# Тестовое задание

Для работы использовать java 15 build tool maven DB PostgresSQL.

Реализовать импорт данных из Старой системы в Новую систему. Импортируются текстовые заметки. Текстовая заметка делается конкретным пользователем (user) для конкретного пациента (clients в Старой системе и patients в Новой системе). При импорте:

* Заметка в Старой системе идентифицируется по полю guid заметки в Старой системе.
* Дата создания заметки в Новой системе должна равняться дате создания заметки в Старой системе
* В новой системе заметка должна также привязываться к пациенту и пользователю системы.
* Сопоставление клиентов из Старой системы и пациентов из Новой системы:
  + Одному пациенту в Новой системе может соответствовать либо один и более клиентов в Старой системе, либо ни одного клиента.
  + Одному клиенту в Старой системе соответствует один пациент либо ни одного пациента в Новой системе.
  + В старой системе идентификатором клиента является поле guid.
  + В Новой системе у пациента в таблице patient\_profile есть строковое поле old\_client\_guid которое содержит набор guid из старой системы записанных через разделитель (запятую). Можно предложить и реализовать альтернативный механизм хранения списка guid в patient\_profile.
* Сопоставление пользователей из Строй и Новой систем:
  + Одному пользователю в Старой системе соответствует один пользователь в Новой системе, либо ни одного пользователя.
  + Идентификатором пользователя в Старой системе является логин пользователя client\_note.logged\_user.
  + Пользователь в Новой системе имеет поле company\_user.login. Для одного и того же пользователя логины в обеих системах совпадают. Если пользователя с таким логином в новой системе нет, то его необходимо будет создать.
* Импорт в приложении запускается по расписанию без участия пользователя.
* Импортируются заметки только для пациентов которые имеют в Новой системе статус Active (patient\_profile.status\_id имеет одно из следующих значений 200,210,230)
* Импорт должен также обновлять в Новой системе импортированные ранее заметки, если они были изменены в Старой системе. Если заметка была изменена и в старой, и в новой системе, то должна сохраниться наиболее поздняя по дате изменения версия.
* Реализовать логирование сбоев в процессе импорта. Реализовать логирование статистики импорта.
* Тело заметки содержится в поле client\_note.comments

Для получения данных из Старой системы используется REST API.

* Метод /clients возвращает список клиентов из старой системы
* Метод /notes возвращает список заметок для конкретного клиента из старой системы. Для вызова данного метода кроме clientGuid необходимо знать agency к которому приписан client в старой системе. Код agency можно получить только из сущности client из поле agency. В новой системе информация об agency отсутствует.

Скрипт создания пациентов и пользователей в Новой системе предоставляется.

Количество пациентов в Новой системе 5 тысяч, количество заметок у пациента в среднем 15.

Результат:

* Разработать реализацию импорта ноутов как Spring Boot приложение с встроенным запуском импорта по расписанию (каждые два часа, в 15 минут первого часа).
* Использовать JPA, желательно Spring Data. Создать минимально необходимые сущности JPA.
* При необходимости можно вносить изменения в структуру (добавлять поля и таблицы) в существующую БД Новой системы.
* Код должен компилироваться. Наличие тестов будет плюс.
* Имитация REST API старой системы доступна в виде приложения, которое можно запустить локально.

REST API получения списка client и notes из старой системы.

//-------------------------------------------

Get all clients

//-------------------------------------------

Request URL: /clients

Request Method: POST (empty body)

Response

[

    {

        "agency": "vhh4",

        "guid": "01588E84-D45A-EB98-F47F-716073A4F1EF",

        "firstName": "Ne",

        "lastName": "Abr",

        "status": "INACTIVE",

        "dob": "10-15-1999",

        "createdDateTime": "2021-11-15 11:51:59"

    }

]

//-------------------------------------------

Get all client notes for client with exactly client guid

//------------------------------------------

Request URL: /notes

Request Method: POST

Request Payload

{

    "agency": "vhh4",

    "dateFrom": "2019-09-18",

    "dateTo": "2021-09-17",

    "clientGuid": "01588E84-D45A-EB98-F47F-716073A4F1EF"

}

Где “agency” – соответствующий agency клиента, а “clientGuid” – его guid.

“dateFrom” и “dateTo” задают временной промежуток выборки заметок

Response

[

    {

        "comments": "Patient Care Coordinator, reached out to patient caregiver is still in the hospital.",

        "guid": "20CBCEDA-3764-7F20-0BB6-4D6DD46BA9F8",

        "modifiedDateTime": "2021-11-15 11:51:59",

        "clientGuid": "C5DCAA49-ADE5-E65C-B776-3F6D7B5F2055",

        "datetime": "2021-09-16 12:02:26 CDT",

        "loggedUser": "p.vasya",

        "createdDateTime": "2021-11-15 11:51:59"

    }

]

“clientGuid”– guid клиента.

“loggedUser” – логин пользователя, создавшего заметку. В новой системе это будет поле company\_user.login

# Таблицы в новой системе

## Таблица пользователей Новой системы

**CREATE** **TABLE** company\_user (

id **int8** **NOT** **NULL** **GENERATED** **BY** **DEFAULT** **AS** **IDENTITY**,

login **varchar**(255) **NOT** **NULL**,

**CONSTRAINT** company\_user\_pkey **PRIMARY** **KEY** (id),

**CONSTRAINT** company\_user\_uniq\_login **UNIQUE** (login)

);

## Пациенты в Новой системе:

**CREATE** **TABLE** patient\_profile (

id **int8** **NOT** **NULL** **GENERATED** **BY** **DEFAULT** **AS** **IDENTITY**,

first\_name **varchar**(255) **NULL**,

last\_name **varchar**(255) **NULL**,

old\_client\_guid **varchar**(255) **NULL**,

status\_id **int2** **NOT NULL**,

**CONSTRAINT** patient\_profile\_pkey **PRIMARY** **KEY** (id)

);

## Заметки

**CREATE** **TABLE** patient\_note (

id **int8** **NOT** **NULL** **GENERATED** **BY** **DEFAULT** **AS** **IDENTITY**,

created\_date\_time **timestamp** **NOT** **NULL**,

last\_modified\_date\_time **timestamp** **NOT** **NULL**,

created\_by\_user\_id **int8** **NULL**,

last\_modified\_by\_user\_id **int8** **NULL**,

note **varchar**(4000) **NULL**,

patient\_id **int8** **NOT** **NULL**,

**CONSTRAINT** patient\_note\_pkey **PRIMARY** **KEY** (id)

);

**ALTER** **TABLE** patient\_note **ADD** **CONSTRAINT** fk\_pat\_note\_to\_modifyed\_user **FOREIGN** **KEY** (last\_modified\_by\_user\_id) **REFERENCES** company\_user(id);

**ALTER** **TABLE** patient\_note **ADD** **CONSTRAINT** fkpat\_note\_to\_created\_user **FOREIGN** **KEY** (created\_by\_user\_id) **REFERENCES** company\_user(id);

**ALTER** **TABLE** patient\_note **ADD** **CONSTRAINT** fk\_pat\_note\_to\_patient **FOREIGN** **KEY** (patient\_id) **REFERENCES** patient\_profile(id);