatcher.controller.MeasControllerTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.011 s -- in com.example.schedulewatcher.controller.MeasControllerTest
[INFO] Running com.example.schedulewatcher.service.NotificationServiceTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.178 s -- in com.example.schedulewatcher.service.NotificationServiceTest
[INFO] Running com.example.schedulewatcher.service.BuzzClientTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.366 s -- in com.example.schedulewatcher.service.BuzzClientTest
[INFO] Running com.example.schedulewatcher.service.ScheduleFetchServiceTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.114 s -- in com.example.schedulewatcher.service.ScheduleFetchServiceTest
[INFO] Running com.example.schedulewatcher.service.SubscriptionServiceTest
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.633 s -- in com.example.schedulewatcher.service.SubscriptionServiceTest
[INFO] Running com.example.schedulewatcher.service.UserSessionServiceTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.010 s -- in com.example.schedulewatcher.service.UserSessionServiceTest
[INFO] Running com.example.schedulewatcher.util.ScheduleDiffUtilTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.017 s -- in com.example.schedulewatcher.util.ScheduleDiffUtilTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 43, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jacoco:0.8.12:report (report) @ schedule-watcher ---
[INFO] Loading execution data file D:\Course-projects\src\target\jacoco.exec
[INFO] Analyzed bundle "Schedule Watcher" with 12 classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO]
[INFO] Total time: 0.309 s
[INFO] Finished at: 2025-12-01T00:02:04+03:00
[INFO]
[INFO]
Process finished with exit code 0

```

Schedule Watcher

Sessions

Schedule Watcher

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cbty	Missed Lines	Missed Methods	Missed Clases				
<div><div></div><div>com.example.schedulewatcher.controller</div></div>	<div><div></div></div>	71%	<div><div></div></div>	41%	78	127	59	233	7	41	0	7
<div><div></div><div>com.example.schedulewatcher.service</div></div>	<div><div></div></div>	89%	<div><div></div></div>	59%	46	90	26	255	1	26	0	4
<div><div></div><div>com.example.schedulewatcher.util</div></div>	<div><div></div></div>	89%	<div><div></div></div>	75%	7	16	6	36	1	4	0	1
Total	518 of 2,692	80%	158 of 324	51%	131	233	101	524	9	71	0	12

Generated with: 360°/v0.0.0-1.1: 100/333/100/100

Интеграционное тестирование

Покрываемые пользовательские истории

В рамках интеграционного тестирования были покрыты две базовые пользовательские истории сервиса уведомлений об изменениях в расписании:

- US-1. Подписка студента и быстрые оповещения.
Студент подписывается на расписание своей группы и должен видеть список последних изменений по своим подпискам на странице /events.
- US-2. Отметка “важных” занятий.
Студент может пометить отдельные подписки как важные, чтобы выделять их и получать по ним приоритетные уведомления.

Тестовая среда

- Spring Boot 3.3 (web / MVC / Security / Data JPA).
- Встроенный контейнер сервера (RANDOM_PORT).
- База данных: **H2 in-memory** в режиме MODE=PostgreSQL, конфигурация в профиле test (application-test.yml): автоматическое создание/удаление схемы (ddl-auto=create-drop), отключена инициализация schema.sql, отключён планировщик (scheduling.enabled=false).
- JPA-репозитории UserRepository, SubjectRepository, SubscriptionRepository, EventRepository.
- Тестовый контекст Spring Boot (@SpringBootTest) + настройка MockMvc (@AutoConfigureMockMvc).
- Профиль запуска тестов: @ActiveProfiles("test").

Сценарий 1. Просмотр студентом только своих событий расписания

- Given:
 - В базе создан пользователь *A* с e-mail student@example.com.
 - Созданы три предмета: *Algorithms*, *Physics* и *Chemistry*.
 - Для пользователя *A* созданы две подписки: на *Algorithms* и на *Physics* (канал "web").
 - В таблицу events записаны два события изменения расписания:
 1. событие e1 для предмета *Algorithms*;
 2. событие e2 для предмета *Chemistry* (на который пользователь *A* **не** подписан).
- When:
 - В тесте эмулируется сессия пользователя *A* — в MockHttpSession кладётся его идентификатор (uid), как это делает UserServiceService.
 - Выполняется HTTP-запрос GET /events через MockMvc.
- Then:
 - Контроллер возвращает статус 200 ОК и шаблон представления events.
 - В модели присутствует атрибут events.
 - HTML-ответ содержит название предмета *Algorithms* и не содержит *Chemistry* — тем самым проверяется, что список событий фильтруется по подпискам текущего пользователя.

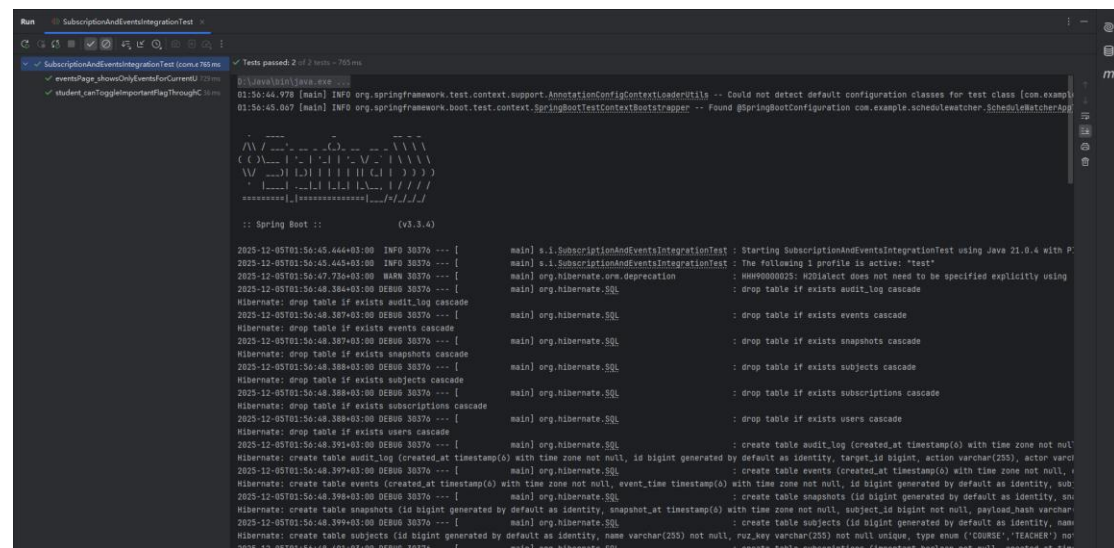
Сценарий 2. Пометка подписки как “важной”

- Given:
 - В базе создан пользователь *B* с e-mail student2@example.com.
 - Создан предмет Distributed Systems.
 - Для пользователя *B* создана подписка на этот предмет, у которой флаг important = false.
- When:
 - В тесте эмулируется сессия пользователя *B* (аналогично сценарию 1).
 - Через MockMvc выполняется запрос POST /subscriptions/{id}/important с идентификатором данной подписки.
- Then:
 - Контроллер возвращает перенаправление 302 на /subscriptions.
 - После перезагрузки подписки из БД значение поля important становится true.
 - Тем самым подтверждается корректная работа сценария изменения важности подписки (US-2).

Автоматизация

Интеграционные сценарии реализованы в классе `SubscriptionAndEventsIntegrationTest` с использованием следующих приёмов:

- аннотации `@SpringBootTest(webEnvironment = RANDOM_PORT)` и `@AutoConfigureMockMvc` поднимают полноценный Spring-контекст с MVC-слоем и H2-базой;
- с помощью `MockMvc` эмулируются реальные HTTP-запросы к контроллерам `/events` и `/subscriptions/{id}/important`;
- `MockHttpSession` используется для имитации текущего авторизованного пользователя, с которым работает `UserSessionService`;
- JPA-репозитории используются напрямую из теста для предзаполнения данных и последующей проверки состояния БД (флаг `important` и состав набора событий).



```
Run SubscriptionAndEventsIntegrationTest
Tests passed: 2 of 2 tests - 785 ms

01:56:44.979 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils -- Could not detect default configuration classes for test class [com.example.student.events.SubscriptionAndEventsIntegrationTest]:
01:56:45.807 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper -- Found @SpringBootConfiguration com.example.scheduler.SchedulerApplication

:: Spring Boot ::
(v3.3.4)

2025-12-05T01:56:45.644+03:00 INFO 36376 --- [main] s.i.SubscriptionAndEventsIntegrationTest : Starting SubscriptionAndEventsIntegrationTest using Java 21.0.4 with P
2025-12-05T01:56:45.643+03:00 INFO 36376 --- [main] s.i.SubscriptionAndEventsIntegrationTest : The following 1 profile is active: "test"
2025-12-05T01:56:47.736+03:00 WARN 36376 --- [main] org.hibernate.orm.deprecation : HHH90000023: H2Dialect does not need to be specified explicitly using
2025-12-05T01:56:48.384+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : drop table if exists audit_log cascade
2025-12-05T01:56:48.387+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : drop table if exists events cascade
2025-12-05T01:56:48.387+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : drop table if exists snapshots cascade
2025-12-05T01:56:48.388+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : drop table if exists subjects cascade
2025-12-05T01:56:48.388+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : drop table if exists subscriptions cascade
2025-12-05T01:56:48.388+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : drop table if exists users cascade
2025-12-05T01:56:48.388+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : create table audit_log (created_at timestamp(6) with time zone not null, id bigint generated by default as identity, target_id bigint, action varchar(255), actor varchar(255))
2025-12-05T01:56:48.392+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : create table events (created_at timestamp(6) with time zone not null, id bigint generated by default as identity, subject_id bigint not null, payload_ham varchar(255))
2025-12-05T01:56:48.397+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : create table snapshots (id bigint generated by default as identity, snapshot_at timestamp(6) with time zone not null, subject_id bigint not null, payload_ham varchar(255))
2025-12-05T01:56:48.399+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : create table subjects (id bigint generated by default as identity, name varchar(255) not null, ruz_key varchar(255) not null unique, type enum ('COURSE', 'TEACHER') not null)
2025-12-05T01:56:48.399+03:00 DEBUG 36376 --- [main] org.hibernate.SQL : create table subscriptions (important boolean not null, created_at timestamp(6) with time zone not null, subject_id bigint not null, user_id bigint not null)
```

Нагрузочное тестирование

Проверить, укладывается ли эндпойнт чтения событий /events в заявленные нефункциональные требования при типичной нагрузке курса: P95 времени ответа чтения ≤ 300 мс и целевой диапазон производительности 200–400 RPS.

Тестовая среда

- Операционная система: локальная среда разработки.
- База данных: H2 in-memory (профиль test, как в интеграционных тестах).
- Приложение запускается тем же тестовым профилем, что и в интеграционном тестировании:
 - отключён планировщик и внешние HTTP-запросы к RUZ;
 - схема БД создаётся автоматически средствами Hibernate (ddl-auto=create-drop).

Методика

Нагрузочное тестирование реализовано в отдельном автоматизированном тесте EventsLoadTest со следующей логикой:

1. Инициализация данных
В таблицах subjects и events создаётся один предмет (Load Test Course) и 10 событий для него, чтобы страница /events возвращала непустой HTML.
2. Запуск приложения
Тест помечен @SpringBootTest(webEnvironment = RANDOM_PORT), поэтому поднимается полноценное приложение на случайном порту. Для отправки запросов используется TestRestTemplate.
3. Генерация нагрузки
 - создаётся пул из 20 потоков (ExecutorService)
 - каждый поток выполняет 50 запросов GET /events
 - суммарно за прогон выполняется 1000 запросов.
В каждом запросе измеряется время отсылки и получения ответа (наносекунды → миллисекунды).
4. Сбор метрик
После завершения всех запросов тест вычисляет:
 - общее время теста (time)
 - средний RPS: $RPS = total_requests / (elapsed_time_sec)$
 - 95-й перцентиль задержки ответа p95 (по отсортированному массиву всех измеренных латентностей).

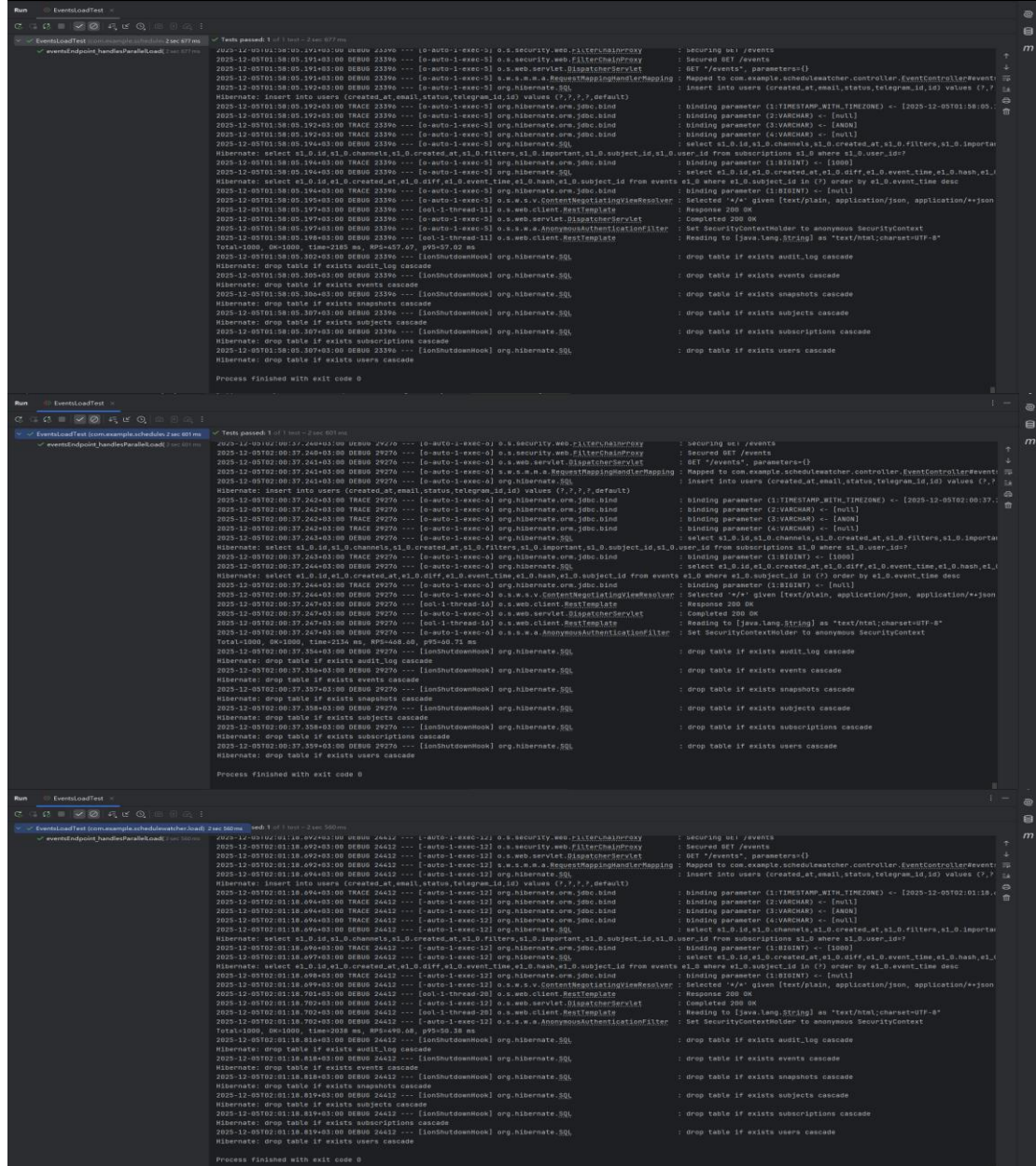
5. Критерии успешности

- все запросы должны завершиться со статусом 2xx (OK=1000);
- значения p95 должны укладываться в проектное ограничение ≤ 300 мс.

Результаты измерений

Тест EventsLoadTest был запущен три раза подряд. Получены следующие показатели:

Прогон	Total	OK	Время теста, ms	RPS	p95, ms
1	1000	1000	2185	457.67	57.02
2	1000	1000	2134	468.60	60.71
3	1000	1000	2038	490.68	50.38



Агрегированные значения:

- Средний RPS ≈ 472 запросов/сек (диапазон 458–491 RPS).
- Средний p95 ≈ 56 мс (диапазон 50–61 мс).
- Во всех трёх прогонах не было ни одной ошибочной операции (OK=1000).

Сопоставление с нефункциональными требованиями

Сравним полученные результаты с целями, сформулированными на этапе проектирования: P95 чтения ≤ 300 мс и ожидаемая нагрузка 200–400 RPS на пользовательский API.

- По задержкам: измеренный 95-й процентиль для /events находится в интервале **50–61** мс, что существенно лучше целевого ограничения 300 мс и оставляет достаточный запас на сетевые задержки и использование настоящей PostgreSQL в боевой среде.
- По производительности: достигнутый RPS 460–490 запросов/сек превышает усреднённый проектный диапазон 200–400 RPS и сопоставим с пиковыми значениями (до 1–2k RPS при кеш-хитах), указанными в архитектурной части отчёта.

Выводы и ограничения

1. /events демонстрирует:
 - устойчивую обработку 1000 запросов с 20 параллельными потоками;
 - RPS около 470 и p95 порядка 50–60 мс, что удовлетворяет и даже превосходит заявленные нефункциональные требования.
2. Полученные цифры нельзя напрямую интерпретировать как продуктивные показатели:
в реальной эксплуатации будут добавлены накладные расходы PostgreSQL, сетевой задержки, кешей, внешних вызовов и т.д. Тем не менее, результаты показывают, что:
 - контроллер /events и слой чтения из БД реализованы эффективно;
 - при переходе на PostgreSQL ожидается, что p95 останется значительно ниже 300 мс при должной настройке индексов и кеширования.

Сборка

1. запустить сборку

```
D:\Course-project>docker build -t schedule-watcher .
[+] Building 936.7s (17/17) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 417B                                0.0s
=> [internal] load metadata for docker.io/library/eclipse-temurin:17-jre 3.9s
=> [internal] load metadata for docker.io/library/maven:3.9-eclipse-temurin-17 4.0s
=> [auth] library/maven:pull token for registry-1.docker.io        0.0s
=> [auth] library/eclipse-temurin:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore                                    0.0s
=> => transferring context: 171B                                     0.0s
=> [build 1/6] FROM docker.io/library/maven:3.9-eclipse-temurin-17@sha256:e4a7ace3dc0d645ed97f 0.1s
=> => resolve docker.io/library/maven:3.9-eclipse-temurin-17@sha256:e4a7ace3dc0d645ed97f8d9ad0 0.1s
=> [internal] load build context                                    0.0s
=> => transferring context: 66.46kB                                  0.0s
=> [stage-1 1/3] FROM docker.io/library/eclipse-temurin:17-jre@sha256:75ab7d1b4b18483e9245342c 0.1s
=> => resolve docker.io/library/eclipse-temurin:17-jre@sha256:75ab7d1b4b18483e9245342cbee253b5 0.1s
=> CACHED [build 2/6] WORKDIR /app                                  0.0s
=> [build 3/6] COPY pom.xml .                                       0.1s
=> [build 4/6] RUN mvn -q -DskipTests dependency:go-offline         892.4s
=> [build 5/6] COPY src ./src                                       0.3s
=> [build 6/6] RUN mvn -q -DskipTests package                       37.5s
=> CACHED [stage-1 2/3] WORKDIR /app                                  0.0s
=> [stage-1 3/3] COPY --from=build /app/target/schedule-watcher-1.0.0.jar /app/app.jar 0.2s
=> exporting to image                                              1.9s
=> => exporting layers                                              1.5s
=> => exporting manifest sha256:0de095fe9b1dcf1c67885cdc088ff468b97dd5fb936de43faf2fd49b44fdd87 0.0s
=> => exporting config sha256:8b7efa9c6b362365b62b3e9f3c95c08ea39b09bc48d16b3fa0567e7484c89979 0.0s
=> => exporting attestation manifest sha256:a1c9d6f56fe9cb8e8e33a79657fbb8b05911b35685013c884f 0.1s
=> => exporting manifest list sha256:353701f4bafd18838b395d5f4a7a16855e75dec317313f9e1227f561ca 0.0s
=> => naming to docker.io/library/schedule-watcher:latest          0.0s
=> => unpacking to docker.io/library/schedule-watcher:latest       0.3s
```

2. запустить unit тесты

```
D:\Course-project>docker run --rm -v "%cd%:/app" -w /app maven:3.9-eclipse-temurin-17 mvn test
[INFO] Scanning for projects...
```

```
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.example.schedulewatcher.controller.AdminControllerTest
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.284 s -- in com.example.schedulewatcher.controller.AdminControllerTest
[INFO] Running com.example.schedulewatcher.controller.AdminSubscriptionControllerManualDiffTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.087 s -- in com.example.schedulewatcher.controller.AdminSubscriptionControllerManualDiffTest
[INFO] Running com.example.schedulewatcher.controller.EventControllerTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.176 s -- in com.example.schedulewatcher.controller.EventControllerTest
[INFO] Running com.example.schedulewatcher.controller.SubscriptionControllerTest
[INFO] Tests run: 10, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.213 s -- in com.example.schedulewatcher.controller.SubscriptionControllerTest
[INFO] Running com.example.schedulewatcher.controller.WebControllerTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.018 s -- in com.example.schedulewatcher.controller.WebControllerTest
[INFO] Running com.example.schedulewatcher.service.NotificationServiceTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.255 s -- in com.example.schedulewatcher.service.NotificationServiceTest
[INFO] Running com.example.schedulewatcher.service.RuzClientTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.726 s -- in com.example.schedulewatcher.service.RuzClientTest
[INFO] Running com.example.schedulewatcher.service.ScheduleFetchServiceTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.118 s -- in com.example.schedulewatcher.service.ScheduleFetchServiceTest
[INFO] Running com.example.schedulewatcher.service.SubscriptionServiceTest
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.040 s -- in com.example.schedulewatcher.service.SubscriptionServiceTest
[INFO] Running com.example.schedulewatcher.service.UserSessionServiceTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.011 s -- in com.example.schedulewatcher.service.UserSessionServiceTest
[INFO] Running com.example.schedulewatcher.util.ScheduleDiffUtilTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.015 s -- in com.example.schedulewatcher.util.ScheduleDiffUtilTest
[INFO]
[INFO] Results:
[INFO] Tests run: 43, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jacoco:0.8.11:report (report) @ schedule-watcher ---
[INFO] Loading execution data file /app/target/jacoco.exec
[INFO] Analyzed bundle 'Schedule Watcher' with 12 classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 02:12 min
[INFO] Finished at: 2025-12-08T19:42:03Z
[INFO] -----
```


3. запустить интеграционные тесты

```
D:\Course-project>docker run --rm -v "%cd%:/app" -w /app maven:3.9-eclipse-temurin-17 mvn test -Dgroups=integration -DexcludedGroups=
[INFO] Scanning for projects...
```

```
[INFO] -----
[INFO] T E S T S
[INFO] Running com.example.schedulatcher.integration.SubscriptionAndEventsIntegrationTest
19:52:44.185 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils -- Could not detect default configuration classes for test class [com.example.schedulatcher.integration.SubscriptionAndEventsIntegrationTest]: SubscriptionAndEventsIntegrationTest does not declare any static, non-private, non-final, nested classes annotated with @Configuration.
19:52:44.515 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper -- Found @SpringBootConfiguration com.example.schedulatcher.SchedulatcherApplication for test class com.example.schedulatcher.integration.SubscriptionAndEventsIntegrationTest

:: Spring Boot ::      (v3.3.0)
```

```
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 19.19 s -- in com.example.schedulatcher.integration.SubscriptionAndEventsIntegrationTest
[INFO] Results:
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- jacoco:0.8.11:report (report) @ schedule-watcher ---
[INFO] Loading execution data file /app/target/jacoco.exec
[INFO] Analyzed bundle 'Schedule Watcher' with 12 classes
[INFO] BUILD SUCCESS
[INFO] Total time: 82:46 min
[INFO] Finished at: 2025-12-08T19:53:11Z
[INFO]
```

4. запуск приложения

```
D:\Course-project>docker compose up --build
[+] Building 3.8s (10/13) FINISHED
=> [internal] load local bake definitions                                0.0s
=> reading from stdin 0/20                                             0.0s
=> [internal] load build definition from Dockerfile                    0.0s
=> transferring dockerfile: 417B                                       0.0s
=> [internal] load metadata for docker.io/library/eclipse-temurin:17-jre 3.1s
=> [internal] load metadata for docker.io/library/maven:3.9-eclipse-temurin-17 0.4s
=> [auth] library/eclipse-temurin:pull taken from registry-1.docker.io 0.0s
=> [internal] load .dockerignore                                         0.0s
=> transferring context: 171B                                           0.0s
=> [build 1/6] FROM docker.io/library/maven:3.9-eclipse-temurin-17@sha256:ea7ace3dc8d645ed97f8d9ad8bd3fbb14faad158138f27f116d77185a639b62 0.1s
=> resolve docker.io/library/maven:3.9-eclipse-temurin-17@sha256:ea7ace3dc8d645ed97f8d9ad8bd3fbb14faad158138f27f116d77185a639b62 0.1s
=> [internal] load build context                                         0.0s
=> transferring context: 6.61kB                                          0.0s
=> [stage-1 1/3] FROM docker.io/library/eclipse-temurin:17-jre@sha256:75ab7d1bb18483e9245342cbee253b558952c1def5c1c18956196338a81683e 0.1s
=> resolve docker.io/library/eclipse-temurin:17-jre@sha256:75ab7d1bb18483e9245342cbee253b558952c1def5c1c18956196338a81683e 0.1s
CACHED [stage-1 2/3] WORKDIR /app                                     0.0s
=> CACHED [build 2/6] WORKDIR /app                                       0.0s
=> CACHED [build 3/6] COPY pom.xml                                       0.0s
=> CACHED [build 4/6] RUN mvn -q -DskipTests dependency:go-offline      0.0s
=> CACHED [build 5/6] COPY src ./src                                     0.0s
=> CACHED [build 6/6] RUN mvn -q -DskipTests package                    0.0s
=> CACHED [stage-1 3/3] COPY --from=build /app/target/schedule-watcher-1.0.0.jar /app/app.jar 0.0s
=> exporting to image                                                    0.2s
=> exporting layers                                                       0.0s
=> exporting manifest sha256:71f402d87025bf7d3ee02dc4e8c148b7707d17f6c76e9a6d4d16ad5b7262ae 0.0s
=> exporting config sha256:c9a605f667f906d921999252581a7761588a5888d156b567631ca37827e708 0.0s
=> exporting attestation manifest sha256:86bc199c57d127798329384f8cabef8724c91f9116ee856537282a7cb569e2 0.1s
=> exporting manifest list sha256:a280ec18c7131a58b94f2e94c511943ca236ad4e6993c8ac78f8da5a18c1f 0.0s
=> naming to docker.io/library/course-project-app:latest               0.0s
=> unpacking to docker.io/library/course-project-app:latest            0.0s
=> resolving provenance for metadata file                                0.0s
[+] Building 2.2s
course-project-app Built 0.0s
Container sr_app Recreated 0.1s
```