

Вимоги та рекомендації розробнику щодо моделювання регламенту реєстра

ЗМІСТ

1. Загальний опис	1
2. Вимоги до роботи з моделлю даних	2
2.1. Вибір імен для таблиць і критеріїв пошуку	3
2.2. Підтримка актуальності changeSet	3
2.3. Встановлення актуальних колонок в критеріях пошуку	3
2.4. Використання пагінації і лімітів у критеріях пошуку	4
2.5. Використання індексів для пошуку у базі даних	4
2.6. Аналіз створених запитів у pgAdmin	4
3. Вимоги на етапі створення бізнес-процесу	4
3.1. Встановлення часового обмеження на користувацьку задачу	4
3.2. Створення можливості повернутися на конкретний етап бізнес-процесу, який був створений раніше, та до головного меню	5
3.3. Налаштування відправки повідомлень в особистий кабінет користувача	5
3.4. Налаштування в кабінеті посадової особи можливості відхилити реєстрацію форми	5
3.5. Ідентифікатори послуг в кабінеті посадової особи	5
3.6. Налаштування прав доступу	6
3.7. Категоризація доступних послуг в кабінеті користувача	7
4. Вимоги на етапі створення UI форм	7
5. Вимоги до формування аналітичних звітів у застосунку Redash	8



ВАЖЛИВО При розробці реєстру наполегливо рекомендується притримуватись наданих нижче рекомендацій та обов'язково проводити [аудит реєстру](#).

1. Загальний опис

Під час створення бізнес-процесу перед розробником стоїть задача не лише створити логічну модель бізнес-процесу, яка описуватиме послуги, що надаються фізичним і юридичним особам державними органами, а також реалізувати додаткові можливості під час роботи з регламентом.

Під час створення бізнес-процесу розробник має дотримуватися загальних рекомендацій

щодо розробки регламенту реєстрів:

- При моделюванні бізнес-процесу дотримуйтесь принципів [KISS](#). Необхідно уникати створення великих бізнес процесів, при можливості розділяйте великі бізнес-процеси на менші
- При активній розробці регламенту реєстрів необхідно робити повний [cleanup](#), узгоджений з командою, мінімум раз на тиждень. Ця вимога стосується виключно dev середовища.
- При створенні UI форм не рекомендується використовувати елементи із розділу “Експериментальні” для їх коректної роботи

Основні вимоги до розробника реєстру можна поділити на такі типи (в залежності від етапу, коли їх необхідно реалізувати):

- вимоги до роботи з моделлю даних
- вимоги до моделювання бізнес-процесу в кабінеті адміністратора регламентів чи в застосунку для ПК Camunda Modeler
- вимоги до створення UI форми в кабінеті адміністратора регламентів
- вимоги до формування аналітичних звітів

2. Вимоги до роботи з моделлю даних

Для створення фізичної моделі даних під час розробки регламенту реєстру використовується [Liquibase](#) - система управління версіями баз даних, що описується за допомогою строго типізованих XML-шаблонів. Під час розробки моделі даних розробники мають дотримуватися наступних вимог:

- необхідно дотримуватися одного стилю при створенні назв для таблиць, полів таблиць, критеріїв пошуку, аналітичних представлень, індексів, ключів
- всі імена та id повинні бути інформативні
- при командній роботі важливо забезпечити структуровану організацію усіх файлів (створення сутностей, бізнес процеси, форми) для їх легкого розуміння та управління
- для підтримки сумісності регламенту реєстра і платформи необхідно використовувати лише теги із [стандартної бібліотеки Liquibase](#) або бібліотеки [liquibase-ddm-ext](#), які описані у документації
- уникайте створення зайвих changeSet та слідкуйте за їх актуальністю
- встановлюйте лише ті колонки, які дійсно необхідні для критеріїв пошуку
- використовуйте пагінацію та ліміти на критеріях пошуку
- використовуйте індекси для пришвидшення пошуку у базі даних
- після створення складних багатотабличних критеріїв пошуку необхідно переглядати [план відпрацювання запитів](#)
- за можливістю уникайте передачі з форм людиночитних параметрів

2.1. Вибір імен для таблиць і критеріїв пошуку

Важливо дотримуватися одного стилю під час вибору імен для таблиць і критеріїв пошуку - всі назви мають бути або у snake_case, або у camelCase. Для кращої ідентифікації. Для пошукових критеріїв додавайте префікс або суфікс "sc" до назви (важливо дотримуватися обраного варіанту для назв усіх пошукових критеріїв). Таким чином, за назвою буде легше ідентифікувати таблицю та критерій пошуку. Приклад рекомендованої назви: для таблиці - koatuu для критерія пошуку - `sc_koatuu_contains_name` або `koatuu_contains_name_sc`

2.2. Підтримка актуальності changeSet

Із зростанням кількості changeSet збільшується складність файлу, що робить його важким у керуванні та розумінні. Чим більше створено changeSet, тим більше часу займає застосування змін в базі даних. Тому необхідно відстежувати актуальність changeSet, особливо після cleanup. Під час роботи над регламентом реєстру можливо послідовно створювати різну кількість changeSet на змінення, видалення чи повторне створення таблиць та їх полів. У цьому випадку читабельність файлу зменшується, тому, якщо до cleanup були присутні послідовні changeSet на виправлення, видалення або внесення змін до моделі даних, то після cleanup рекомендується видалити ChangeSet, які модифікували таблиці, і залишити лише ті, які створюють актуальну модель даних. Якщо треба внести зміни у створену таблицю, рекомендується видалити всю таблицю (а також залежні від неї компоненти в зворотньому до їх створення порядку) і створити заново таблицю з актуальними змінами.

Видалення одразу декількох таблиць або критеріїв пошуку можна об'єднати в один changeSet:

```
<changeSet id="drop-table" author="me">
<dropTable tableName="test" />
<dropTable tableName=" test_hst"/>
</changeSet>
```

2.3. Встановлення актуальних колонок в критеріях пошуку

У критеріях пошуку необхідно вказувати лише ті колонки, які необхідні для використання іншими підсистемами. Чим більше колонок вказано у критерії пошуку, тим повільніше буде виконуватися запит. Краще створювати кілька критеріїв пошуку для використання на різних етапах бізнес-процесу.

Важливо враховувати продуктивність створених критеріїв пошуку при роботі з великими обсягами даних, адже певні оператори та функції (наприклад, ORDER BY, GROUP BY, JOIN, функції агрегації COUNT(), SUM(), AVG(), MIN(), MAX()) можуть призвести до значних витрат ресурсів. Тому для операторів, які вимагають значних обчислювальних ресурсів, доцільно використовувати індекси та аналізувати створені запити для визначення продуктивності.

2.4. Використання пагінації і лімітів у критеріях пошуку

Ліміти на критеріях пошуку дозволяють визначити максимальну кількість результатів, які повертаються за пошуковою умовою. Якщо не вказувати ліміти, то можливий витік даних і проблеми з продуктивністю системи. Детальніше про атрибут `limit`. Також для зменшення об'єму даних, що передаються системою, та зменшення навантаження на сервер бази даних разом з атрибутом `limit` рекомендовано використовувати атрибут `pagination`.

2.5. Використання індексів для пошуку у базі даних

При створенні критеріїв пошуку, необхідно пересвідчитись, що для полів, які задіяні для зв'язку таблиць та формування умов вибірки представлення, у таблицях з великою кількістю значень наявні індекси пошуку. Індекси допомагають покращити швидкість та продуктивність запитів до бази даних. Недоцільно використовувати індекси для пошуку у таблицях, в яких зберігаються невелика кількість значень, так як СКБД Postgres виділяє ресурси для створення і підтримання індексів і кожного разу при запиті пошуку до бази даних розраховує оптимальні варіанти як проводити пошук: якщо в таблиці зберігається невелика кількість даних, то СКБД Postgres не буде використовувати індекси для пошуку. У такому випадку індекси стають зайвими, адже вони не використовуються при пошуку, але займають пам'ять і ресурси СКБД Postgres. Ознайомтись детальніше з прикладом створення і використанні [індексу пошуку](#).

2.6. Аналіз створених запитів у pgAdmin

Для оптимізації SQL-запитів в PostgreSQL, одним з основних інструментів є команда `EXPLAIN` (дозволяє побачити план виконання запиту без його фактичного виконання) і її розширена версія `EXPLAIN ANALYZE` (показує план виконання і виконує запит, вимірюючи його реальний час і ресурси). Ці команди додаються перед самим запитом і дозволяють отримати детальну інформацію про те, як база даних планує виконати запит, і допомагають виявити потенційні проблеми з швидкодією. Рекомендується проводити аналіз усіх запитів до бази даних.

3. Вимоги на етапі створення бізнес-процесу

Нижче приведені основні вимоги під час розробки бізнес процесу.

3.1. Встановлення часового обмеження на користувацьку задачу

Покинуті з різних причин бізнес-процеси (помилка у бізнес-процесі чи незавершений користувачем процес), накопичуються у системі та фактично стають системним сміттям. Вони не мають перспективи завершення, тому наполегливо рекомендується до

користувацької задачі завжди додавати подію “Таймер”. Після правильного налаштування елементу з таймером незавершений процес у кабінеті користувача буде автоматично закриватися після відпрацювання таймера. Детальніше про подію [Timer](#). Рекомендуємо за замовчуванням додавати до БП подію “Таймер” з налаштуванням закривати бізнес-процес через тиждень. У окремих випадках, розуміючи бізнес вимоги замовника, збільшувати або зменшувати цей період. Особливо актуально для високонавантажених систем.

3.2. Створення можливості повернутися на конкретний етап бізнес-процесу, який був створений раніше, та до головного меню

Для можливості коригування раніше введених даних у користувача має бути можливість на етапі підпису форми повернутися на форму заповнення даних. Детальніше про [моделювання повернення до першої форми](#). Також користувач повинен мати можливість на етапі підпису форми повернутися до головного меню без збереження внесених у форму даних. Детальніше про [моделювання скасування внесення даних і повернення до головного меню](#).

3.3. Налаштування відправки повідомлень в особистий кабінет користувача

Для сповіщення користувачів про обробку їхньої заявки або зміну її статусу, необхідно відправляти in-app (inbox) повідомлення у кабінет користувача. Також рекомендується відправляти додаткові сповіщення коли час, встановлений таймером, закінчується, і бізнес-процес автоматично завершується. Більше про налаштування відправлення in-app повідомлень користувачам.

3.4. Налаштування в кабінеті посадової особи можливості відхилити реєстрацію форми


На етапі розгляду звернення посадовою особою необхідно додати додаткову гілку для можливості відмовити у наданні послуг (якщо така можливість передбачена на нормативному рівні). У такому випадку посадова особа має вказати причину відмови реєстрації і результат відмови, разом з вказаною причиною відмови має зберігатися в дата фабриці. У випадку відхилення посадовою особи реєстрації, рекомендується також сповіщати отримувача послуги щодо відхилення реєстрації і надавати йому можливість внести корективи до форми.

3.5. Ідентифікатори послуг в кабінеті посадової особи

В кабінеті посадової особи усі виконані бізнес-процеси зберігаються у вкладці “Мої послуги” → “Надані послуги”. Так як посадова особа може реєструвати одні і ті самі послуги різними особам, рекомендуємо додавати додаткові ідентифікатори, які надають можливість

відрізняти послуги для різних осіб по індивідуальним даним (ПІБ, ЄДРПОУ/РНОКПП). Для цього необхідно налаштувати ідентифікатори для процесів: наприклад, відображати ПІБ отримувача послуг або його ЄДРПОУ/РНОКПП, або інший ідентифікатор, який допоможе швидко віднайти потрібну послугу в переліку. Для реалізації ідентифікації під час моделювання бізнес-процесу використовуйте налаштування бізнес-ключів. Детальніше про [налаштування бізнес-ключів](#) на різних етапах бізнес-процесу.

Після налаштування бізнес-ключа, в кабінеті посадової особи для кожного налаштованого бізнес-процесу буде доданий ідентифікатор послуги

 Кабінет посадової особи			Головна	Доступні послуги	<u>Мої послуги</u>
<h1>Мої послуги</h1>					
<div>Послуги у виконанні</div> <div>Надані послуги</div>					
Послуга	Ідентифікатор послуги		Дата старту ↓		
2.04 Переведення з одного ЗО в інший протягом року за заявою	Сидоренко Іван Петрович		26.05.2022 11:14		
2.04 Переведення з одного ЗО в інший протягом року за заявою	Володар Вікторія Петрівна		26.05.2022 11:14		
2.04 Переведення з одного ЗО в інший протягом року за заявою	Володар Вікторія Петрівна		26.05.2022 11:14		

3.6. Налаштування прав доступу

При налаштуванні прав доступу до бізнес-процесу для уникнення зловживань, помилок та шахрайства необхідно створювати ролі, які мають мінімальну кількість необхідних прав і будуть потрібні лише для виконання конкретних етапів бізнес-процесу. Більше про [принципи управління доступом](#).

Рекомендації щодо налаштування прав доступу користувачів:

1. На самому початку користувачі-отримувачі послуг отримують тимчасові ролі `unregistered-individual` (для фізичних осіб), `unregistered-entrepreneur` (для фізичних осіб-підприємців), `unregistered-legal` (для юридичних осіб). Ці ролі система визначає автоматично, на основі даних електронного підпису та інформації з Єдиного державного реєстру (ЄДР).
2. Після відкриття особистого кабінету для користувачів з тимчасовими ролями з префіксом `unregistered-`, автоматично запускається єдиний доступний бізнес-процес — онбординг.

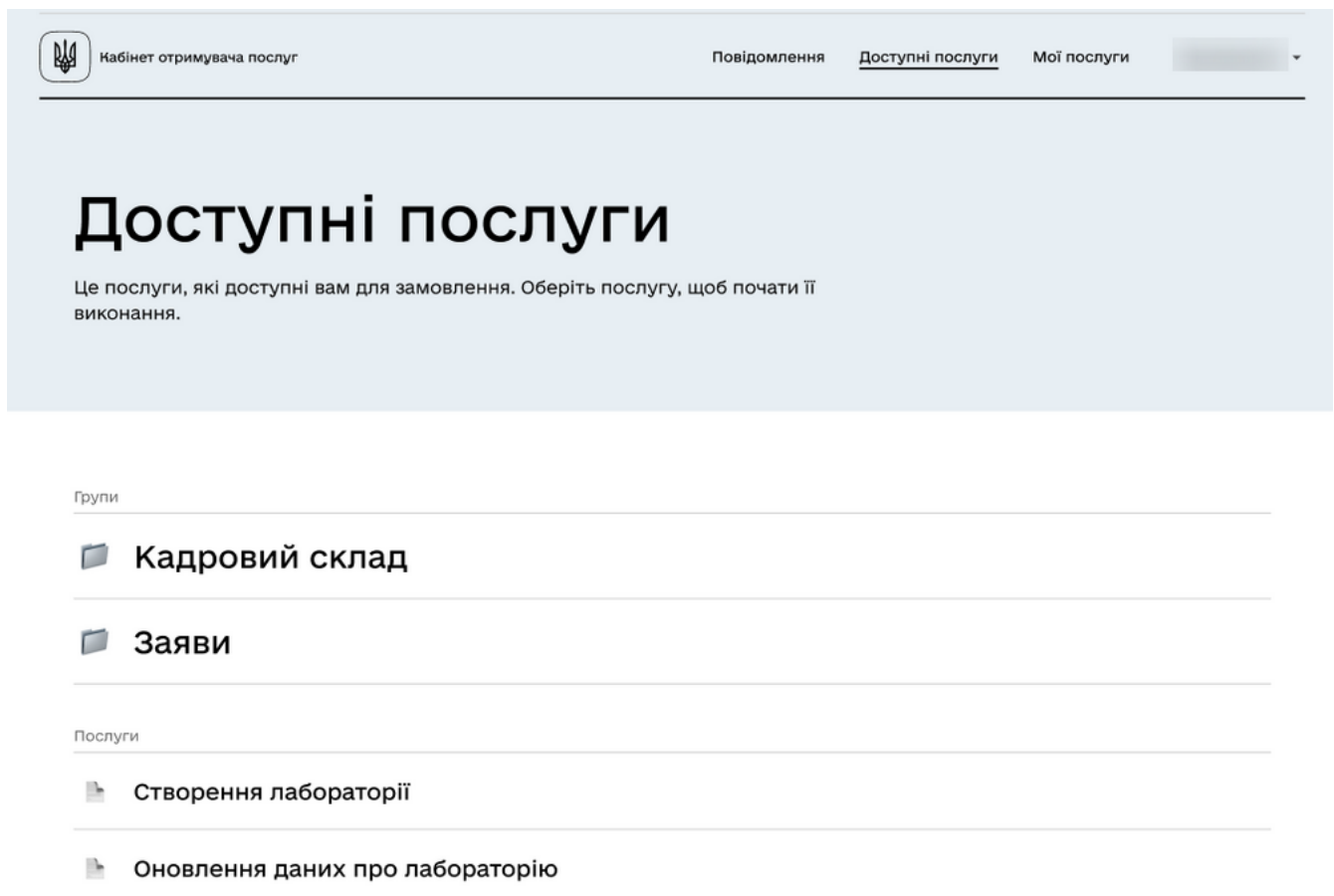
- Після успішного завершення процесу онбордингу, тимчасова роль користувача автоматично змінюється на відповідну постійну: individual (фізична особа), entrepreneur (фізична особа-підприємець), legal (юридична особа).
- Після присвоєння постійної ролі користувач отримує доступ до відповідних бізнес-процесів, згідно з його новою роллю.

Варто зазначити, що вище наведене стосується кабінету Отримувача послуг. Для Кабінету посадової особи використовується інший підхід.

3.7. Категоризація доступних послуг в кабінеті користувача

В кабінеті користувача може відображатися велика кількість послуг, в результаті чого виникають труднощі з пошуком необхідної послуги. Тому рекомендовано групувати можливі послуги по категоріям, в цьому випадку користувач бачить папки, що логічно об'єднують доступні для користувача папки. Налаштування категоризації послуг розробник може проводити через веб-інтерфейс адміністративного порталу. [Покрокова інструкція для категоризації доступних послуг в кабінеті користувача.](#)

Приклад групування послуг в кабінеті користувача:



4. Вимоги на етапі створення UI форм

Основні вимоги до UI форм:

форми мають бути якомога простішими якщо користувач має вносити велику кількість даних, краще створити декілька форм для поетапного внесення інформації необхідно встановлювати валідаційні правила для перевірки даних, які вносить користувач. Валідація має бути налаштована на усіх обов’язкових полях.

5. Вимоги до формування аналітичних звітів у застосунку Redash

У застосунку Redash є можливість створювати аналітичні звіти з відображенням всієї інформації, збереженої в базі даних. Ці звіти доступні для перегляду посадовими особами. Тому для надійного зберігання персональних даних отримувачів послуг і унеможливлення їх розповсюдження, варто уникати створення звітів, що містять персональні дані. Іншими словами, наявність в проекті звіта чи звітів, що містить персональні дані має бути продиктована або вимогами законодавства, або справді важливою бізнес потребою. В деяких випадках може бути використане маскування частини даних, наприклад замість повного номеру телефону може виводитись значення типу “097*12”. Але загальне правило: у звітах не повинно міститися конфіденційних даних отримувачів послуг таких, як повне ім’я, код ЄДРПОУ/РНОКПП, номер телефону, адреса або інші поля, які відносяться до персональних даних людини.