**Інструкція адміністратора**

**Scriptum**

Зміст

[1. Типи документів 4](#_Toc53141957)

[1.1. Головне 4](#_Toc53141958)

[1.2. Ролі 5](#_Toc53141959)

[1.3. Стани 7](#_Toc53141960)

[1.3.1. Налаштування можливих станів для даного типу документа. 7](#_Toc53141961)

[1.3.2. Додавання дії 7](#_Toc53141962)

[1.4. Процеси 9](#_Toc53141963)

[1.5. Події 10](#_Toc53141964)

[1.6. Конструктор 12](#_Toc53141965)

[1.6.1. Дизайнер 13](#_Toc53141966)

[1.6.2. Доступ 14](#_Toc53141967)

[1.6.3. Обов’язковість 15](#_Toc53141968)

[1.6.4. Перегляд 16](#_Toc53141969)

[1.7. Доступ 16](#_Toc53141970)

[1.8. Історія 18](#_Toc53141971)

[2. Налаштування 18](#_Toc53141972)

[2.1. Довідник ролей 18](#_Toc53141973)

[2.2. Довідник станів 20](#_Toc53141974)

[2.3. Журнал реєстрації 22](#_Toc53141975)

[2.4. Маски реєстраційних номерів 23](#_Toc53141976)

[2.5. Шаблон розпізнавання 25](#_Toc53141977)

[2.6. Шаблон документа 26](#_Toc53141978)

[2.7. Довідник типів зв’язків 29](#_Toc53141979)

[3. Бізнес-процеси 30](#_Toc53141980)

[3.1. Дефініції процесів 30](#_Toc53141981)

[3.2. Форми задач 30](#_Toc53141982)

[3.3. Дефініції завдань 32](#_Toc53141983)

[3.4. Процеси 32](#_Toc53141984)

[4. Довідники 33](#_Toc53141985)

[4.1. Групи виконавців 33](#_Toc53141986)

[4.2. Класифікатори 34](#_Toc53141987)

[4.3. Виконавці для груп виконавців 35](#_Toc53141988)

[5. Ярлики 36](#_Toc53141989)

[5.1. Створення ярликів 37](#_Toc53141990)

[5.2. Створення групи ярликів 38](#_Toc53141991)

[5.3. Редагування ярликів/груп ярликів 38](#_Toc53141992)

[5.3.1. Властивості 38](#_Toc53141993)

[5.3.2. Типізовані 39](#_Toc53141994)

[5.3.3. Команда 40](#_Toc53141995)

[5.4. Видалення ярликів/груп ярликів з панелі навігації 40](#_Toc53141996)

[6. Атрибути 40](#_Toc53141997)

[6.1. Бібліотека атрибутів 40](#_Toc53141998)

[6.2. Групи атрибутів 41](#_Toc53141999)

[6.3. Створення атрибутів та груп атрибутів 41](#_Toc53142000)

[7. Спеціальні атрибути 44](#_Toc53142001)

[7.1. Образ документа 44](#_Toc53142002)

[Завантаження образу документа з файлу 45](#_Toc53142003)

[Сканування образу документа 45](#_Toc53142004)

[Підпис образу документа КЕП 46](#_Toc53142005)

[Версії образу документа 46](#_Toc53142006)

[7.2. Додатки документів 46](#_Toc53142007)

[7.3. Зв’зки документів 48](#_Toc53142008)

[7.5. Обговорення по документу (чат) 48](#_Toc53142009)

# 1. Типи документів

Тип документа - базовий елемент Системи, що містить набір атрибутів, необхідний для формування документів в Системі, список ролей, що приймають участь в обробці документа, список станів, які документ може приймати в процесі обробки, опис екранної та друкованої форми документа, а також права доступу як до документу так і до окремих атрибутів документа в його різних станах.

Ярлик «Типи документів», що знаходиться в панелі навігації дозволяє створювати користувацькі налаштування для різних типів документів та, в подальшому, використовувати налаштування як повноцінний шаблон документа.

Створення нового типу документа відбувається по натисканню кнопки «Додати», що розташовано вгорі реєстру «Типи документів». Після чого відкривається вікно створення типу документа, що складається з таких вкладок (Рис. 1. 1):

* Головне
* Ролі
* Стани
* Процеси
* Події
* Конструктор
* Доступ
* Історія

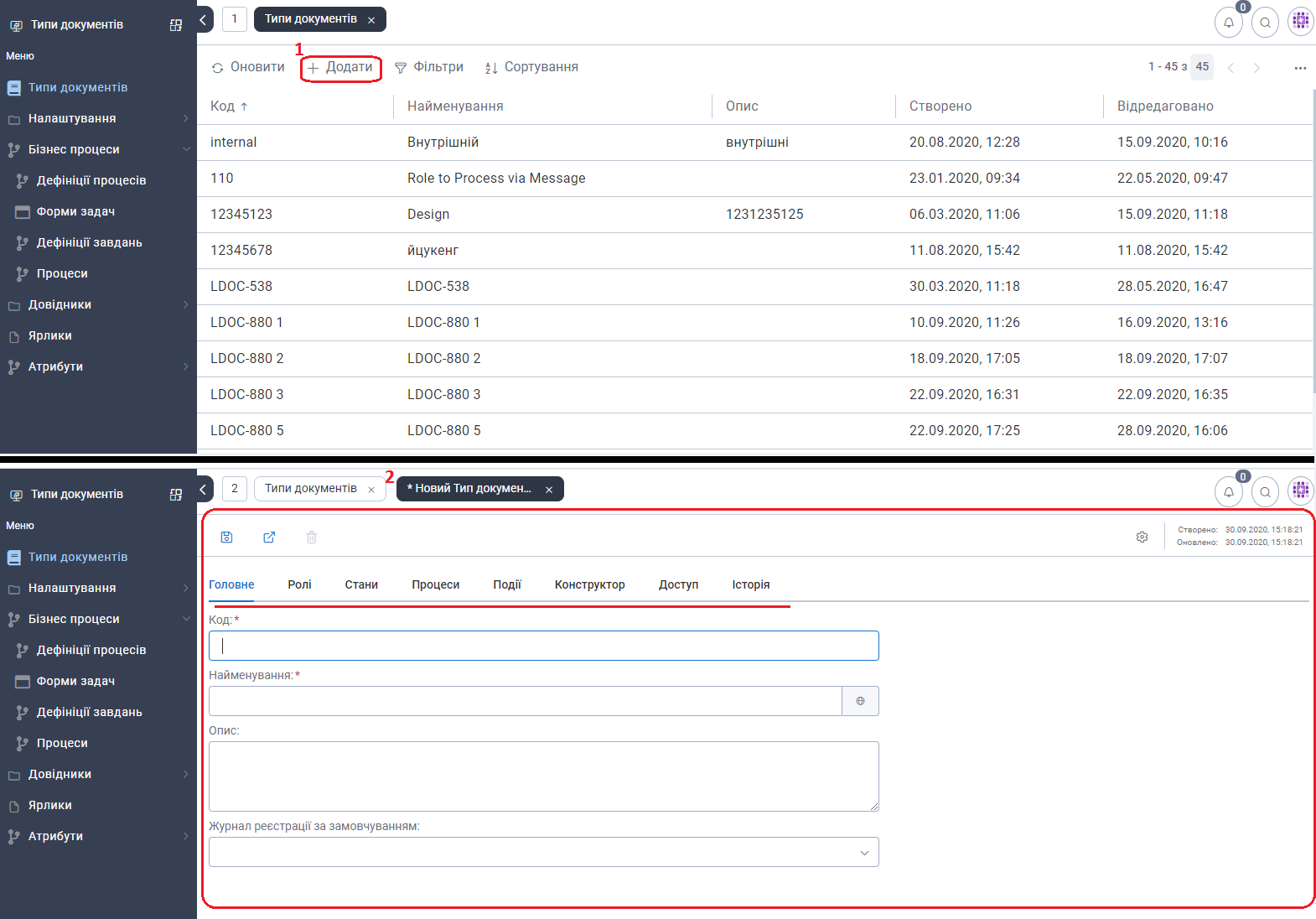


Рис. 1. 1. Створення Типу документа

## 1.1. Головне

В блоці «Головне» заповнюється загальна інформації про документ (Рис. 1. 2). Та складається з полів, що представлено в Таблиця 1.

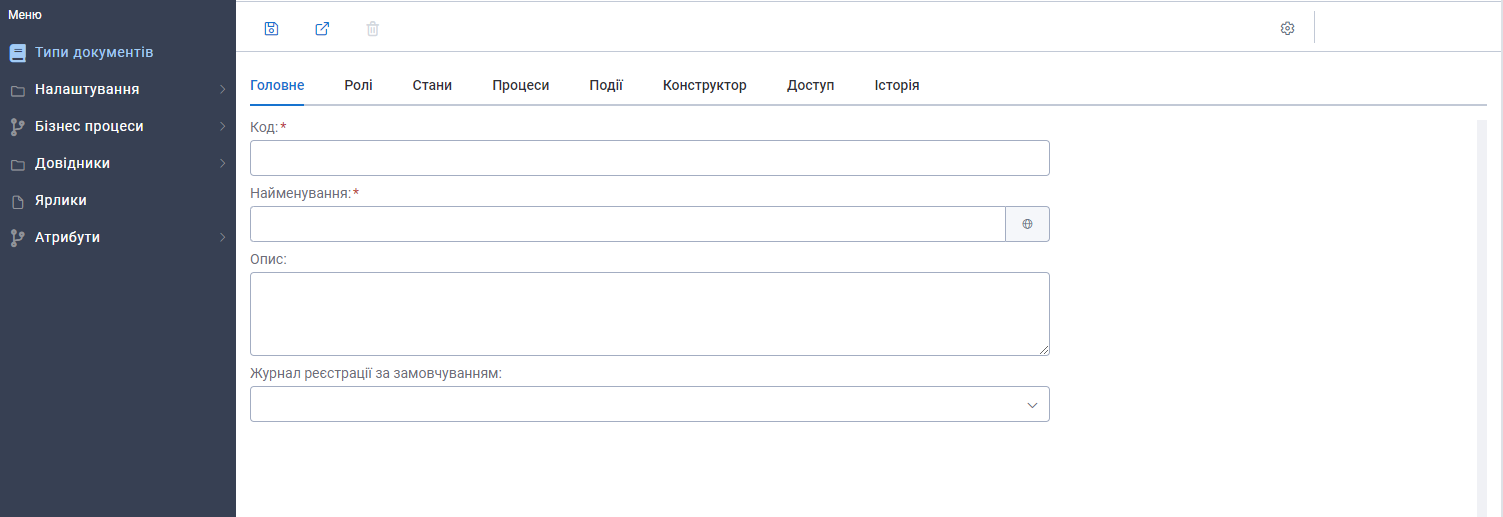


Рис. 1. 2. Вкладка «Головне» типу документів

Таблиця 1

| **Назва поля** | **Опис** |
| --- | --- |
| Код\* | Унікальний ідентифікатор типу документа. Текстове поле. |
| Найменування\* | Відображувана назва типу документа. |
| Опис | Короткий опис даного типу документа. |
| Журнал реєстрації за замовчуванням | Вибір журналу реєстрації за допомогою якого можлива реєстрація документів. Вказаний в даному полі журнал буду обиратися автоматично при створенні документів такого типу. |

Примітка: Позначені символом «\*» поля в даній таблиці та далі по документу – являються обов’язковими до заповнення.

На кожен тип документа можна створити журнал реєстрацій окрім стандартних, що описано в пункті 2.3. Журнал реєстрації.

## 1.2. Ролі

Для включення ролі в роботу з даним типом документів необхідно заповнити довідник ролей (Рис. 1. 3).

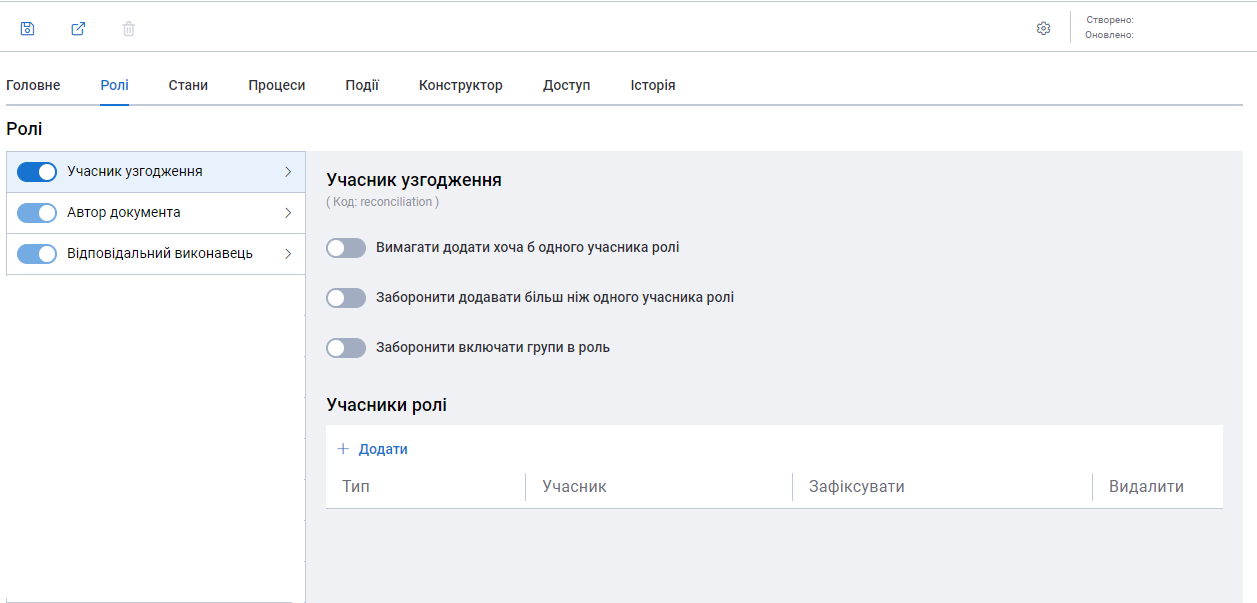


Рис. 1. 3. Налаштування ролей документа

Надання можливості роботи з даним типом документа здійснюється за допомогою активації перемикача для відповідної ролі (Рис. 1. 4).

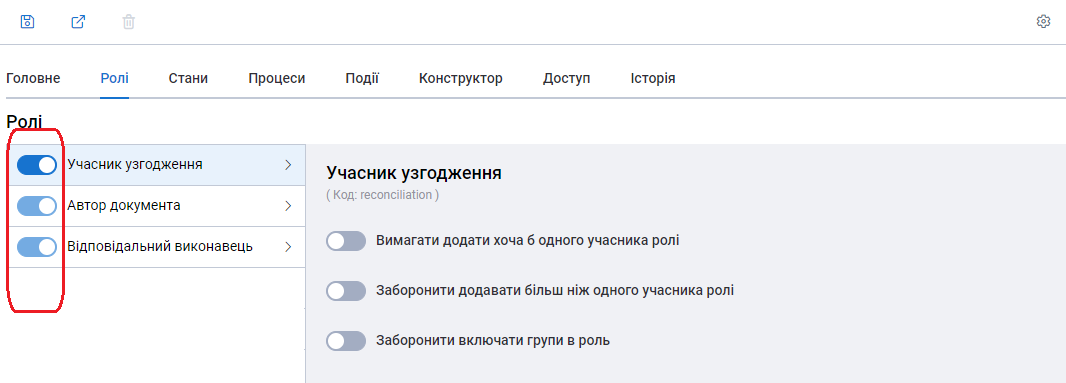


Рис. 1. 4. Активація ролей

Обов’язковою для документа являється роль «Автор документа», яка назначається автоматично на конкретного користувача, який створив документ.

Відповідальний виконавець – роль, на яку назначається відповідальний виконавець документу та яка автоматично включається в документ, але не є обов’язковою за умови, що вона не відключена в довіднику ролей.

Для кожної ролі можна додати, за допомогою активації відповідного перемикача, параметри (Рис. 1. 5):

* Вимагати додати хоча б одного учасника ролі – для даного типу документа обов’язково необхідний представник цієї ролі.
* Заборонити додавати більш ніж одного учасника ролі – для даного типу документів не можна додати біль ніж одного представника даної ролі.
* Заборонити включати групи в роль – для даного типу документа таку роль можна надати тільки конкретним штатним одиницям.

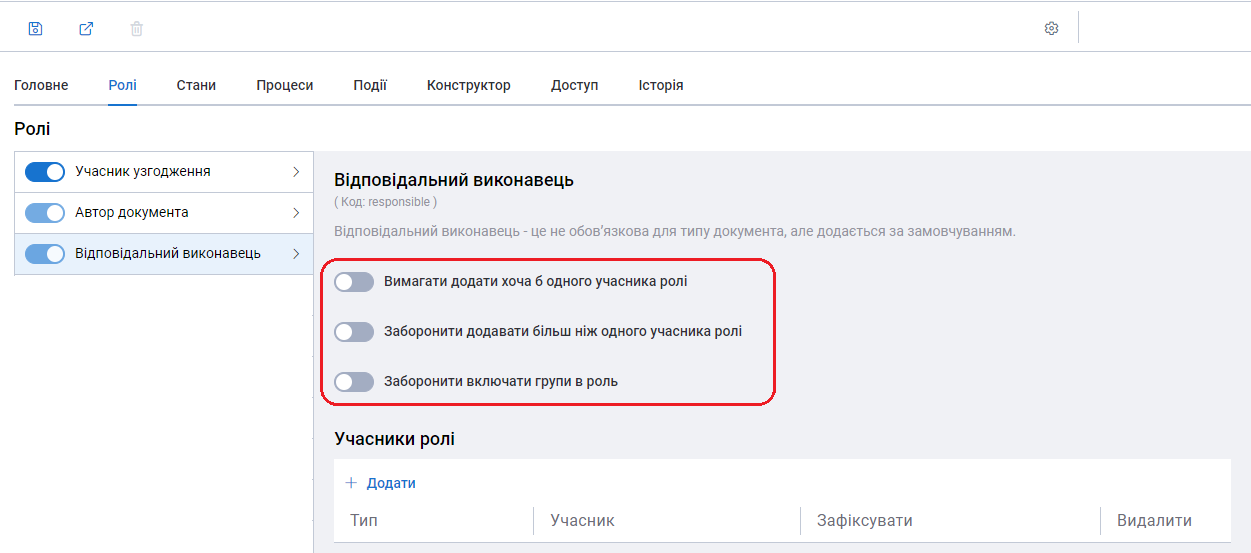


Рис. 1. 5. Параметри ролей

Для обраної ролі необхідно додати учасників – користувачів, які матимуть всі права та можливості відповідної ролі (Рис. 1. 6).

Учасники узгодження – всі хто мають узгодити даний документ та можуть назначатися як окремі користувачі – штатні одиниці, так і групи користувачів – групи виконавців (за умови, що вимикач «Заборонити включати групи в роль» не активований).

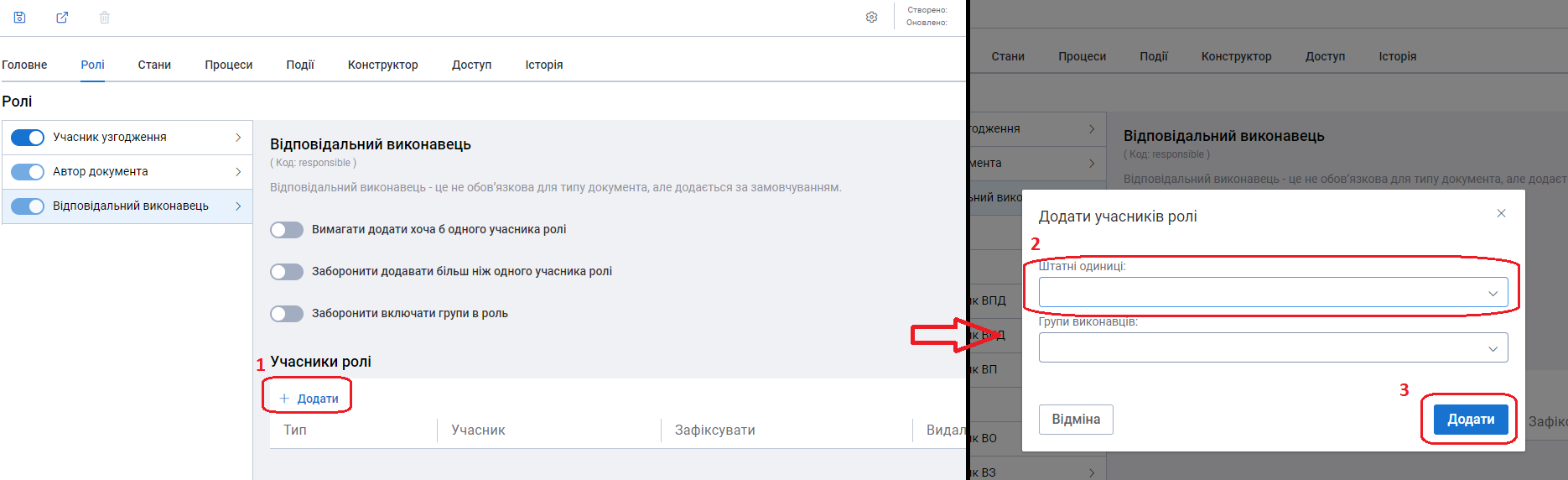


Рис. 1. 6. Додавання учасників ролей

При активації перемикача «Зафіксувати», даного користувача чи групу користувачів, неможна видалити з цієї ролі (Рис. 1. 7).

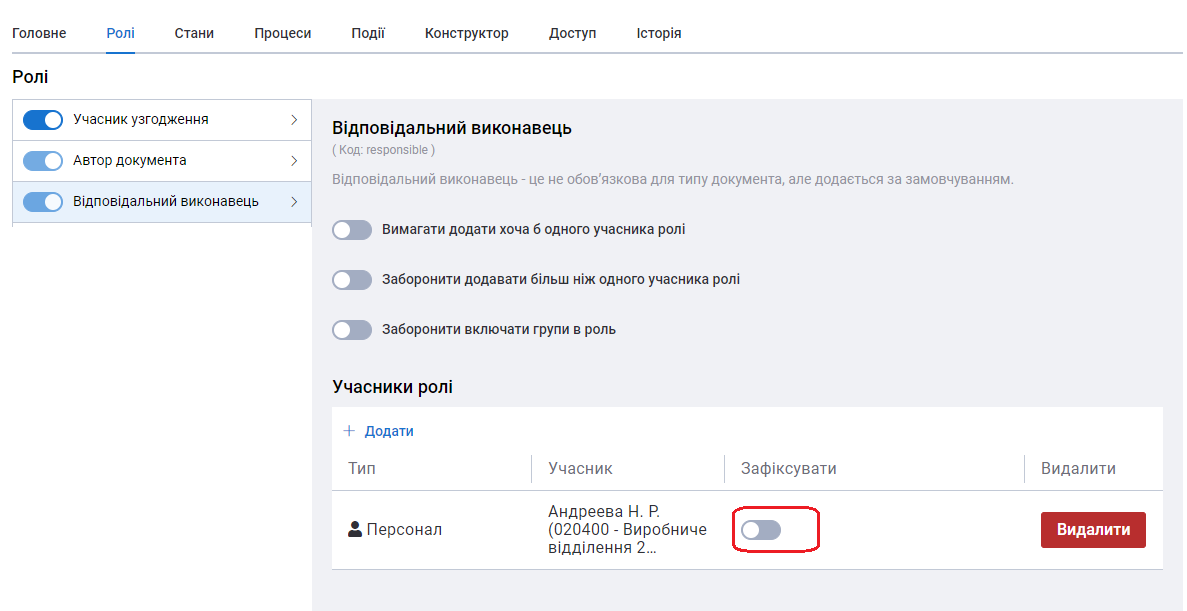


Рис. 1. 7. Фіксування користувача

Створення нової ролі описано в пункті 2.1. Довідник ролей.

## 1.3. Стани

### 1.3.1. Налаштування можливих станів для даного типу документа.

Для документа можна додати всі можливі стани з довідника станів.

Стан – це характеристика документа, що відображає результати проходження ним визначеного етапу обробки. Зміна стану документа здійснюється при проходженні ним одного з етапів обробки.

Для кожного стану можна дозволити чи заборонити видалення документів на цьому етапі.

Створення нового стану описано в пункті 2.2. Довідник станів.

### 1.3.2. Додавання дії

Для кожного стану можна налаштувати дії, при яких буде здійснюватися перехід до наступного стану (Рис. 1. 8). Як приклад:

Доступні стани документа:

* Новий – новий документ
* Завершений – документ закритий
* Чернетка – документ в стані чернетки
* На контролі – документ на контролі
* В обробці – документ знаходиться в стані обробки

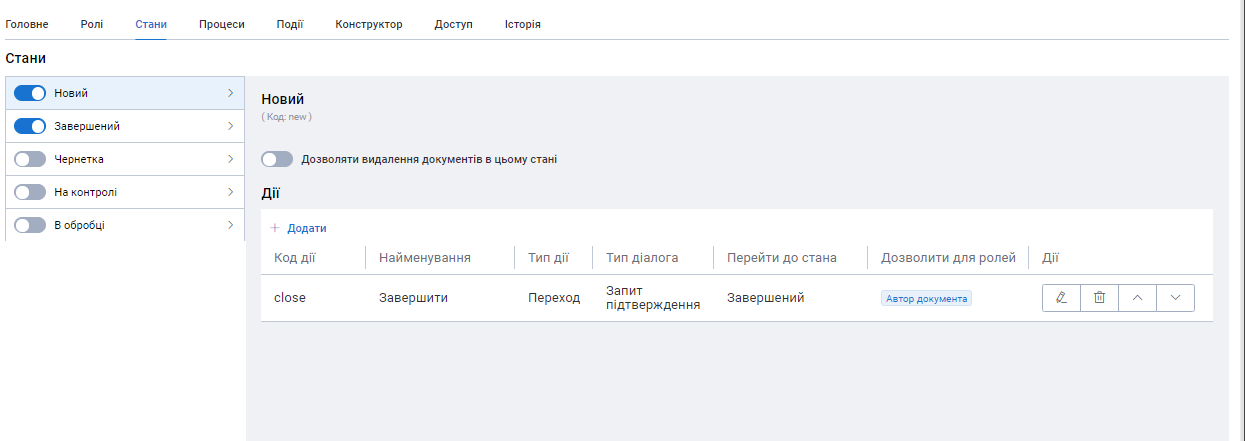


Рис. 1. 8. Додавання станів документу

Для кожного стану можна дозволити видалення документів активувавши перемикач «Дозволяти видалення документів в цьому стані», що розташований над блоком додавання дії (Рис. 1. 8).

Щоб додати дію, необхідно натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» та заповнити дані у формі, що з’явилась та натиснути кнопку «Додати дію» на формі (Рис. 1. 9). Поля форми описано в Таблиця 2.

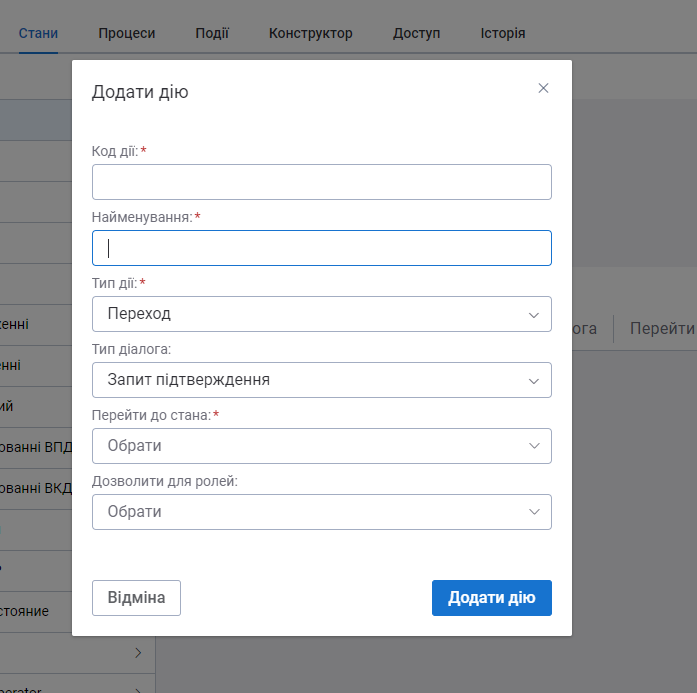


Рис. 1. 9. Параметри додавання дії

Таблиця 2

| **Назва поля** | **Опис** |
| --- | --- |
| Код дії\* | Унікальний ідентифікатор даної дії |
| Найменування\* | Відображувана назва даної дії |
| Тип дії\* | Вибір одного з варіантів дій:   * Перехід до наступного стану * Зарезервувати номер документа * Зареєструвати документ |
| Тип діалога | Вибір одного з варіантів підтвердження виконання дії:   * Запит підтвердження * Запит підтвердження та коментаря |
| Перейти до стана\* | Вибір одного з доступних станів для переходу при виконанні даної дії, які налаштовані у типі документу:   * Новий * Закритий   Примітка: Всі можливі переходи, з одного стану в інший, позначені стрілками на діаграмах життєвих циклів документів. |
| Дозволити для ролей | Налаштування до виконання даної дії по ролям, які активовані (налаштовані) у типі документу |

## 1.4. Процеси

Для кожного типу документів можна налаштувати список виконуваних процесів, що являються елементами BPMN 2.0.

Для додавання процесів необхідно натиснути кнопку «Додати процес» та заповнити поля з форми, що відкрилась (Рис. 1. 10). Поля форми описано в Таблиця 3.

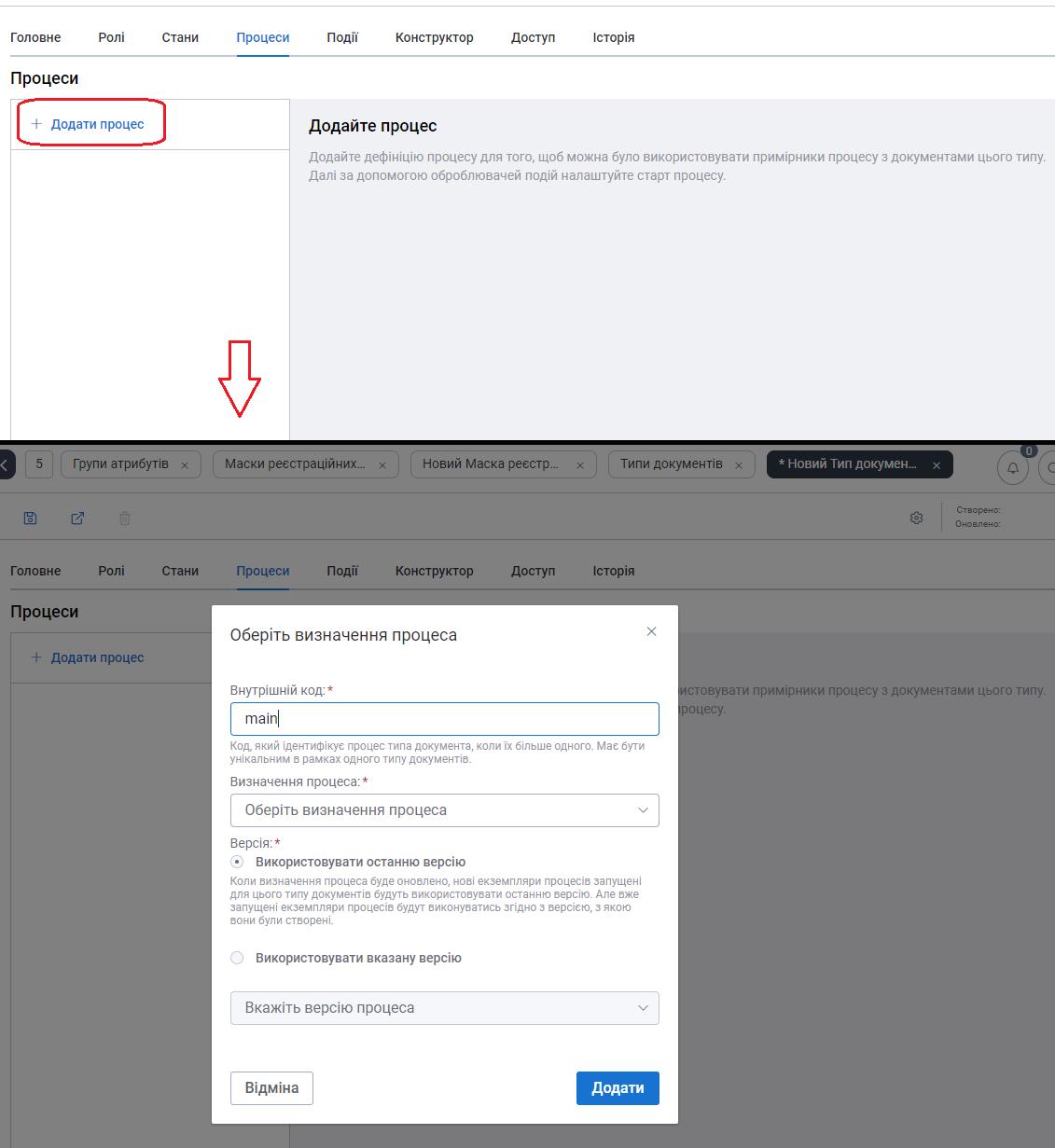


Рис. 1. 10. Параметри додавання процесів

Таблиця 3

| **Назва поля** | **Опис** |
| --- | --- |
| Внутрішній код\* | Унікальний ідентифікатор процесу, що використовується для ідентифікації старту процесу |
| Визначення процесу\* | Вибір з випадаючого списку, необхідної дефініції процесу. |
| Версія\* | Можливість автоматично використовувати останню версію чи зафіксувати конкретну версію даного процесу |

Доданий процес можна редагувати чи видалити, натиснувши на відповідні кнопки, а також перейти до самої дефініції, натиснувши кнопку з назвою процесу (Рис. 1. 11).

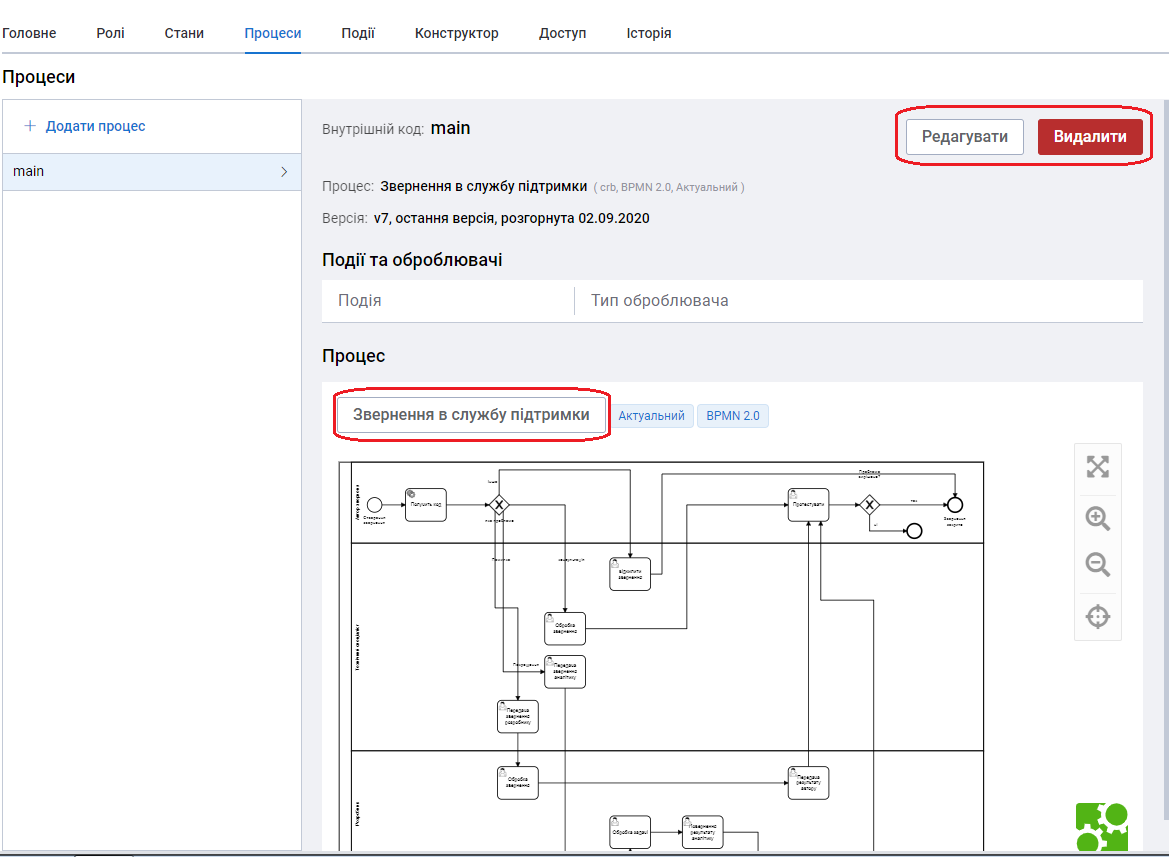


Рис. 1. 11. Зміна/видалення процесу

Створення бізнес-процесів описано в пункті 3.1. Дефініції процесів.

## 1.5. Події

Подія – факт, що активує чи впливає на подальший розвиток одного чи більше процесів. Події ініціюють дії чи являються їх результатом та відносяться до конкретної точки часу.

Настання визначеної події викликає старт визначеного процесу.

Зв'язок визначається за допомогою змінної процесу, яка вказується як частина змінної події в BPM. Для того, щоб передати змінні в процес – необхідно створити оброблювач подій (Рис. 1. 12). Він викликатиметься одразу під час настання події.

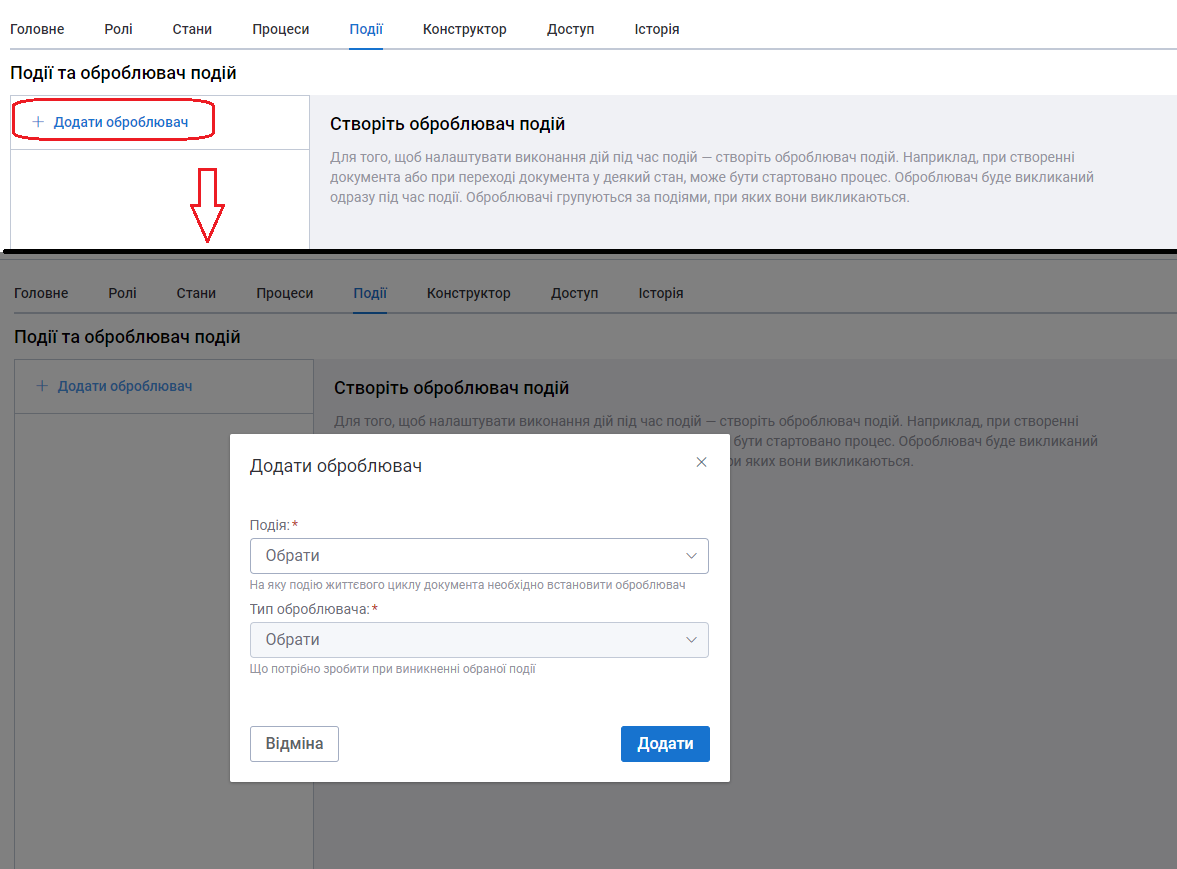


Рис. 1. 12. Параметри додавання події

Поля форми додавання оброблювача описано в Таблиця 4.

Таблиця 4

| **Назва поля** | **Опис** |
| --- | --- |
| Подія\* | Вибір події, на яку встановлюється оброблювач – обирається одна з користувацьких дій, що були створені на етапі додання станів чи автоматична дія – перехід/вихід з можливого стану |
| Тип оброблювача\* | Що необхідно зробити при виникненні обраної події:   * Стартувати процес * Відіслати повідомлення у процес * Скасувати процес * Перейти у стан * Зарезервувати номер документа * Зареєструвати документ   Залежно від обраного типу оброблювача, додається ще одне поле:   * Процес – обрання процесу документа, що має стартувати. При виборі типом «Стартувати процес», «Відіслати повідомлення у процес», «Скасувати процес» * Повідомлення – назва повідомлення, що буде доставлено у процес. При виборі типом «Відіслати повідомлення у процес» * Причина припинення процесу – вказується причина. При виборі типом «Скасувати процес» * Стан – новий/закритий при виборі типом «Перейти у стан» та вказання стану * Ім’я змінної – назва змінної. При виборі типом «Перейти у стан» та вказання змінної |

В процес можна передати змінні, а також отримати їх з процесу за допомогою атрибутів, а також можна додати ролі натиснувши відповідну кнопку «Додати атрибути» чи «Додати ролі» (Рис. 1. 13).

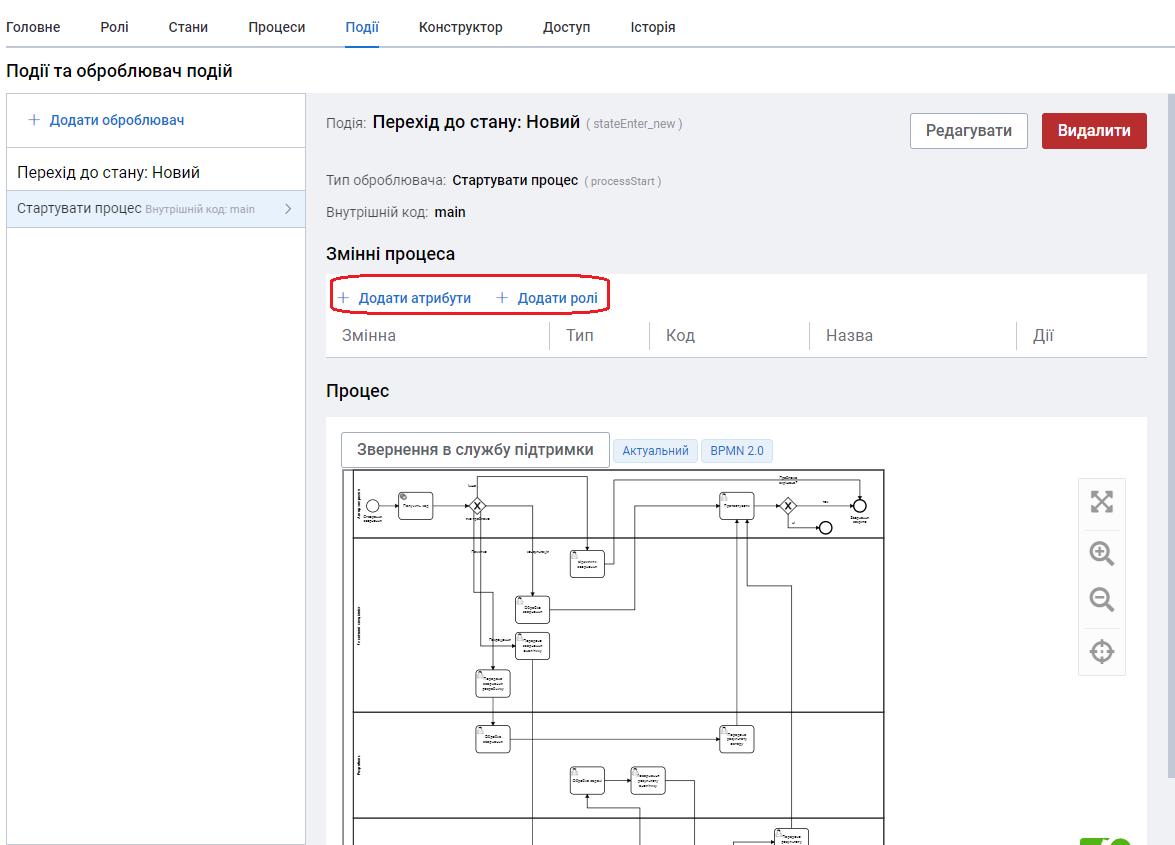


Рис. 1. 13. Додавання атрибуту та/або ролі

## 1.6. Конструктор

Конструктор документів (Рис. 1. 14) дозволяє, за допомогою графічного інтерфейсу, налаштувати форму для заповнення даного типу документа: додавати до форми атрибути для заповнення та налаштовувати вид цих полів, управляти доступом до цих атрибутів, їх відображення для різних ролей.

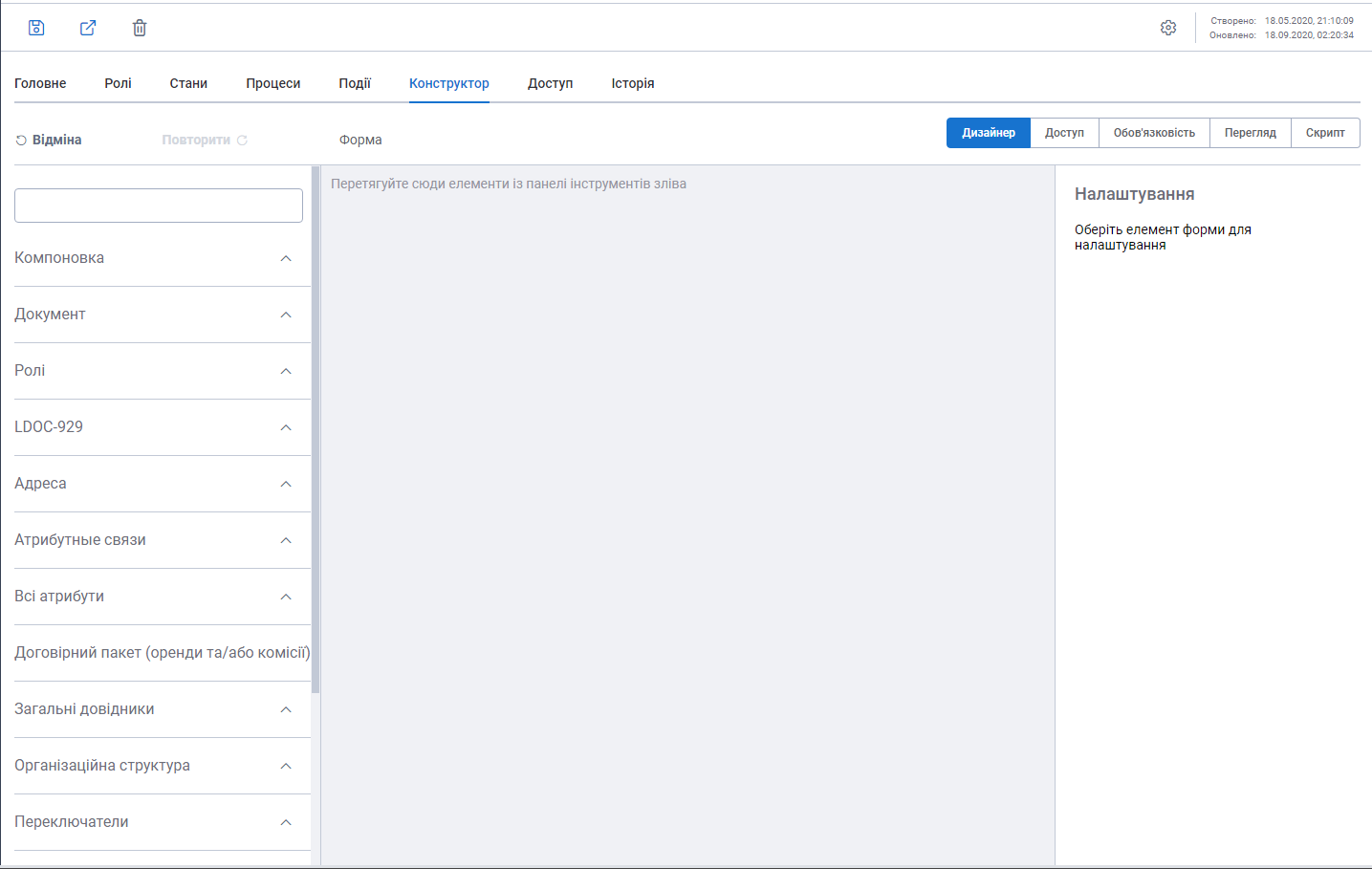


Рис. 1. 14. Конструктор документів

Режим конструктора складається з декількох етапів, що дозволяють налаштувати користувацькі форми за допомогою графічного інтерфейсу:

* Дизайнер
* Доступ
* Обов’язковість
* Перегляд
* Скрипт

### 1.6.1. Дизайнер

В режимі конструктора «Дизайнер», створюється макет майбутньої форми (Рис. 1. 14).

Зліва розміщена панель атрибутів – атрибути, які можна включати в документ (стандартні та додані).

Існує можливість виконати пошук необхідного атрибута за допомогою поля пошуку над панеллю атрибутів (Рис. 1. 14).

За допомогою переміщення атрибутів з панелі атрибутів на форму, здійснюється додавання атрибуту.

Для компоновки атрибутів на формі, використовуються атрибути групи «Компоновка» - вкладки та секція (Рис. 1. 15).

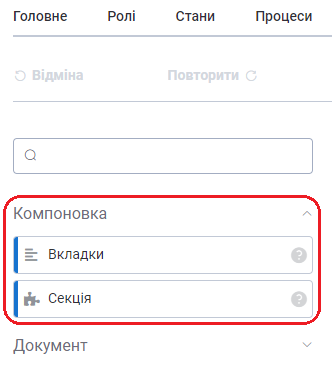


Рис. 1. 15. Атрибути компоновки

Для того, щоб створити вкладку, необхідно додати атрибут «Вкладка» (група «Компоновка») до форми, а після, перетягуючи секції з панелі інструментів зліва чи перетягуючи секції з інших місць на формі, заповнити вкладку. Потім додавати елементи до секції, перетягнувши елементи з панелі інструментів зліва чи перемістивши елемент з інших місць на формі.

Порядок заповнюваних атрибутів в документі також налаштовується за допомогою перетягування атрибутів вгору або вниз в межах форми.

При конструюванні документу можна використовувати атрибути різних груп, щоб не дублювати атрибути.

Примітка: При додавання одного й того ж атрибуту на форму відбувається його дублювання, та в подальшому, при заповненні документу, дублювання значень.

Праворуч, в цьому ж вікні, розміщена панель налаштування форми (Рис. 1. 16). Для кожного атрибуту власний набір налаштувань.

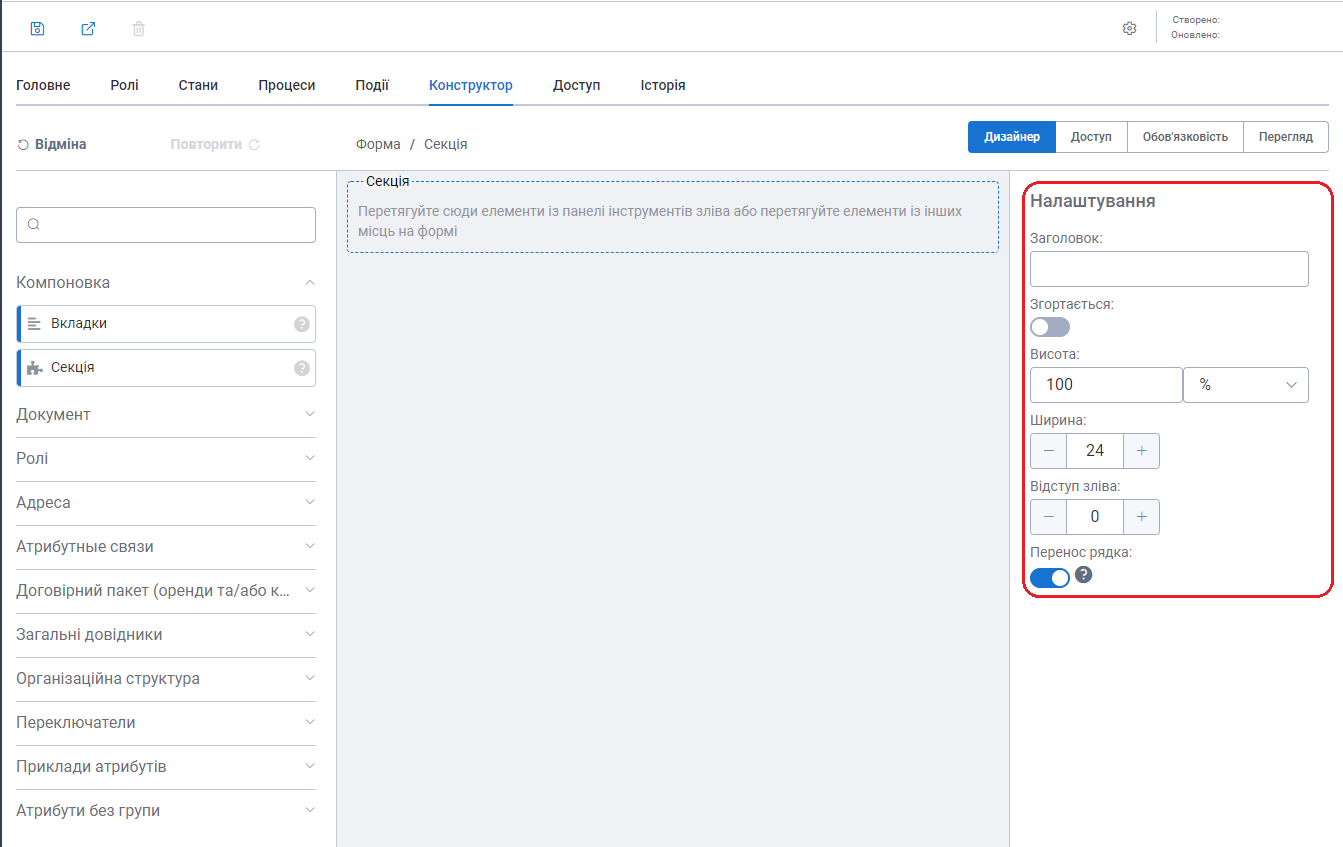


Рис. 1. 16. Панель налаштування атрибуту

Створення атрибутів описано в пункті 6. Атрибути.

Допустимі налаштування:

* Налаштування доступу – можливість заповнення даного поля
* Мітка – відображувана назва поля
* Ширина мітки – допустимий розмір мітки
* Розміщення мітки – вибір варіанту розміщення мітки на формі – згори, справа, зліва.
* Ширина – ширина поля для заповнення. Стандартний розмір форми по ширині – 24 см. За замовчуванням створюється атрибут на всю ширину форми – 24 см.
* Відступ зліва – величина відступу зліва для даного поля
* Висота – налаштування висоти поля – автоматично 1 см або вказати розмір поля у відсотках
* Заповнювач – текст, що відображується в пустому полі.

### 1.6.2. Доступ

Для кожного атрибуту можна налаштувати свій рівень доступу залежно від ролі користувача чи стану документа в режимі конструктора «Доступ» (Рис. 1. 17).

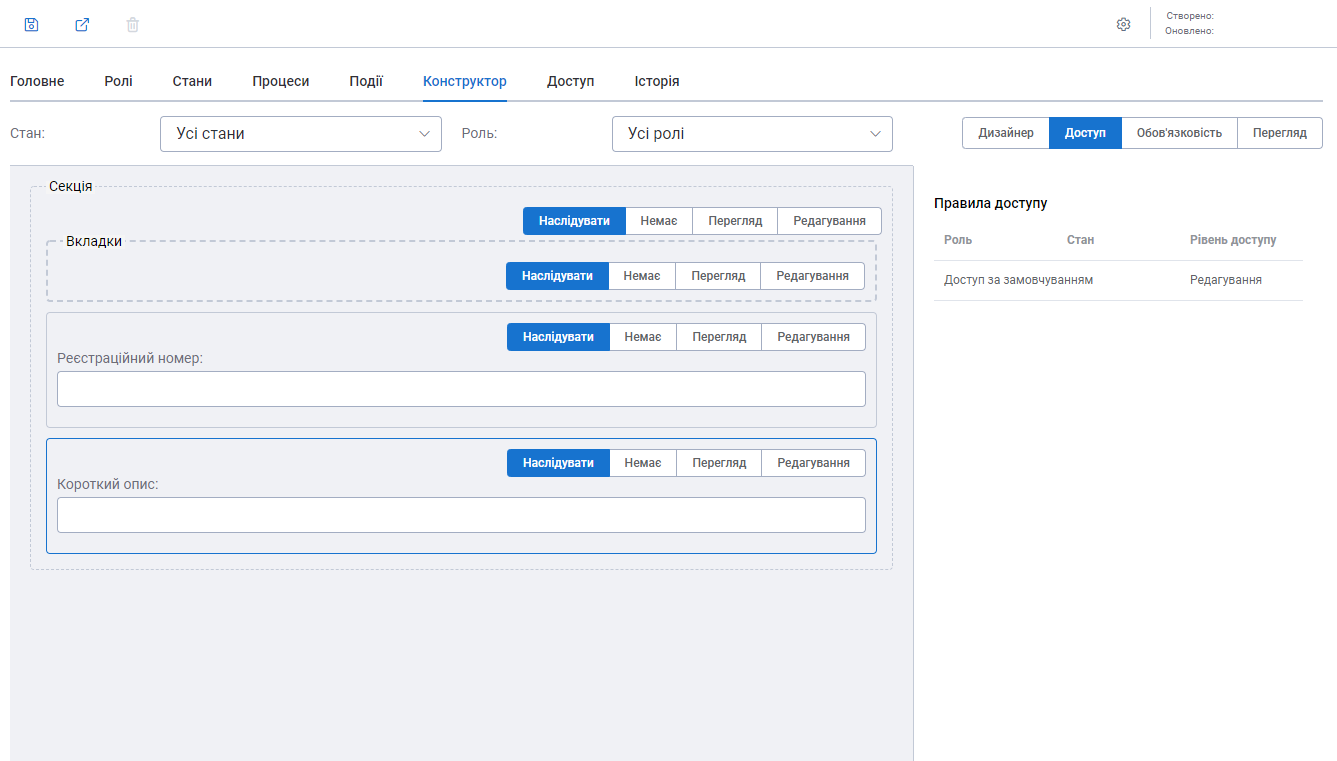


Рис. 1. 17. Рівні доступу форми

Варіанти доступу описано в Таблиця 5.

Таблиця 5

| **Назва кнопки** | **Опис** |
| --- | --- |
| Наслідувати | Доступ до редагування встановлюється за замовчуванням для всіх ролей, що мають права на зміну таких документів |
| Немає | Відсутність доступу до поля, не відображається при перегляді |
| Перегляд | Поле відображається при перегляді без можливості редагування |
| Редагування | Поле доступне для змін |

При обранні атрибуту, праворуч відображається панель «Правила доступу». Панель містить всю інформацію про рівні доступу до цього поля для всіх ролей та станів документа.

### 1.6.3. Обов’язковість

В режимі конструктора «Обов’язковість», можна вказувати правила обов’язковості заповнення для кожного атрибуту форми (Рис. 1. 18). Залежно від розміщення атрибуту у вкладці/секції або поза ними правила розширюватимуться.

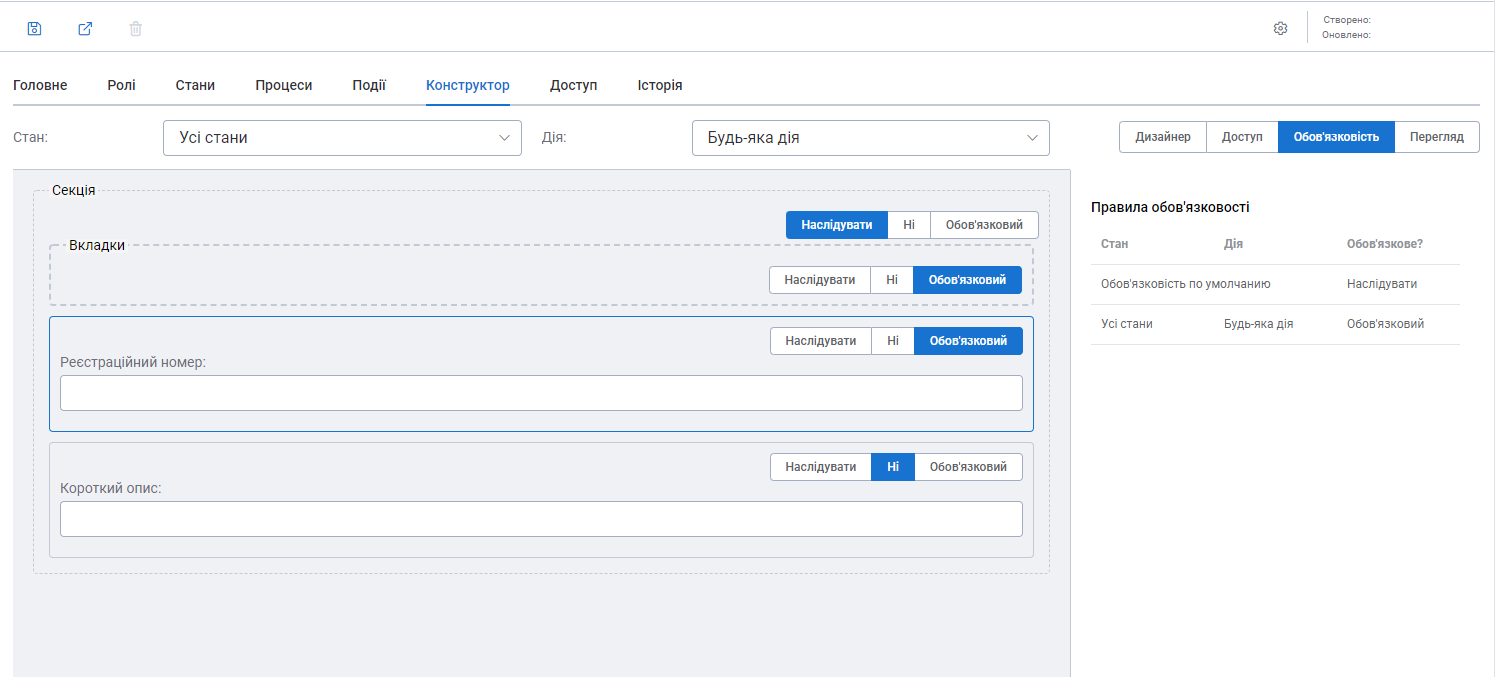


Рис. 1. 18. Правила обов’язковості полів

Варіанти правил для атрибутів описано в Таблиця 6.

Таблиця 6

|  |  |
| --- | --- |
| **Назва кнопки** | **Опис** |
| Наслідувати | Заповнення встановлюється за замовчуванням для всіх ролей, що мають права на зміну таких документів |
| Ні | Атрибут не обов’язковий до заповнення |
| Обов’язковий | Атрибут обов’язковий до заповнення |

### 1.6.4. Перегляд

В режимі конструктора «Перегляд», можна переглядати створеної форми документа та її можливості під визначеними станами та ролями типу документа (Рис. 1. 19).

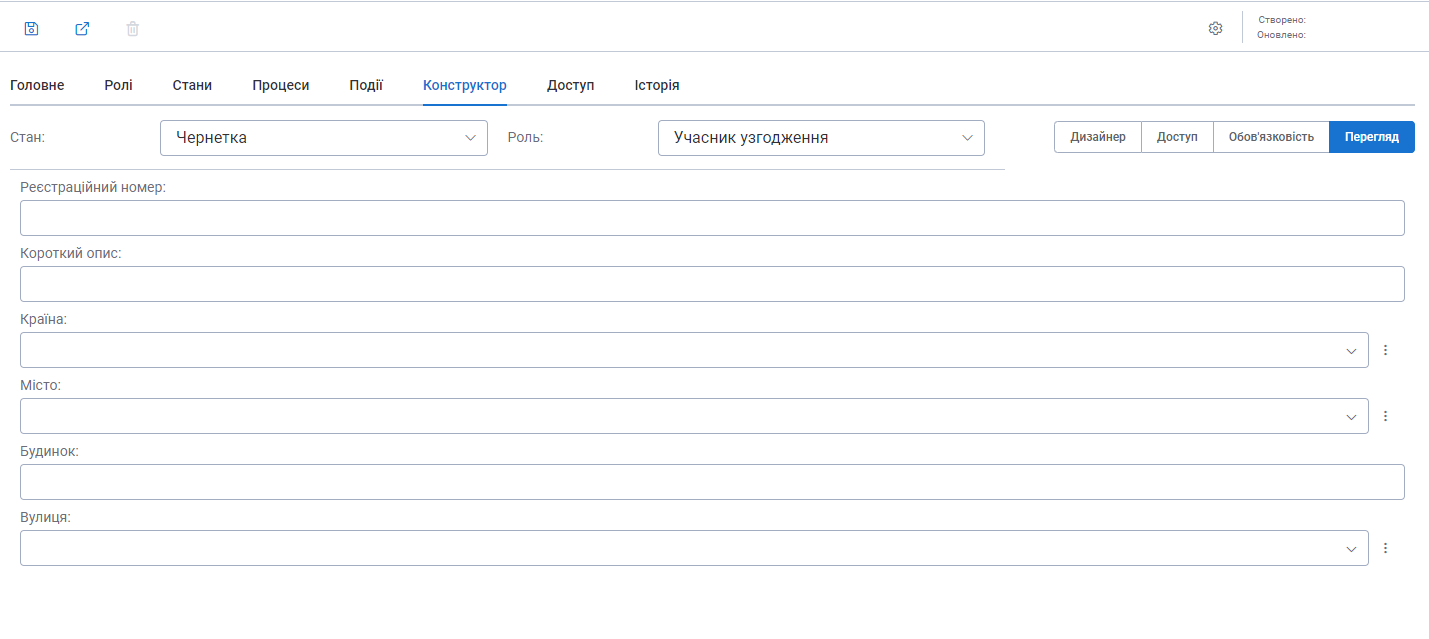


Рис. 1. 19. Перегляд створеного документа

### 1.6.5. Скрипт

В режимі конструктора «Скрипт» (Рис. 1. 20) можливо створити розширення поведінки форми за допомогою налаштованої логіки. Скрипти використовують мову програмування JavaScript та виконуються браузером, як основний код додатку.

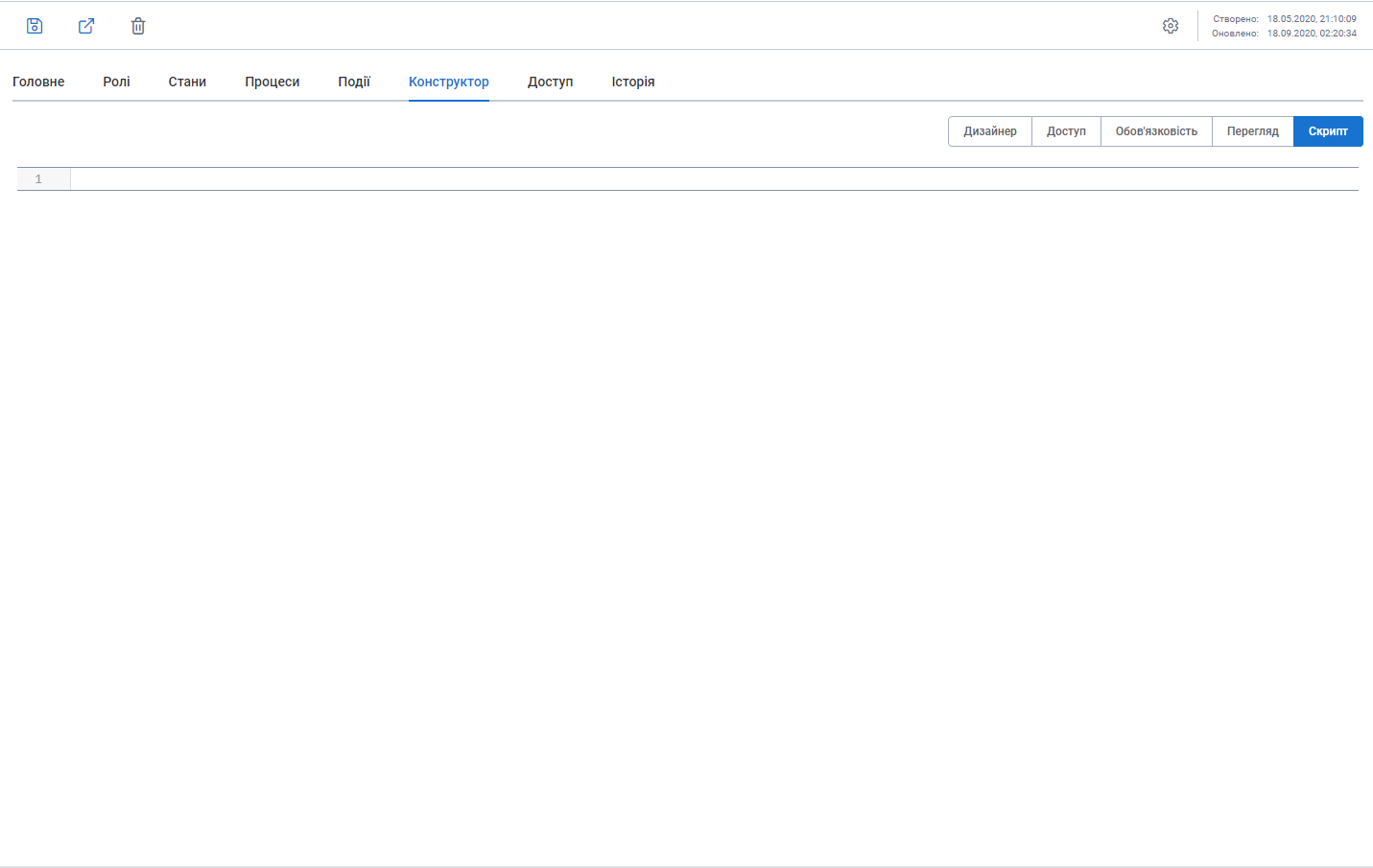


Рис. 1. 20

Для створення скриптів необхідні деякі знання основ JavaScript:

* Основи мови (вирази, інструкції, функції, змінні, змінні та інше)
* Модулі JavaScript
* Асинхронні функції

Окрім того, щоб робити запит до сервера, ознайомтеся з [документацією UnityBase](https://unitybase.info/), особливо з [модулем @ unitybase / ub-pub](https://unitybase.info/api/ubpub-v5/module-@unitybase_ub-pub.html), об’єктом *connection* та класом *ClientRepository*.

***Модуль***

Кожен скрипт форми **повинен** бути модулем CommonJS, тобто він повинен містити такий рядок, як:

module.exports = {

// ...

}

Примітка: Будь-який скрипт записується всередині конструкції:

module.exports = {

}

Взаємодія з формою подійно-орієнтована, та об’єкт, що експортується модулем, повинен містити обробники подій як властивості. Наприклад:

module.exports = {

loaded() {

console.log('Inside "loaded" event handler for the form!')

}

}

Примітка: Ви можете використовувати інший тип синтаксису для визначення функцій, але найбільш зручним для читання та рекомендованим являється перший:

module.exports = {

// Option 1 - the most readable

loaded() {

console.log('Inside "loaded" event handler for the form!')

},

// Old-school

loaded2: function () {

console.log('Inside "loaded" event handler for the form!')

},

// Fancy

loaded3: () => {

console.log('Inside "loaded" event handler for the form!')

}

}

В скрипті може бути лише один module.exports. У випадку, якщо в одному скрипті необхідно задати дві чи більше подій, вони перераховуються в одному module.exports, через кому. Не визначайте декілька module.exports - це не працюватиме.

***Events (Події)***

Назва ключа в об’єкті, що експортується (не плутати з назвою функції, це різні речі), визначає на яку подію функція буде реагувати. Прив’язка події до функції основана на конвенції про іменування.

**Кожен** обробник подій може бути **асинхронним** чи повертати promise (не обов’язково, але може якщо потрібно, в такому випадку код буде очікувати завершення обробки події).

Іноді одна й те сама подія оброблюється декількома обробниками. Наприклад, може бути конкретний обробник подій дій action\_beforeExecute\_action1 та загальний обробник подій action\_beforeExecute. Це не звичайна ситуація, але якщо це відбудеться, загальний обробник подій буде виконаний перед конкретним обробником події.

***Аргумент події***

Кожен обробник подій приймає рівно один аргумент, за узгодженням, якщо обробник не використовує параметр, не вказуйте його, але якщо обробник використовує аргумент події, назвіть його event.

module.exports = {

loaded(event) {

console.log(

'Inside "loaded" event handler for the form for document: %s',

event.document.getNativeAttribute('docNumber')

)

}

}

Ви можете «вимагати» службові сценарії, як і в будь-якому клієнтському модулі UB.

const UB = require('@unitybase/ub-pub')

module.exports = {

// ...

}

Примітка: якщо ви використовуєте зовнішні сценарії, ви несете відповідальність за підтримку сценарію форми при зміні зовнішнього модуля (та іноді вони можуть мати критичні зміни), використовуйте його відповідно.

Можливості аргументу події залежать від форми об’єкту. Наприклад, для форми задач BPM подія міститиме властивість task, а для документу Scriptum міститиме властивість document. Задачі, пов’язані зі Scriptum Document матимуть властивості task та document.

Це також точка налаштування, для великої інтеграції можна розробити додатковий інструмент доступу до об’єктів у вашій користувацькій моделі та зробити його видимим всередині скриптів форми.

Кожна така властивість називається засобом доступу до об’єкту даних оскільки надає засоби для отримання чи встановлення значень для атрибутів відповідного об’єкту.

API об’єктів доступу до об’єктів:

* getAttribute - значення, що повертає, користувацький атрибуту, (з бібліотеки атрибутів, а не атрибут об’єкту).

event.document.getAttribute('color')

* getNativeAttribute - значення, що повертається, атрибуту, що налаштовується. «Native» позначає атрибут сутності, а не атрибут з бібліотеки атрибутів.

event.document.getNativeAttribute('docNumber')

* setAttribute - змінити значення користувацького атрибуту. Щоб очистити значення, встановіть null.

event.document.setAttribute(

'fullName',

event.document.getAttribute('firstName') + ' ' + event.document.getAttribute('lastName')

)

* setNativeAttribute – змінити значення власного атрибуту. Щоб очистити значення, встановіть null. «Native» позначає атрибут сутності.

Деякі атрибути відносяться до словників. Їх значення – числа – ID записів, на які вони вказують. Якщо вам потрібно в скрипті значення зі словника, необхідно написати додатковий код, наприклад:

module.exports = {

async document\_attributeChanged\_organization(event) {

if (event.value) {

const orgCode = await UB.Repository('org\_organization')

.attrs('code')

.where('ID', '=', event.value)

.selectScalar()

console.log('Selected organization with ID=%d, code=%s', event.value, orgCode)

} else {

console.log('Cleaned up "organization" attribute!')

}

}

}

Примітки до прикладу:

* Зверніть увагу, що використовується UB.Repository – це частина UnityBase API, зверніться до [документації UnityBase](https://unitybase.info/api/ubpub-v5/ClientRepository.html) для отримання детальної інформації про це.
* Всі UB.Repository методи вибору являються асинхронними, тому обробник подій також повинен бути асинхронним, і ми «чекатимемо» виклику «selectScalar».
* В цьому прикладі показано, як робити запит словника, такий синтаксис event.document.getAttribute('organization.code')неправильний та не повертає бажане значення.

Деякі засоби доступу до об’єкту можуть надавати додаткові методи. Наприклад, засіб оступу до об’єкту Scriptum Document надає такі методи:

* Властивість state повертає код поточного стану документу

if (event.document.state === 'draft') {

// It's a draft!

}

* userHasRole – функція, що перевіряє чи є у користувача документна роль (не UB), його код.

if (event.document.userHasRole('author')) {

// It's the document author!

}

* getParticipants – функція, що повертає інформацію про ролі учасників документу. Значення функції, що повертається – масив об’єктів наступної структури:

{

ID: 3000012,

unitType: 'ORG',

code: '0123123',

name: 'organization1',

isFixed: true

}

Застосування:

const participantsOfResponsibleRole = event.document.getParticipants('responsible')

* removeParticipant - функція, видаляє будь-якого учасника з ролі документа. Першим аргументом функції являється код ролі, а другим повинен бути orgUnitID, наприклад, staffUnitID.

event.document.removeParticipant('responsible', participantID)

* addParticipant - функція, додає учасника до ролі документа. Першим аргументом функції являється код ролі, а другим повинен бути orgUnitID, наприклад, staffUnitID.

event.document.addParticipant('responsible', participantID)

* getCurrentUserRoles - функція, повертає ролі поточного користувача документа у вигляді масиву рядків. Повертає масив рядків, наприклад:['author', 'manager']

const userRoles = event.document.getCurrentUserRoles()

***Події глобальної форми***

Події ініціалізації форми:

* inited
* loaded

Це дві схожі події, єдина різниця між ними в тому, що initied викликаються лише при першому завантаженні даних форми, а loaded викликаються кожен раз при завантаженні форми (користувач може використовувати кнопку панелі інструментів «Оновити» для перезавантаження форми).

Події збереження форми:

* beforeSave
* saved

Всередині beforeSave можна відмінити збереження:

module.exports = {

beforeSave(event) {

if (!!event.document.getAttribute('hasCategory') && !event.document.getAttribute('categoryID')) {

// "categoryID" must be specified, if "hasCategory" toggle is toggled

event.cancelSave()

}

}

}

***Події дії***

Якщо подія «before» та «after»:

* action\_beforeExecute
* action\_beforeExecute\_<actionCode>
* action\_executed
* action\_executed\_<actionCode>

Подія без суфіксу коду дії буде викликатися для будь-якої події, а події зі суфіксом коду дії будуть викликатися лише для визначених дій.

Всі події мають додаткову властивість – actionCode. Що використовується для універсальних обробників подій дій (без коду дії в назві події).

У обробників «before» також є наступні додаткові методи:

* cancelAction – дозволяє не викликати дію. Використовується для перевірок.
* skipStandardConfirmDialog – виклик цього методу означає «Не викликати жодного підтвердження перед дією». Використовується у випадках, коли всередині обробника викликається інтерфейс користувача та необхідно перекрити стандартний. Скоріше за все необхідно буде використовувати обробник асинхронних подій, що повертає повідомлення підтвердження після того як користувач підтвердить дію в інтерфейсі користувача. Додайте ключове слово async :

const customUI = require('path\_to\_my\_custom\_ui\_module')

module.exports = {

async action\_beforeExecute\_approve(event) {

const confirmResult = await customUI.approveActionDialog()

if (!confirmResult) {

event.cancelAction()

}

}

}

***Атрибути події***

Атрибутні події являються варіаціями самої загальної attributeChanged події:

* <dataObject>\_attributeChanged
* <dataObject>\_nativeAttributeChanged
* <dataObject>\_attributeChanged\_<attributeCode>
* <dataObject>\_nativeAttributeChanged\_<attributeCode>

Як видно з шаблону назви події, кожен варіант подій фільтрує події по об’єкту даних (наприклад завдання чи документ), типу атрибуту (налаштовуваний – з бібліотеки атрибутів чи власний – атрибут сутності) та код атрибуту.

Сам об’єкт події містить такі властивості, що стосуються лише до подій attributeChange:

* dataObject
* attributeCode
* attributeKind
* value
* oldValue – попереднє значення

module.exports = {

document\_attributeChanged(event) {

if (event.attributeCode === 'firstName' || event.attributeCode === 'lastName') {

event.document.setAttribute(

'fullName',

event.document.getAttribute('firstName') + ' ' + event.document.getAttribute('lastName')

)

}

}

}

Даний результат можна досягти за допомогою іншого підходу – замість одного обробника подій для всіх змін атрибутів документа з логікою всередині, що аналізує, який атрибут змінився, тієї ж цілі можна досягти маючи два окремих обробника подій – по одному на атрибут:

module.exports = {

document\_attributeChanged\_firstName(event) {

event.document.setAttribute(

'fullName',

event.value + ' ' + event.document.getAttribute('lastName')

)

},

document\_attributeChanged\_lastName(event) {

event.document.setAttribute(

'fullName',

event.document.getAttribute('firstName') + ' ' + event.value

)

}

}

В останньому варіанті код чистіше. Рекомендовано використовувати саме такий підхід.

***Управління макетом форми під час виконання***

***Зміна заголовку форми***

Можна налаштувати заголовок форми, використовуючи event.form.setTitle метод:

module.exports = {

loaded(event) {

event.form.setTitle(

'2020/' + event.document.getNativeAttribute('docNumber')

)

}

}

***Зміна властивостей елементу форми***

Скрипти форм дозволяють змінювати макет форми під час виконання:

* змінити властивість елементу (вузлів) форми
* змінити елемент форми (вузли), необхідні правила, що його підтримують (форми виконання документів та задач)
* змінити елементи форми (вузли) accessRules для форм, що підтримують його (наприклад форми документів)

Об’єкт події має спеціальну властивість form, що містить спеціальний об’єкт з такими методами:

* setNodeProperty
* setNodeRequiredRule
* setNodeAccessRule

Всі вищезгадані методи приймають числовий nodeId в якості першого артументу, key в якості другого та value – третього.

module.exports = {

document\_attributeChanged\_country(event) {

// Do not allow changing 'city' (nodeId = 10), unless country value is set

event.form.setNodeProperty(

10,

'disabled',

!event.value

)

}

}

***Налагоджувальні скрипти***

Виведення інформації в Console.log – це виведення інформації в консоль браузера: F12, закладка Console

Є декілька способів налагодження скриптів форм. Всі вони вимагають створення документу та його випробування:

* помістіть debugger оператор в скрипти, вони відкриють DevTools в браузері (клавіша F12) та спробуйте:

module.exports = {

document\_attributeChanged\_firstName(event) {

debugger

event.document.setAttribute(

'fullName',

event.value + ' ' + event.document.getAttribute('lastName')

)

}

}

* використайте console.debug

module.exports = {

document\_attributeChanged\_firstName(event) {

console.debug('document\_attributeChanged\_firstName event: %j', event)

event.document.setAttribute(

'fullName',

event.value + ' ' + event.document.getAttribute('lastName')

)

}

}

* використайте контрольні точки: відкрийте DevTools в браузері, знайдіть сценарій, підказку – подивіться під clientRequire/models/cust/public/designedForms, залежно від середовища назви моделі може бути не «cust»

При роботі з довідниками передаються та повертаються id значень довідника. Для переліку значень довідника з множинним вибором – використовуйте квадратні дужки (наприклад [3000000001781, 3000000001748])

***Приклади скриптів***

**Для документів**

Подія: loaded (спрацьовує при відкритті документа)

1. Подія: loaded. Введення значень користувацьких атрибутів в console.log

Введення значень користувацьких атрибутів в консоль браузера при відкритті форми документа

module.exports = {  
loaded(event) {  
console.log('Значення першого атрибуту:',event.document.getAttribute('text001'),' Значення другого атрибуту:',event.document.getAttribute('text002'))  
}  
}

1. Подія: loaded. Введення системних атрибутів в console.log

Введення значень системних атрибутів в консоль при відкритті форми документа

module.exports = {  
loaded(event) {  
console.log('Значення атрибуту:',event.document.getNativeAttribute('docNumber'),event.document.getNativeAttribute('regNumber'),event.document.getNativeAttribute('regDate'),  
event.document.getNativeAttribute('subject'),event.document.getNativeAttribute('authorID'))  
}  
}

1. Подія: loaded. Вставка фіксованого значення в атрибут

module.exports = {  
loaded(event) {  
console.log('Inside "loaded" event handler')  
event.document.setAttribute('text001','Значення атрибуту')  
}  
}

1. Подія: loaded. Вставка значення атрибуту з іншого документу

Копіювання атрибуту text001 в text002 при відкритті форми документа

module.exports = {  
loaded(event) {  
console.log('Inside "loaded" event handler')  
event.document.setAttribute('text002', event.document.getAttribute('text001') )  
}  
}getNativeAttribute

1. Подія: loaded. Вставка фіксованого значення атрибуту типу «Довідник» (валюти)

module.exports = {  
loaded(event) {  
console.log('Inside "loaded" event handler')  
event.document.setAttribute('currencyy',3000000001781)  
}  
}

1. Подія: loaded. Вставка фіксованого значення типу «Довідник з множинним вибором» (валюти)

module.exports = {  
loaded(event) {  
console.log('Inside "loaded" event handler')  
event.document.setAttribute('testmultidict1', [3000000001781, 3000000001748])  
}  
}

Подія: attributeChanged (спрацьовує при редагуванні атрибуту)

1. Подія: attributeChanged\_<attr\_code>. Копіювання одного атрибуту в інший, вихідний атрибут – користувацький

Копіювання атрибуту text001 в text002 при редагуванні атрибуту text001

module.exports = {  
document\_attributeChanged\_text001(event) {  
event.document.setAttribute('text002', event.value)  
}  
}

1. Подія: attributeChanged\_<attr\_code>. Копіювання одного атрибуту в інший, вихідний атрибут – системний

Копіювання одного атрибуту в інший, за умови, що вихідний атрибут – системний

module.exports = {  
document\_nativeAttributeChanged\_subject(event) {  
event.document.setAttribute('text001', event.value)  
}  
}

1. Подія: attributeChanged (без вказування конкретного атрибуту). Копіювання декількох атрибутів в третій атрибут, вихідні атрибути – користувацькі

module.exports = {  
document\_attributeChanged(event) {  
if (event.attributeCode === 'text001' || event.attributeCode === 'text002') {  
event.document.setAttribute('text003', event.document.getAttribute('text001') + ' ' + event.document.getAttribute('text002'))  
}  
}  
}

1. Подія: attributeChanged\_<attr\_code>. Обігравання попереднього приладу за допомогою окремих подій: копіювання декількох атрибутів в третій атрибут

module.exports = {  
document\_attributeChanged\_text001(eventa) {  
eventa.document.setAttribute(  
'text003',  
eventa.value + ' ' + eventa.document.getAttribute('text002')  
)  
},  
document\_attributeChanged\_text002(eventb) {  
eventb.document.setAttribute(  
'text003',  
eventb.document.getAttribute('text001') + ' ' + eventb.value  
)  
}  
}

1. Подія: attributeChanged\_<attr\_code> (oldValue). Такий самий приклад як і в попередньому, але використання oldValue (попереднього значення атрибуту)

module.exports = {  
document\_attributeChanged\_text001(eventa) {  
eventa.document.setAttribute(  
'text003',  
eventa.oldValue + ' ' + eventa.document.getAttribute('text002')  
)  
},

document\_attributeChanged\_text002(eventb) {  
eventb.document.setAttribute(  
'text003',  
eventb.document.getAttribute('text001') + ' ' + eventb.oldValue  
)  
}  
}

1. Подія: nativeAttributeChanged (так само як і вище, але атрибут системний)

module.exports = {  
document\_nativeAttributeChanged(event) {  
if (event.attributeCode === 'subject') {  
event.document.setAttribute(  
'text003',  
event.document.getNativeAttribute('subject') + ' ' + event.document.getNativeAttribute('stateID')  
)  
}  
}  
}

1. 2 події

Дві різні події в одному модулі. Перший записує text001 в text002 при редагуванні, другий записує text002 в text003 при відкритті форми документа. Події через кому

module.exports = {  
document\_attributeChanged\_text001(eventa) {  
eventa.document.setAttribute('text002',eventa.value)  
},  
loaded(eventb) {  
eventb.document.setAttribute('text003', eventb.document.getAttribute('text002') )  
}  
}

**Для задач BPMN**

module.exports = {  
loaded(event) {  
console.log('Значення опису задачі:',event.task.getNativeAttribute('comment'))  
}  
}

## 1.7. Доступ

На вкладці «Доступ» представляється доступ до даного типу документа конкретним штатним одиницям із довідника «Внутрішні підрозділи» чи «Внутрішні організації».

Для додавання необхідно натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» та заповнити організаційні одиниці до форми, що з’явилась (Рис. 1. 21).

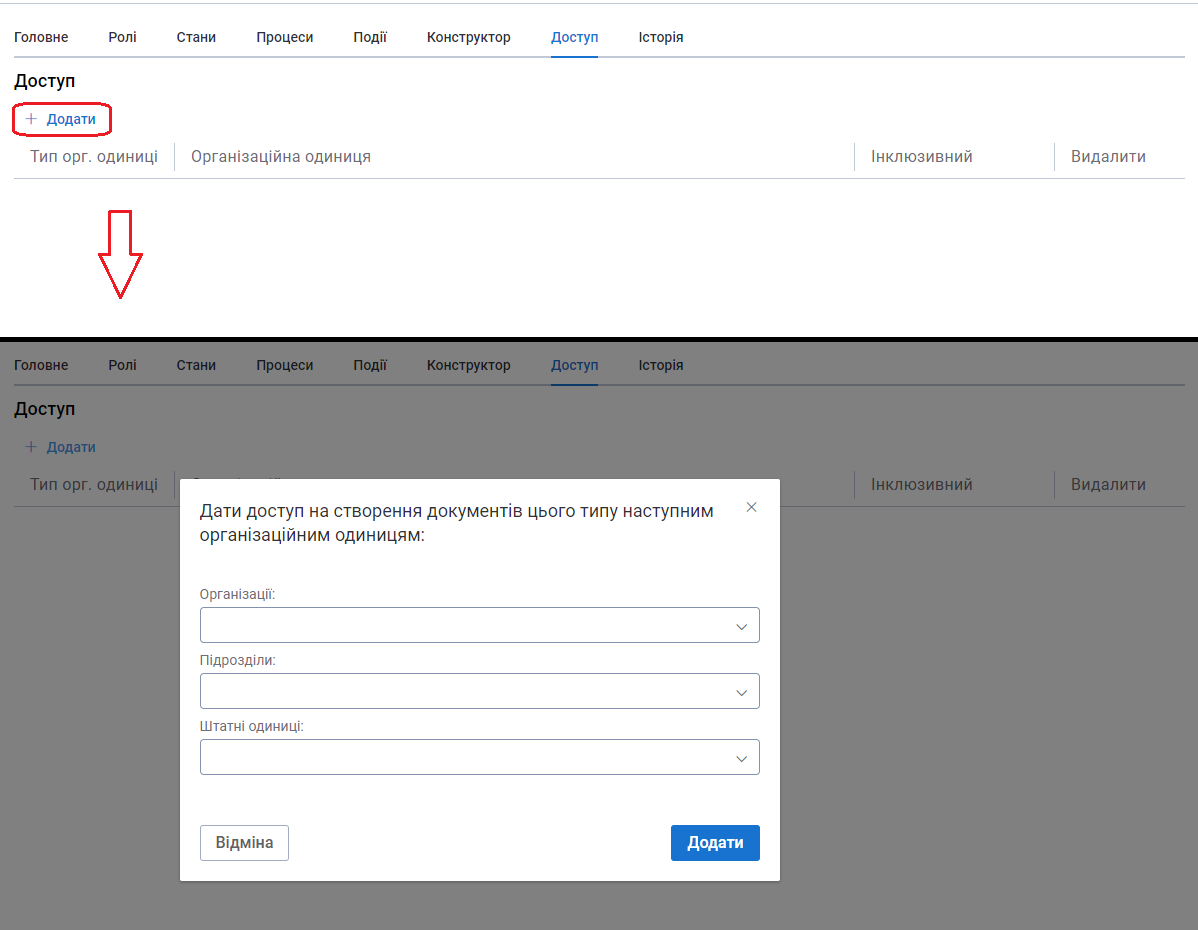


Рис. 1. 21. Налаштування доступу до форми

Опис полів наведено в таблиці Таблиця 7.

Таблиця 7

|  |  |
| --- | --- |
| **Назва поля** | **Опис** |
| Організації | Організації, що мають доступ до документа. При вказанні організації (ій), всі її учасники матимуть доступ до документа. |
| Підрозділи | Підрозділи, що мають доступ до документа. При вказанні підрозділу (ів), всі його учасники матимуть доступ до документа. |
| Штатні одиниці | Штатні одиниці, що мають доступ до документа. При вказанні штатної одиниці (ць), вони матимуть доступ до документа. |

Можна додавати організаційні одиниці до всіх полів одночасно або до окремого поля. До одного поля можна додавати декілька організаційних одиниць.

При доданні нової організаційної одиниці, до блоку «Доступ» записується тип організаційної одиниці, її ім’я, інклюзивність (перемикач передбачає, що всі підзвітні організаційній одиниці елементи, будуть автоматично мати доступ. Якщо перемикач вимкнути, то доступ буде надано організаційній одиниці без підзвітних їй елементам) та кнопка з можливість видалення (Рис. 1. 22).

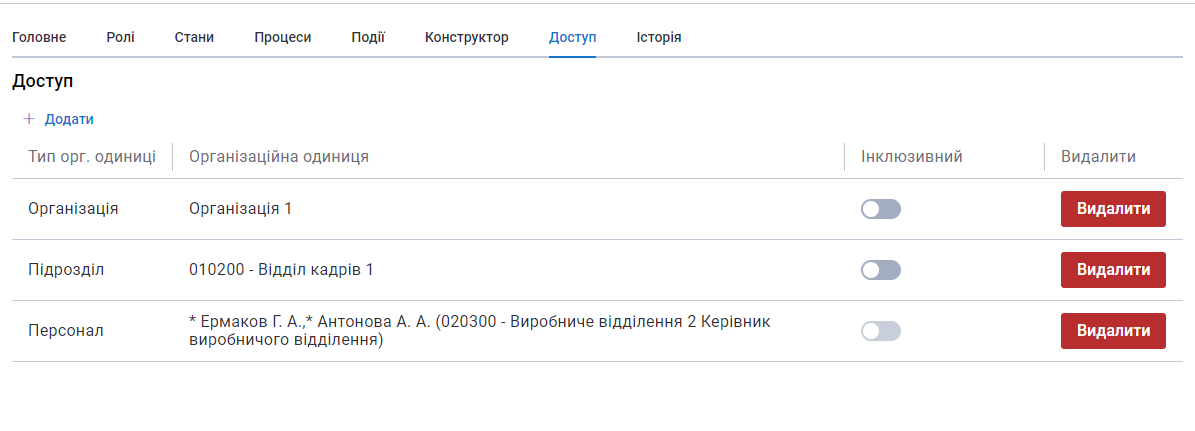


Рис. 1. 22. Реєстр організаційних одиниць доступом до документа

## 1.8. Історія

Історія – вкладка, що дозволяє фіксувати з можливістю подальшого перегляду користувачами та адміністратором подій по конкретному Типу документа (Рис. 1. 23).Системою фіксуються події, що відбуваються з конкретним Типом документа протягом його життєвого циклу.

Адміністратор може налаштовувати перелік типів подій, що будуть фіксуватися Системою.

При фіксації факту виконання завдань в атрибуті «Історія документа» (атрибут групи «Документ») відображатиметься завдання, дата та користувач який його виконав.

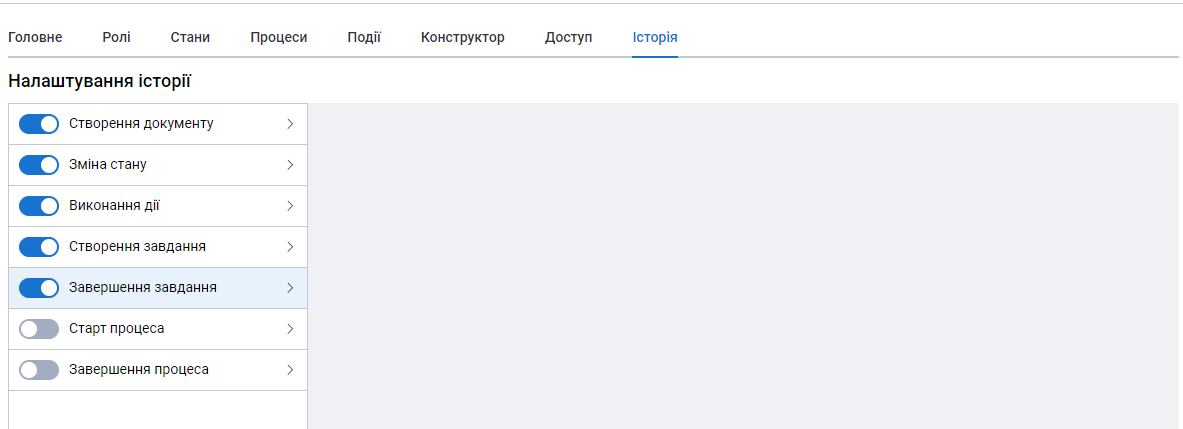


Рис. 1. 23. Налаштування історії типу документа

Для налаштування історії необхідно на вкладці «Історія» зафіксувати перемикач на необхідній(их) з доступних подій:

* Створення документа
* Зміна стану
* Виконання дії
* Створення завдання
* Завершення завдання
* Старт процесу
* Завершення процесу

# 2. Налаштування

## 2.1. Довідник ролей

Довідник ролей – перелік Ролей, що знаходяться в Системі та можуть брати участь в процесі обробки документів визначеного Типу документа (вкладка «Ролі» Типу документа).

Кожна роль може використовуватися лише один раз в одному Типі документа.

На кожен тип документу, окрім стандартних, можна створити власний перелік ролей. Для цього необхідно відкрити робочий стіл «Типи документів», обрати ярлик «Налаштування», далі «Довідник ролей» та натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» (Рис. 2. 1), у формі, що відкрилося, заповнити дані (Рис. 2. 2). Поля форми описано в Таблиця 8.

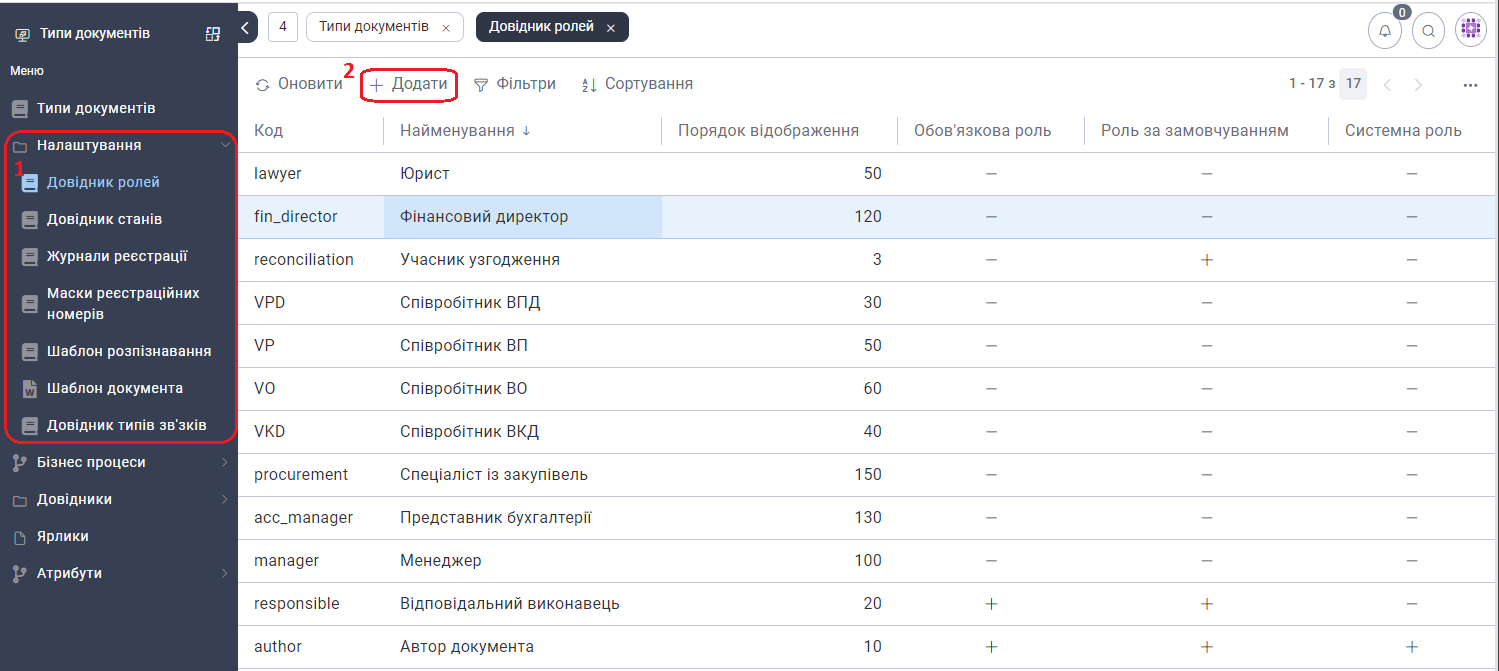


Рис. 2. 1. Створення додаткових ролей

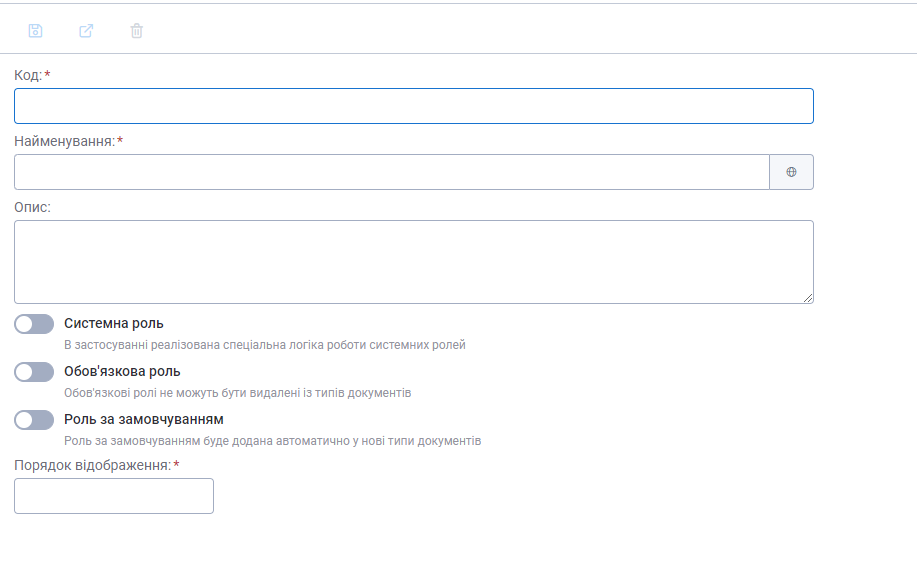


Рис. 2. 2. Параметри створення додаткових ролей

Таблиця 8

|  |  |
| --- | --- |
| **Назва поля** | **Опис** |
| Код\* | Унікальний ідентифікатор даної ролі |
| Найменування\* | Відображувана назва даної ролі |
| Опис | Короткий опис даної ролі |
| Системна роль | Перемикач для вказання ознаки «Системної» ролі |
| Обов’язкова роль | Перемикач для вказання обов’язковості ролі |
| Роль за замовчуванням | Роль за замовчуванням буде відображатись у всіх нових типах документів |
| Порядок відображення\* | Номер порядку відображення ролі в налаштуваннях ролей нових типів документів |

Адміністратор може створювати нові, редагувати існуючі та видаляти ролі, що не використовуються.

Редагування поля «Код» можливе лише для ролей, що не використовуються в жодному з існуючих в Системі Типів документів.

Видалення доступне лише для ролей, що не використовуються в жодному з існуючих в Системі Типів документів.

В довіднику ролей Адміністратор може:

* Створювати нову роль натиснувши кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати».
* Здійснювати сортування та фільтрацію змістовного в реєстрі використовуючи кнопки C:\Users\daryna.karapuzova\Desktop\кк.png «Фільтр» та «Сортування».
* Здійснювати примусове оновлення змістовного в реєстрі C:\Users\daryna.karapuzova\Desktop\1.png
* Здійснювати відкриття форми для редагування/перегляду ролі подвійним кліком на обрану роль або викликавши контекстне меню для відповідної ролі та обравши дію «Редагувати» (Рис. 2. 3)
* Здійснювати копіювання ролі викликавши контекстне меню для відповідної ролі та обравши дію «Копіювати» (Рис. 2. 3). Під час виконання дії відбувається копіювання ролі та відкриття у новій вкладці для можливості редагування та створення нової ролі.
* Здійснювати видалення ролі викликавши контекстне меню для відповідної ролі та обравши дію «Видалити» (Рис. 2. 3).
* Переглядати/копіювати посилання обраної ролі викликавши контекстне меню для відповідної ролі та обравши дію «Посилання» (Рис. 2. 3).
* Переглядати аудит обраної ролі викликавши контекстне меню для відповідної ролі та обравши дію «Аудит» (Рис. 2. 3).
* Переглядати деталі обраної ролі викликавши контекстне меню для відповідної ролі та обравши дію «Деталі» (Рис. 2. 3).

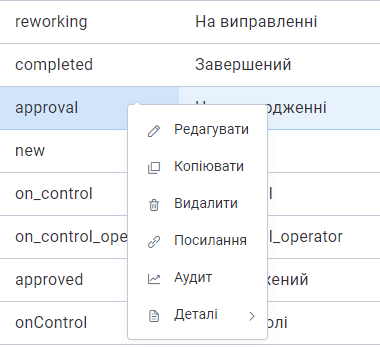


Рис. 2. 3. Контекстне меню реєстру «Довідник ролей»

## 2.2. Довідник станів

На кожен тип документу можна створити перелік станів. Для цього необхідно відкрити робочий стіл «Типи документів», обрати ярлик «Налаштування» далі «Довідник станів» та та натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» (Рис. 2. 4), у формі, що відкрилось, заповнити дані (Рис. 2. 5). Поля форми описано в Таблиця 9.

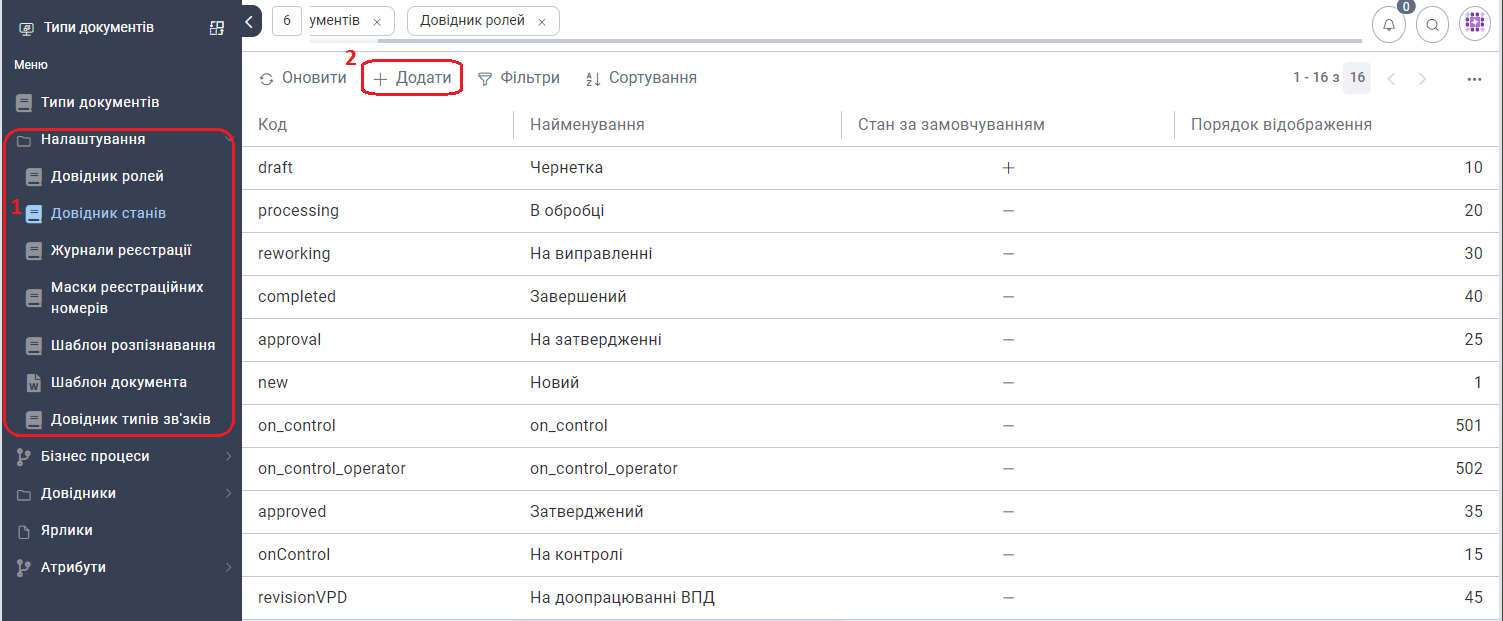


Рис. 2. 4. Додавання нового стану

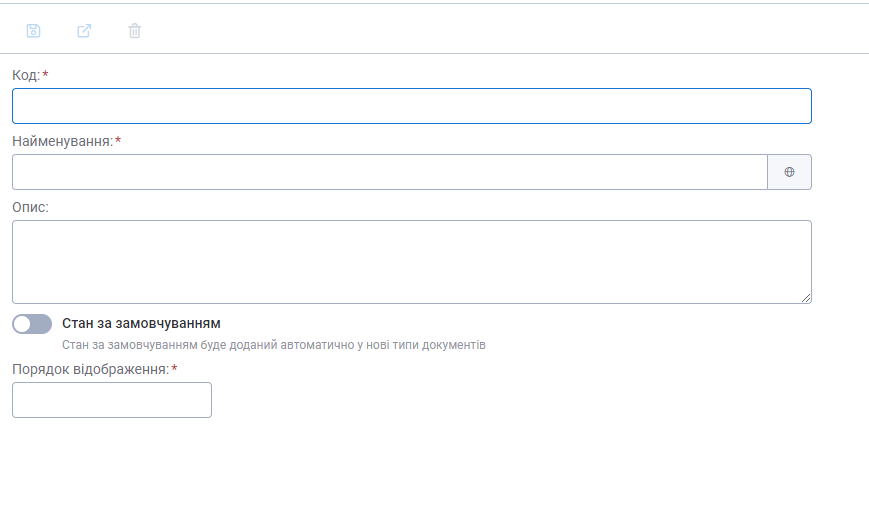


Рис. 2. 5. Параметри створення стану

Таблиця 9

| **Назва поля** | **Опис** |
| --- | --- |
| Код\* | Унікальний ідентифікатор даного стану |
| Найменування\* | Відображувана назва даного стану |
| Опис | Короткий опис даного стану |
| Стан за замовчуванням | Стан за замовчуванням буде відображатись у всіх нових типах документів |
| Порядок відображення\* | Номер порядку відображення стану в налаштуваннях станів нових типів документів |

Адміністратор може створювати нові, редагувати існуючі та видаляти стани, що не використовуються.

Редагування поля «Код» можливе лише для станів, що не використовуються в жодному з існуючих в Системі Типів документів.

Видалення доступне лише для станів, що не використовуються в жодному з існуючих в Системі Типів документів.

В довіднику станів Адміністратор може виконувати аналогічні дії, що й і в довіднику ролей (Рис. 2. 3).

## 2.3. Журнал реєстрації

Система здійснює реєстрацію Документа у відповідному журналі при виконанні ручної або автоматичної реєстрації документа. Для цього використовуються Журнал реєстрації документів та Маски реєстраційних номерів для журналу реєстрації.

Журнал реєстрації призначений для визначення загальних правил реєстрації документів, зокрема реєстраційної маски, що використовується при реєстрації документів в журналі.

Щоб створити журнал необхідно в панелі навігації робочого столу «Типи документів» обрати ярлик «Налаштування», далі «Журнал реєстрації» та натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» (Рис. 2. 6) та у вкладці, що відкрилась, заповнити поля (Рис. 2. 7), опис полів наведено в Таблиця 10.

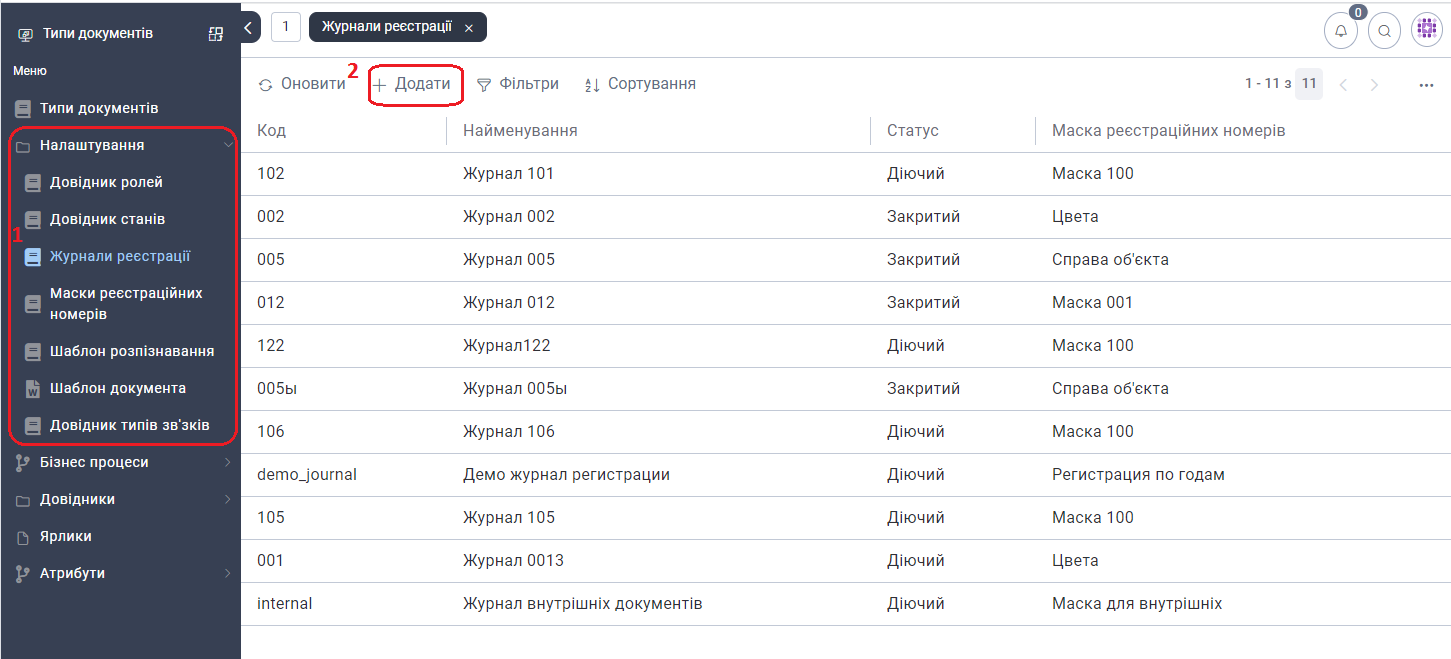


Рис. 2. 6. Створення журналу реєстрації

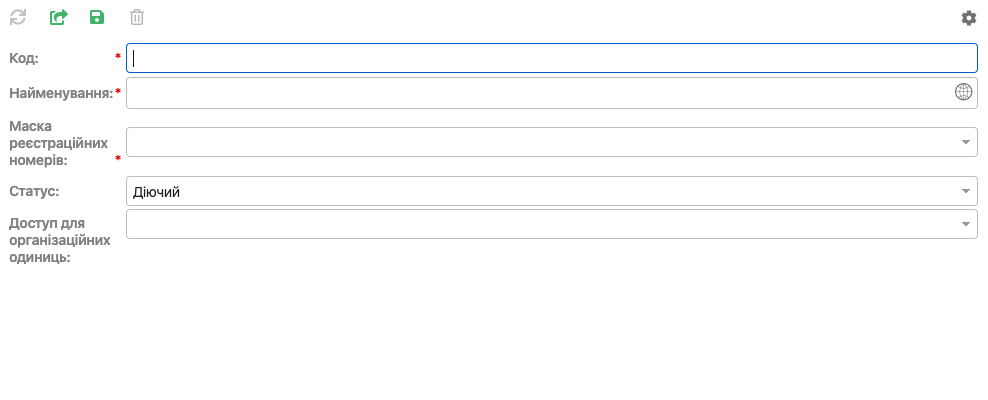


Рис. 2. 7. Параметри налаштування журналу реєстрації

Таблиця 10

| **Назва поля** | **Опис** |
| --- | --- |
| Код\* | Унікальний ідентифікатор журналу |
| Найменування\* | Відображувана назва журналу |
| Маска реєстраційних номерів\* | Зберігає формули, за якими формується реєстраційний номер для документів, зареєстрованих в журналах (створення маски описано в 4. Маски реєстраційних номерів) |
| Статус | Статус журналу. Можливі варіанти:   * Діючий * Закритий |
| Доступ для організаційних одиниць | Перелік організаційних одиниць, які мають доступ до журналу реєстрацій |

Адміністратор може створювати нові, редагувати існуючі та видаляти журнали, що не використовуються.

Редагування поля «Код» можливе лише для станів, що не використовуються в жодному з існуючих в Системі Типів документів.

Видалення доступне лише для журналів, що не пов’язані з жодним з існуючих в Системі Типом документів.

В довіднику журналів реєстрації Адміністратор може виконувати аналогічні дії, що й і в довіднику ролей (Рис. 2. 3).

## 2.4. Маски реєстраційних номерів

Маска реєстраційного номеру призначена для збереження формул, за якими формуватиметься реєстраційний номер для документів, зареєстрованих в журналах.

Щоб створити маску реєстраційних номерів, необхідно в панелі навігації «Типи документів» обрати ярлик «Налаштування», далі «Маски реєстраційних номерів» та натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» (Рис. 2. 8).

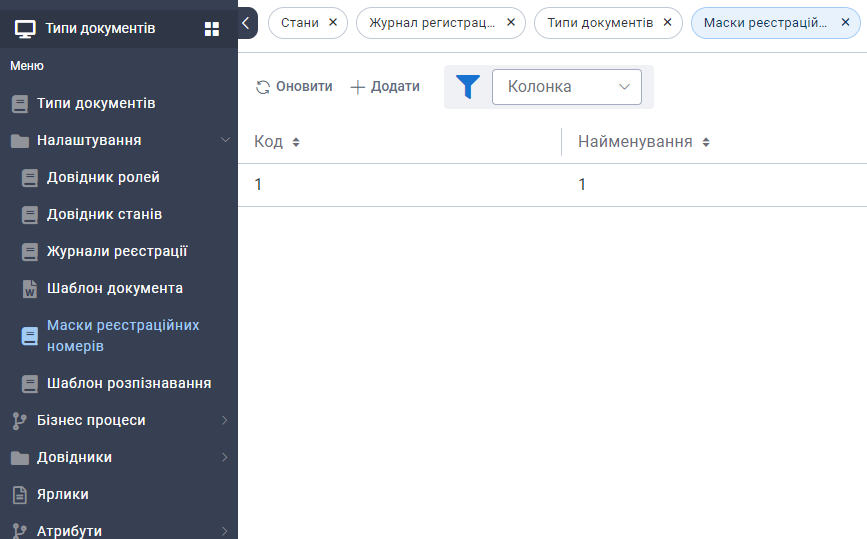


Рис. 2. 8. Створення маски реєстраційних номерів

У вкладці, що відкрилась, заповнити поля (Рис. 2. 9), опис полів наведено в Таблиця 11.

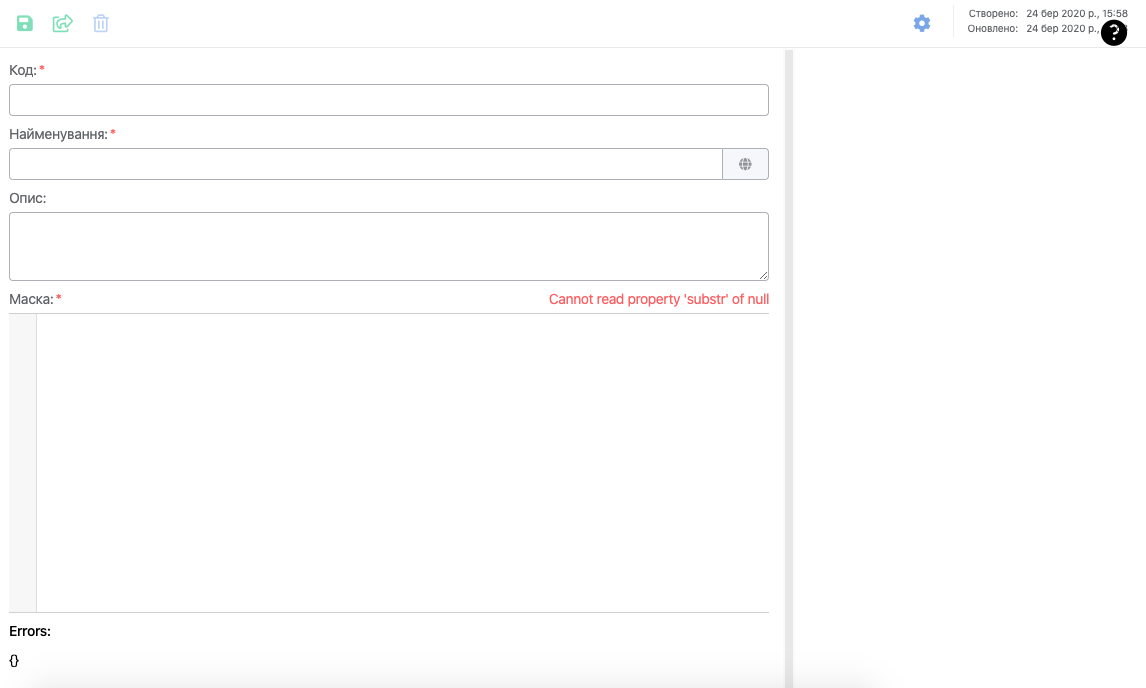


Рис. 2. 9. Параметри створення маски реєстраційних номерів

Таблиця 11

| **Назва поля** | **Опис** |
| --- | --- |
| Код\* | Унікальний ідентифікатор маски |
| Найменування | Відображувана назва даної маски |
| Опис | Короткий опис даного маски |
| Маска\* | Формула формування маски, що може містити наступні елементи:   * Лічильник Журналу реєстрації * Символьні, строкові та числові константи (переносяться в реєстраційний номер в тому вигляді, в якому вказані в масці) * Атрибути документів (системні та користувацькі) з переносом в реєстраційний номер значення атрибуту * Дати (поточна дата, дата із заданим зміщенням від поточної дати, поточна дата в різних форматах) * Арифметичні операції над числовими атрибутами (додавання, віднімання, множення, ділення (з вказанням кількості знаків після комиЮ що має відображатися в реєстраційному номері) на константу чи інший числовий атрибут) * Рядкові операції над рядковими атрибутами (базові рядкові функції SQL – виділення підрядка, конкатенація, переведення у верхній/нижній регістр( * Вибір одного з варіантів по значенню ключа (в якості ключа виступає атрибут документа, як варіантів константи, так і значення атрибутів документа) |

Адміністратор може створювати нові, редагувати існуючі та видаляти маски, що не використовуються.

Редагування поля «Код» можливе лише для масок, що не пов’язані з жодним з існуючих в Системі журналом.

Видалення доступне лише для масок, що не пов’язані з жодним з існуючих в Системі журналом.

В довіднику масок Адміністратор може виконувати аналогічні дії, що й і в довіднику ролей (Рис. 2. 3).

Реєстрація документів у журналі

Користувач може обирати журнал в якому реєструватиметься Документ.

Системою є обмеження щодо вибору користувачем журналу реєстрації:

* Користувач має доступ до журналу реєстрації (сам користувач або підрозділи користувача, що вказано в доступі журналу реєстрації)
* Журнал реєстрації має статус «Діючий».

При запуску стандартного процесу реєстрації (в ручному або автоматичному режимі) Система здійснює заповнення атрибутів документа – «Реєстраційний номер» (відповідно до маски реєстраційного номера журналу) та «Дата реєстрації» (поточна дата)

## 2.5. Шаблон розпізнавання

Шаблон розпізнавання (набір шаблонів) налаштовується для розпізнавання атрибутів документа.

Щоб створити шаблон розпізнавання, необхідно в панелі навігації «Типи документів» далі обрати ярлик «Налаштування» далі обрати «Шаблон розпізнавання» та натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» та у вкладці, що відкрилась, заповнити поля (Рис. 2. 10), опис полів наведено в Таблиця 12.

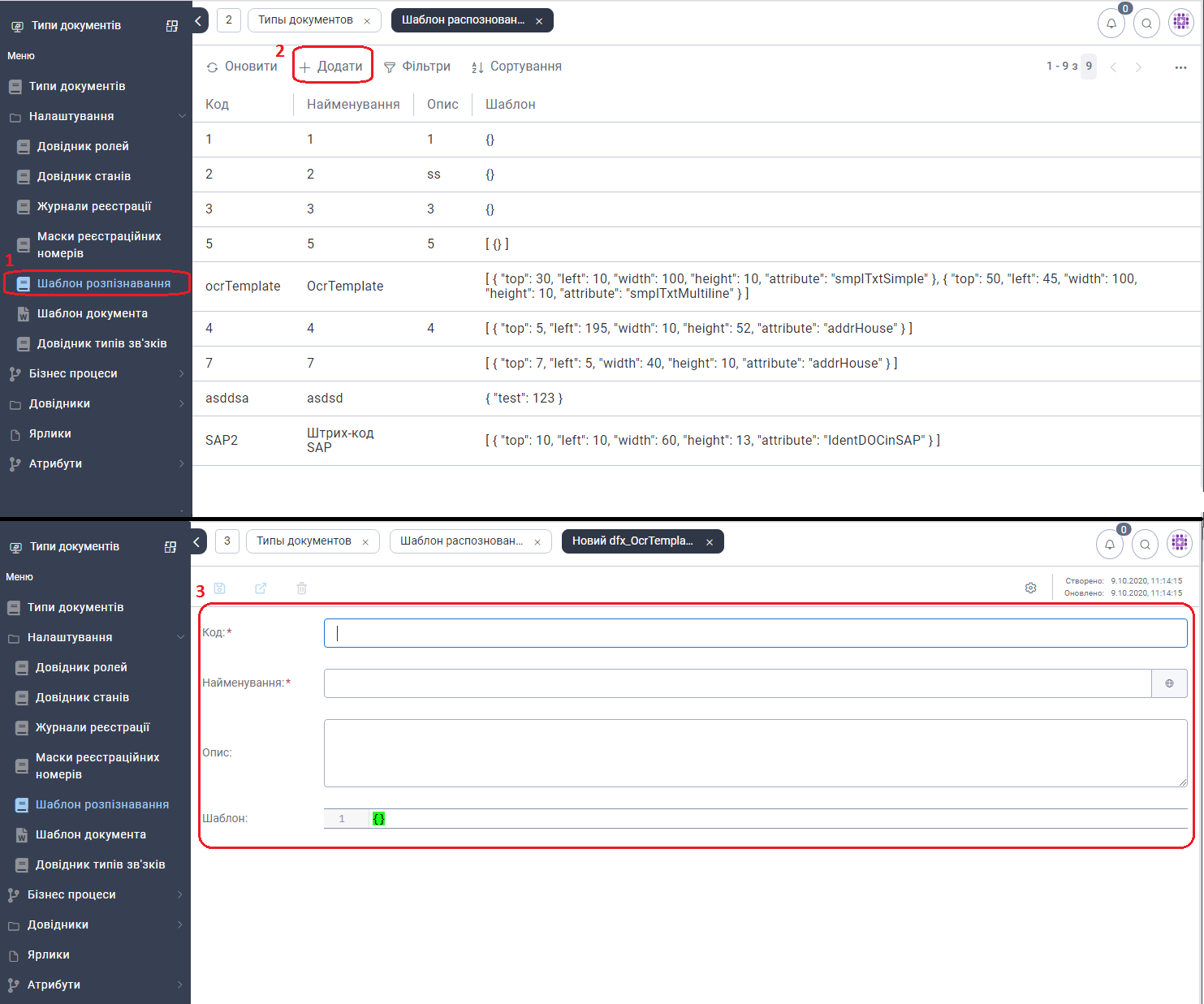


Рис. 2. 10. Додавання шаблону розпізнавання

Таблиця 12

| **Назва поля** | **Опис** |
| --- | --- |
| Код\* | Унікальний ідентифікатор шаблону |
| Найменування\* | Відображувана назва шаблону |
| Опис | Короткий опис шаблону |
| Шаблон | Шаблон розпізнавання в форматі JSON |

Адміністратор може створювати нові, редагувати або видаляти існуючі шаблони. Редагувати можна шаблон, якщо він ще не використовується в Типах документів

Поле «Опис» можна редагувати без обмежень.

Видаляти можна лише шаблон, що не використовується жодним типом документів.

Шаблон створюється без прив’язки до конкретного Типу документа. В самому шаблоні відбувається прив’язка області розпізнавання з конкретним атрибутом документа з переліку атрибутів, що знаходяться в Системі.

Для кожного атрибута в шаблоні є наступний набір налаштувань (Рис. 2. 11):

{

  attribute,          // Пов’язаний атрибут

  areaPage,         // Сторінка файлу/скану на який розміщено

  upperLeftX,     // Координата Х верхнього лівого кута області розпізнавання

  upperLeftY,     // Координата Y верхнього лівого кута області розпізнавання

  width,              // Ширина області розпізнавання

  height,              // Висота області розпізнавання

  dateConvertationTemplate,  // Шаблон перетворення дати

  byDefault,       // Значення, якими атрибут заповнюється за замовчуванням

}

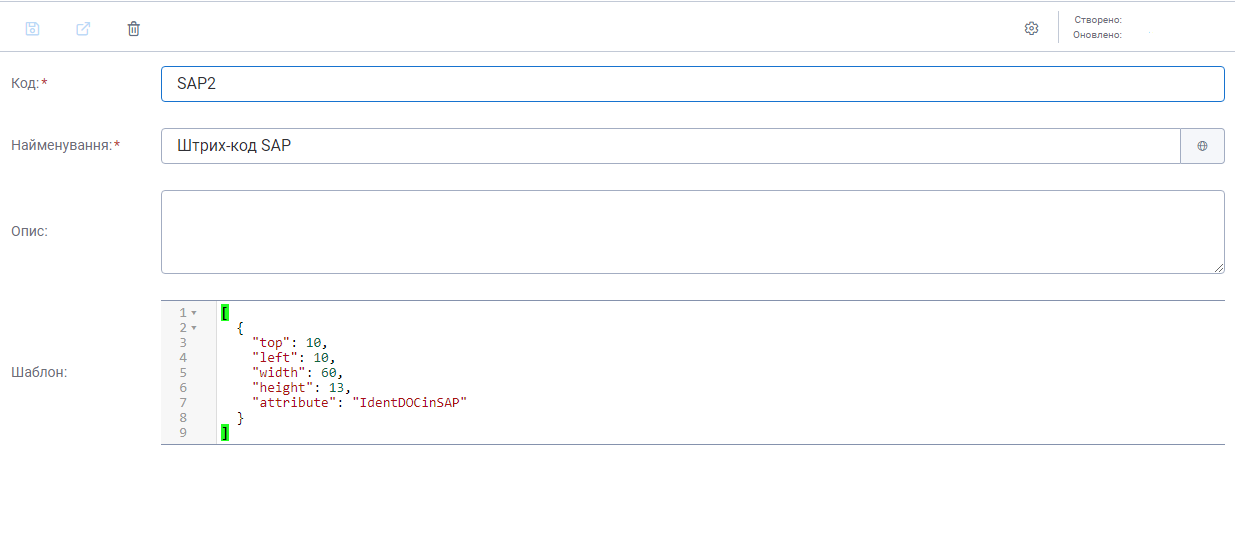


Рис. 2. 11. Приклад налаштування

В довіднику шаблонів розпізнавання Адміністратор може виконувати аналогічні дії, що й і в довіднику ролей (Рис. 2. 3).

## 2.6. Шаблон документа

Шаблон документа призначений для генерації образу документа на основі визначеного шаблону.

Щоб створити шаблон документа, необхідно в панелі навігації робочого столу «Типи документів» обрати ярлик «Налаштування» далі «Шаблон документа» та натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» та у вкладці, що відкрилась, заповнити поля (Рис. 2. 12), опис полів наведено в Таблиця 13.

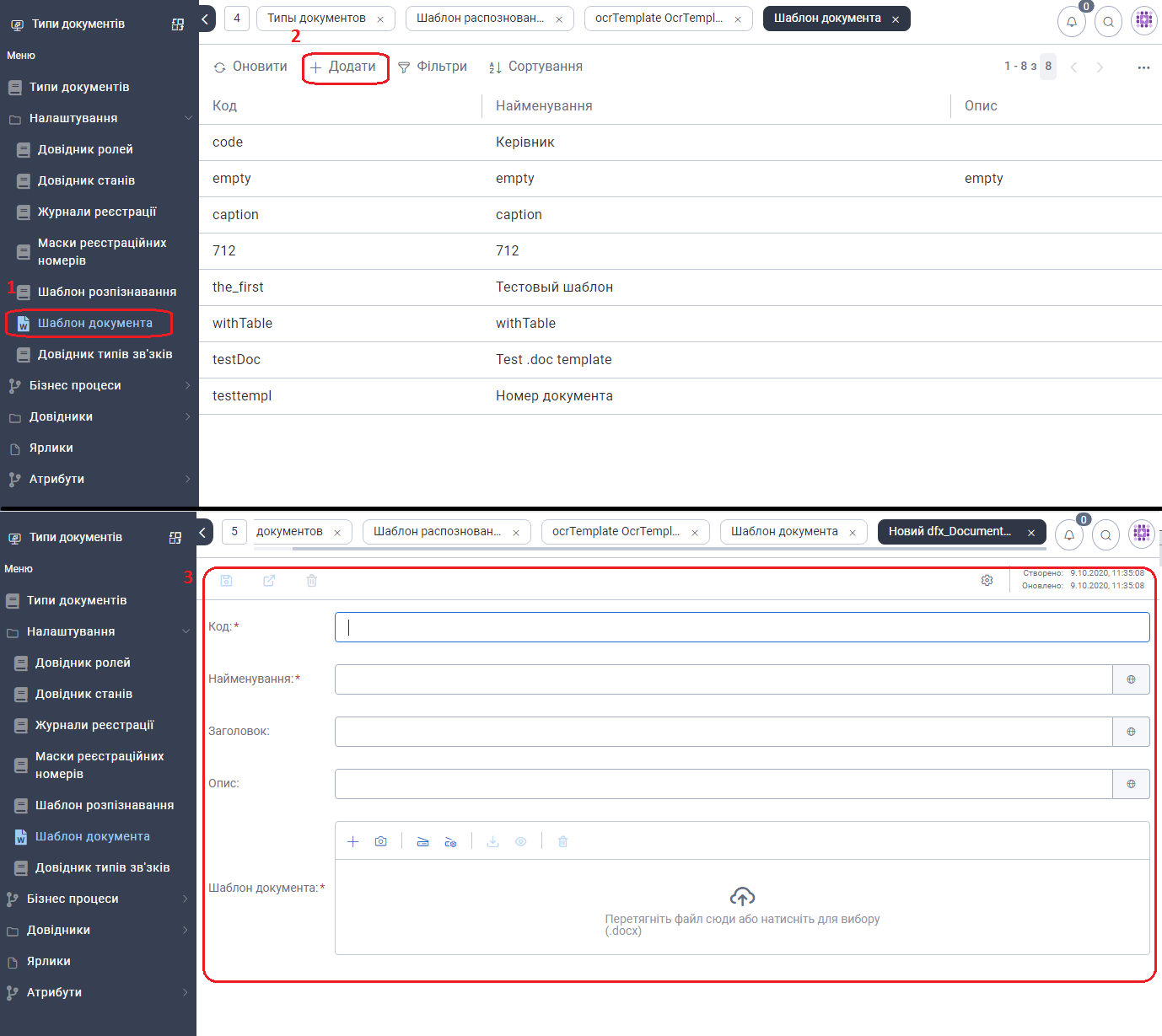


Рис. 2. 12. Створення шаблону документа

Таблиця 13

| **Назва поля** | **Опис** |
| --- | --- |
| Код\* | Унікальний ідентифікатор шаблону |
| Найменування\* | Відображувана назва шаблону |
| Заголовок | Заголовок для шаблону |
| Опис | Короткий опис шаблону |
| Шаблон документа | Поле для додавання шаблону у в розширенні .dоcx  Поля, що мають заповнюватися автоматично та виводитися на згенерований образ мають бути заповнені відповідними даними –атрибутами, що мають позначатися фігурними дужками з кодом атрибуту. Атрибути розміщуються так, як вони мають виводитися.  Приклад:  C:\Users\daryna.karapuzova\Desktop\акан.png  Значення attrValues вказується для атрибутів, які створені вручну для Системи. Для базових ( типізованих) атрибутів системи, attrValues не пишеться. Надалі вказується саме ім’я атрибуту з бібліотеки атрибутів (код).  Якщо атрибут не відноситься до типу даних «Посилання» на системний довідник, то вказується лише його код.  Приклад використання стандартного атрибуту системи: |

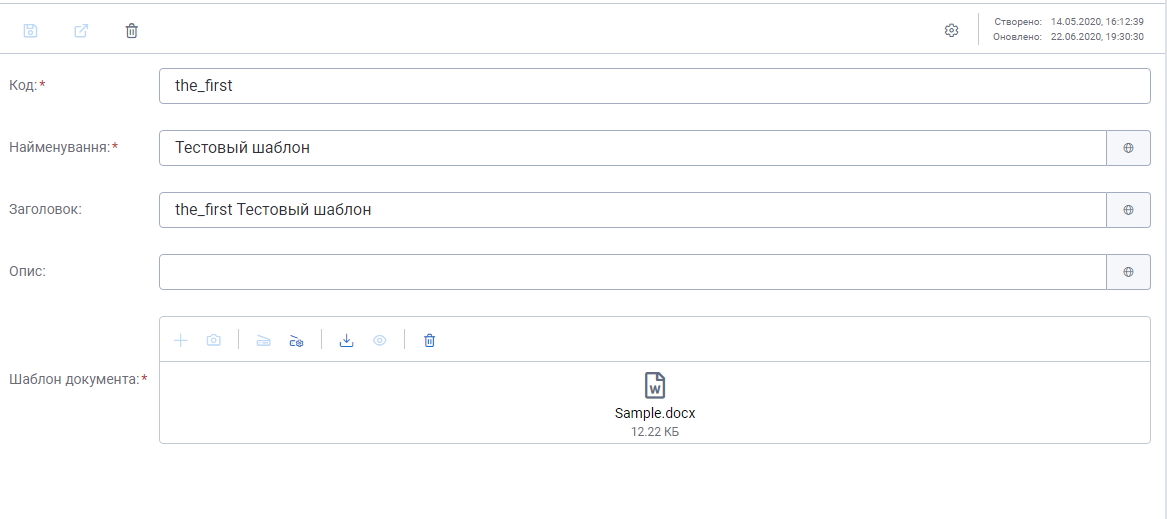


Рис. 2. 13. Приклад заповненого шаблону документа

Адміністратор може створювати нові, редагувати існуючі та видаляти шаблони, що не використовуються в жодному Типі документів.

Редагування поля «Код» можливе лише для шаблонів, що не використовуються в жодному Типі документів.

Видалення доступне лише для шаблонів, що не використовуються в жодному Типі документів.

В довіднику шаблонів документів Адміністратор може виконувати аналогічні дії, що й і в довіднику ролей (Рис. 2. 3).

## 2.7. Довідник типів зв’язків

Щоб створити тип зв’язку, необхідно в панелі навігації робочого столу «Типи документів» обрати ярлик «Налаштування» далі «Довідник типів зв’язку» та натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» та у вкладці, що відкрилась, заповнити поля (Рис. 2. 1), опис полів наведено в Таблиця 14.

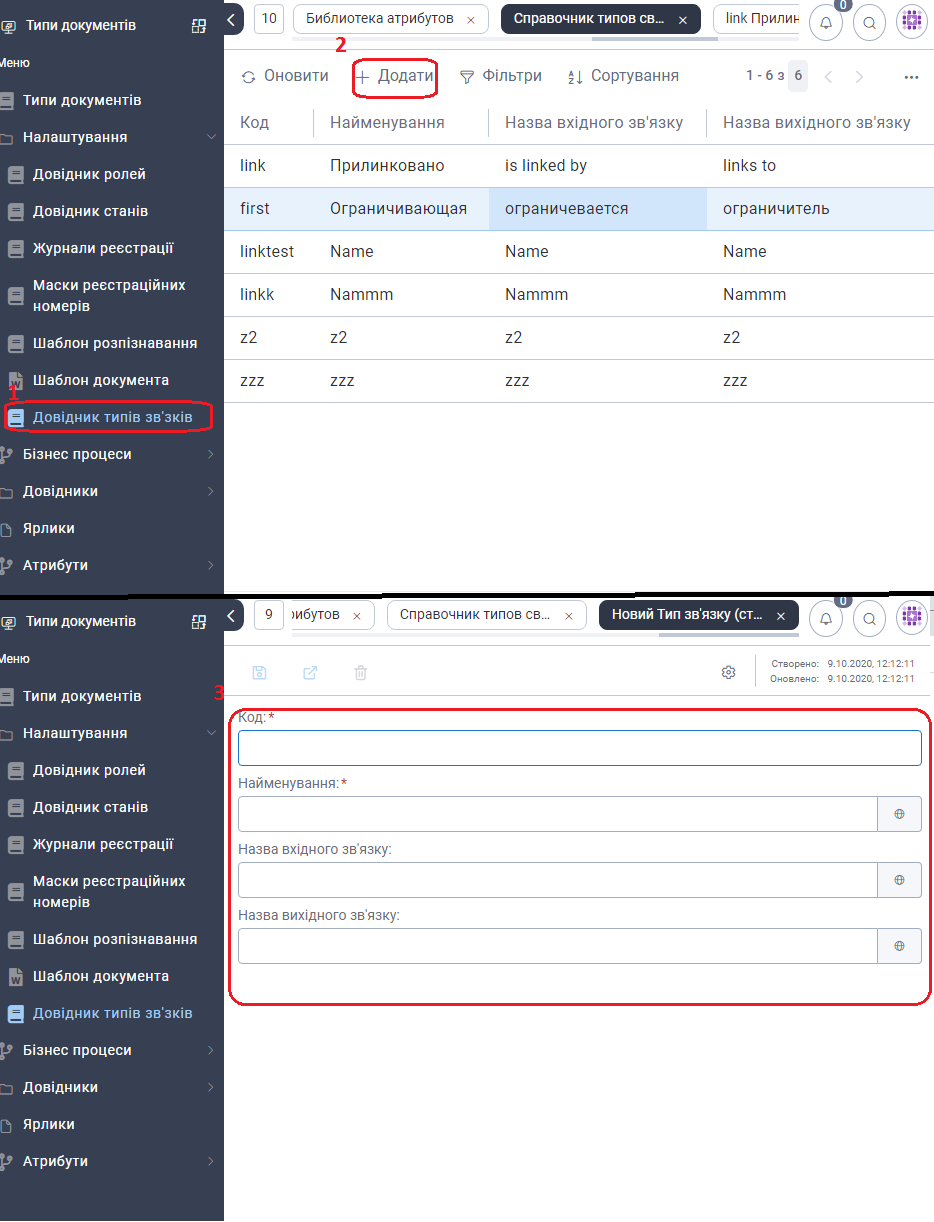


Рис. 2. 14. Додавання типів зв’язків

Таблиця 14

| **Назва поля** | **Опис** |
| --- | --- |
| Код\* | Унікальний ідентифікатор типу зв’язку |
| Найменування\* | Відображувана назва типу зв’язку |
| Назва вхідного зв’язку | Назва вхідного зв’язку |
| Назва вихідного зв’язку | Назва вихідного зв’язку |

Адміністратор може створювати нові, редагувати існуючі та видаляти типи зв’язків, що не використовуються.

Редагування поля «Код» можливе лише для типів зв’язків, що не пов’язані з жодним з Типом документів.

Видалення доступне лише для типів зв’язків, що не пов’язані з жодним з Типом документів.

В довіднику масок Адміністратор може виконувати аналогічні дії, що й і в довіднику ролей (Рис. 2. 3).

# 3. Бізнес-процеси

Створення бізнес-процесів описано в окремій інструкції

## 3.1. Дефініції процесів

Дефініції процесів – реєстр, що містить схеми бізнес-процесів та їх ітерацій.

Бізнес-процеси мають такі статуси:

* Актуальний – розгорнутий бізнес-процес
* Чернетка – бізнес-процес, що збережений але ще не розгорнутий
* Стара версія – попередня версія розгорнутого бізнес-процесу до якого вже було внесено зміни

Бізнес-процеси мають такий тип, що зазначається при створенні:

* BPMN 2.0
* DMN 1.1
* CMMN 1.1

Версіонність процесів формується на основі змін, що збережено в конкретному процесі

Адміністратор може створювати, редагувати, призупиняти дефініції процесів.

## 3.2. Форми задач

Форми задач – перелік налаштованих форм, які можна додати до завдань користувача в бізнес процесі для заповнення визначеним користувачем при виконанні визначеного завдання бізнес-процесу.

Створення форми задачі схоже зі вкладкою «Конструктор» в Типі документа (1.6. Конструктор).

Щоб створити форму задач необхідно в панелі навігації робочого столу «Типи документів» обрати ярлик «Бізнес процеси), далі «Форми задач» і натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати», у вкладці, що відкрилась, заповнити поля на вкладці «Головне», що описано в Таблиця 15 та налаштувати атрибути на вкладці «Форма» (Рис. 3. 1) відбувається по аналогії з вкладкою «Конструктор» Типу документів.

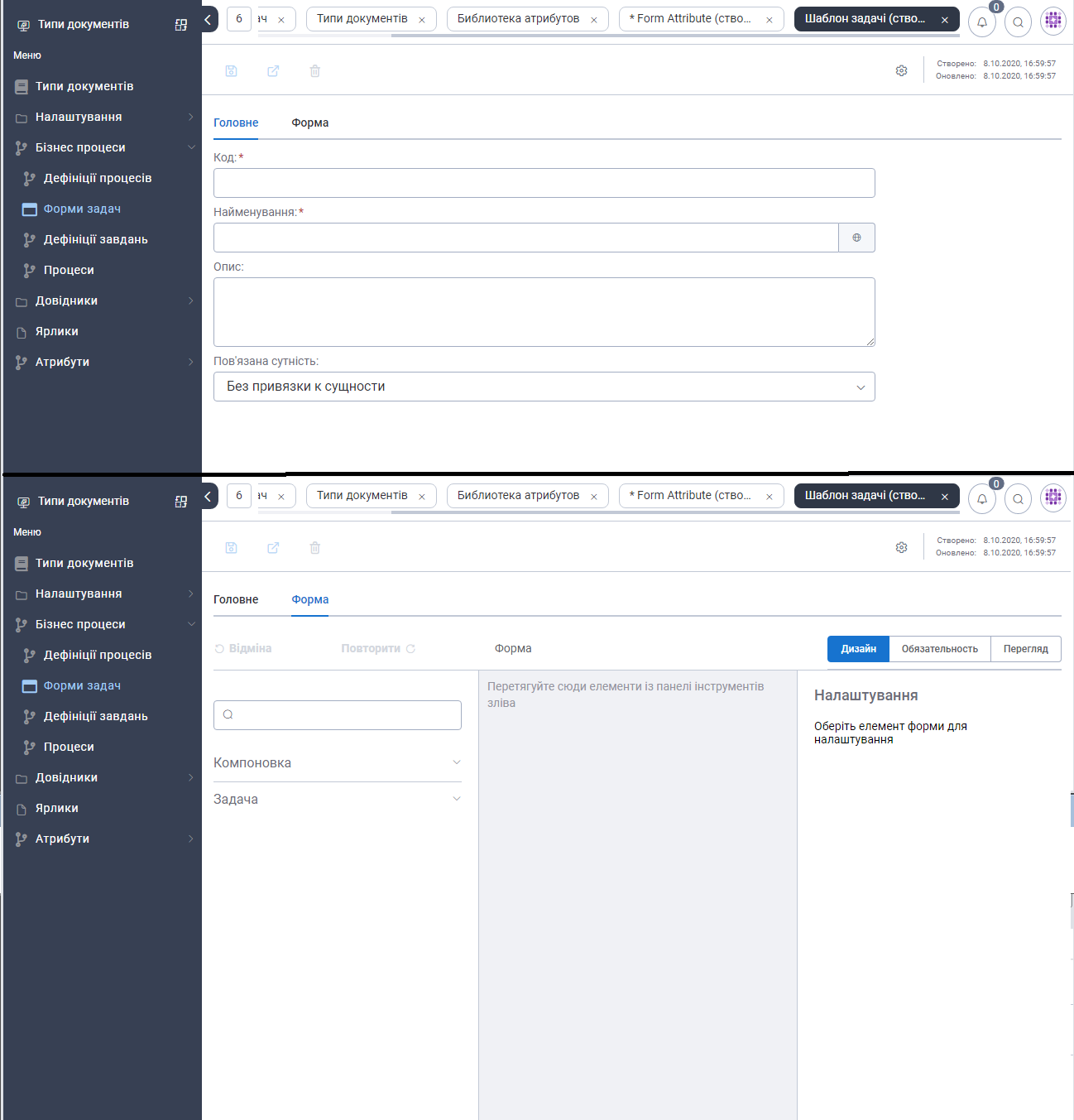


Рис. 3. 1. Створення форми задач

Таблиця 15

| **Назва поля** | **Опис** |
| --- | --- |
| Код\* | Унікальний ідентифікатор форми задачі |
| Найменування\* | Відображувана назва форми задачі |
| Опис | Короткий опис форми |
| Пов’язана сутність | Перелік прив’язок до сутностей. Можливі варіанти:   * Без прив’язки до сутності - при обранні даної прив’язки, на вкладці «Форма» відображаються стандартний набір атрибутів для завдання * Документ Scriptum – при обранні даної прив’язки, на вкладці «Форма» розширюється перелік атрибутів атрибутами типів документів * uba\_user – при обранні даної прив’язки, на вкладці «Форма» відображаються стандартний набір атрибутів для завдання з розширенням атрибутами стосовно користувача |

У випадку необхідності обов’язкового заповнення полів в задачі необхідно для цього поля встановити обов’язковість на вкладці «Форма» по аналогії з вкладкою «Конструктор» Типу документів.

Приєднання форми здачі до завдання користувача в схемі бізнес-процесу відбувається при обранні визначеного завдання користувача та в його панелі налаштувань на вкладці «Форми» в полі «Ключ форми» вказуванням коду форми задачі.

## 3.3. Дефініції завдань

Перелік дефініцій завдань заповнюється автоматично при введені в активність бізнес процесу відповідно BPMN 2.0

## 3.4. Процеси

Процеси – перелік запущених бізнес процесів, що виконуються або вже завершені.

При відкритті визначеного процесу Адміністратор може переглянути в якому він стані (що відбувається з процесом, на якому він етапі) (Рис. 3. 2).

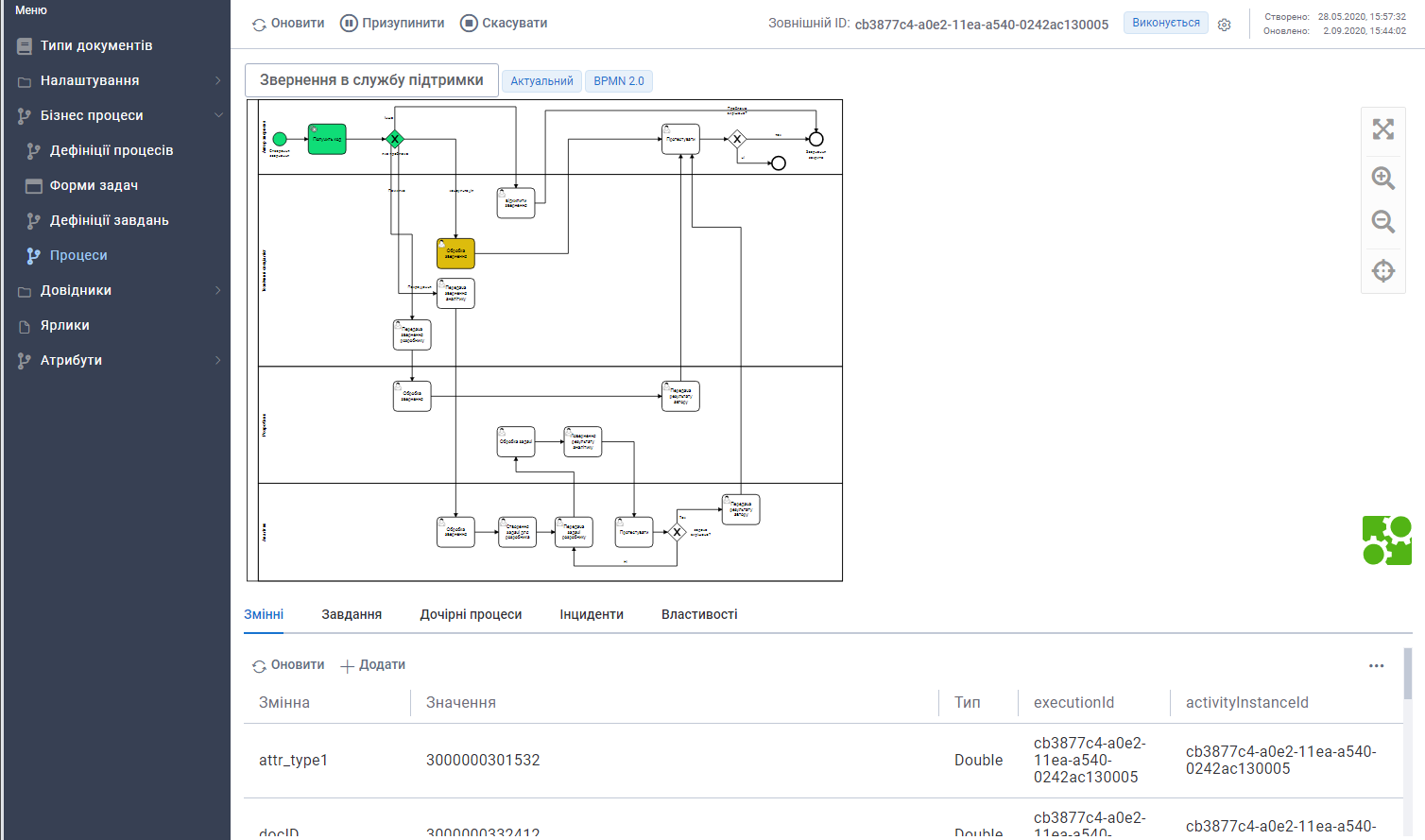


Рис. 3. 2. Перегляд стану процесу

З процесу можна перейти до дефініції процесу натиснувши на назву процесу, що знаходиться в лівому верхньому куті форми, а також можна переглянути статус та тип бізнес-процесу, що знаходиться після кнопки з назвою процесу (Рис. 3. 3).

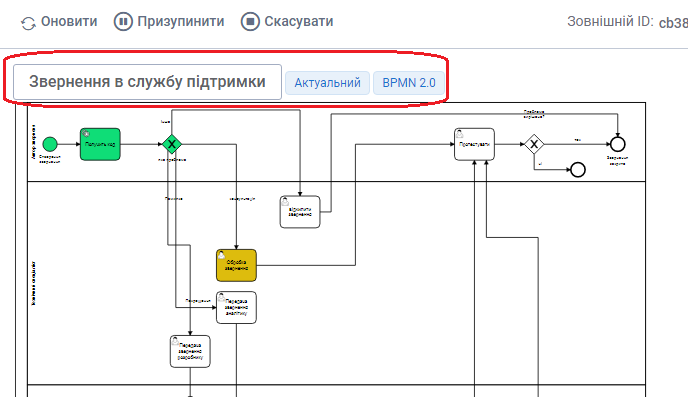


Рис. 3. 3. Назва, статус та тип процесу

Окрім графічного відображення стану процесу Адміністратор може переглянути вкладки з інформацією про процес (Рис. 3. 2):

* Змінні – перелік змінних, які увійшли до процесу
* Завдання – перелік завдань користувачів. Подвійний натиск правою кнопкою миші на обране завдання відкриває форму завдання
* Дочірні процеси – перелік завершених або тих, що виконуються підпроцесів поточного процесу
* Інциденти – перелік інцидентів/помилок процесу, що відбулися з системним завданням. При обранні конкретного інциденту та натисканні кнопки «Повторити» відбувається повторний запуск завдання
* Властивості – інформація про запущений процес

Кольорове відображення елементів процесу:

* Зелений – елемент процесу що виконав свою дію
* Жовтий – елемент, на якому відбувається дія
* Білий – елемент, що не почав виконувати дію
* Червоний – елемент на якому відбувався інцидент

# 4. Довідники

## 4.1. Групи виконавців

Довідник дозволяє при створенні Типу документів обрати заданий набір виконавців з групи.

Для створення групи виконавців необхідно в панелі навігації робочого столу «Типи документів» обрати ярлик «Довідники» далі «Групи виконавців», натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» та у формі, що відкрилась (Рис. 4. 1) заповнити поля, що описані в Таблиця 16.

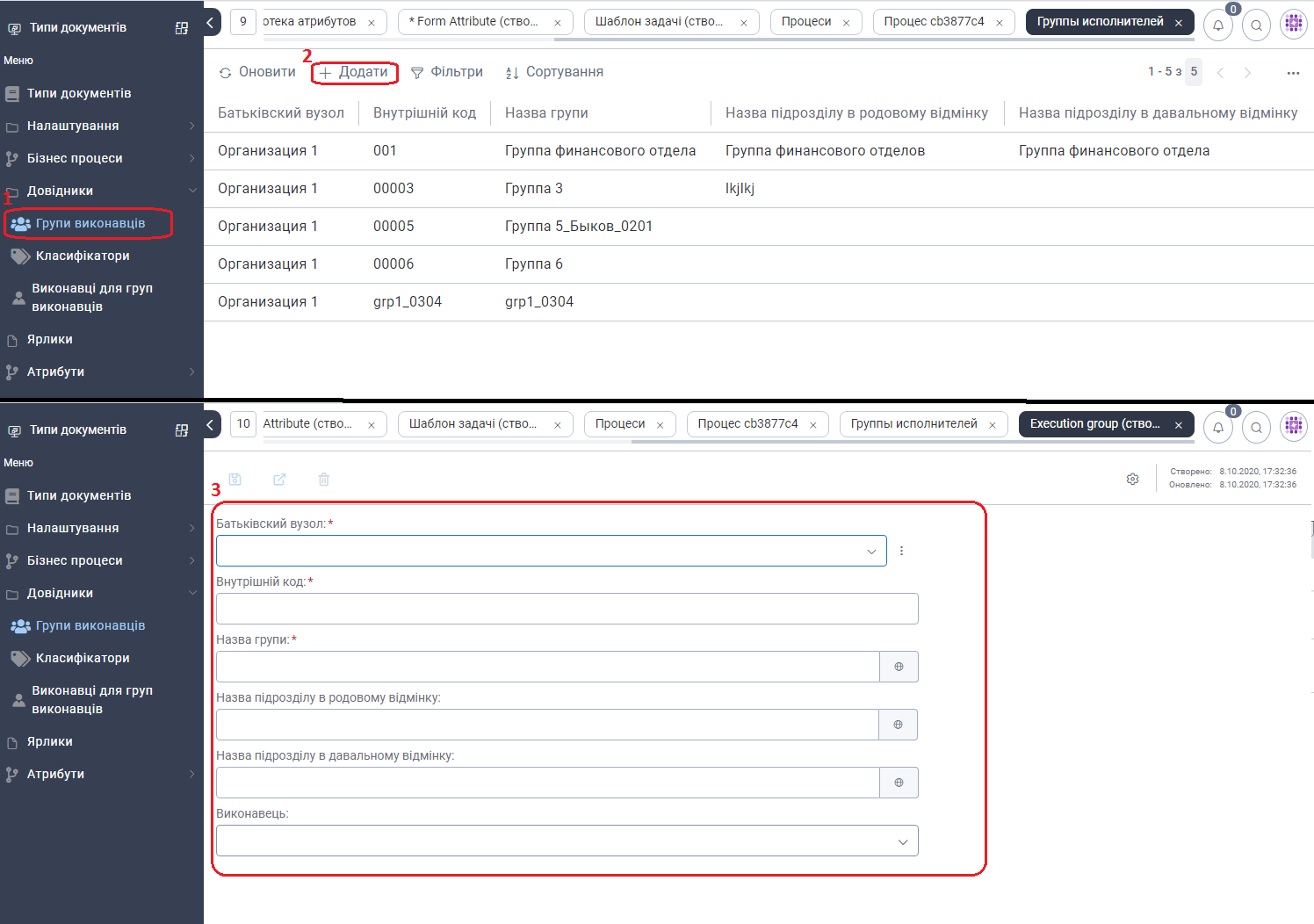


Рис. 4. 1. Створення групи виконавців

Таблиця 16

|  |  |
| --- | --- |
| **Назва поля** | **Опис** |
| Батьківський вузол\* | Поле вибору, в якому обирається організаційна одиниця, якій підпорядковується група, що створюється |
| Внутрішній код\* | Ідентифікатор групи |
| Назва групи | Відображувана назва групи |
| Назва підрозділу в родовому відмінку | Назва групи в родовому відмінку |
| Назва підрозділу в давальному відмінку | Назва групи в давальному відмінку |
| Виконавець | Перелік штатних одиниць, які належатимуть до групи |

## 4.2. Класифікатори

Довідник класифікаторів дозволяє об’єднати групи елементів для вибору з категорії. Класифікатори створюється та коригується лише адміністратором Системи.

Для створення класифікатору необхідно в панелі навігації робочого столу «Типи документів» обрати ярлик «Довідники» далі «Класифікатори», натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» та у формі, що відкрилась (Рис. 4. 2) заповнити поля, що описані в Таблиця 17.

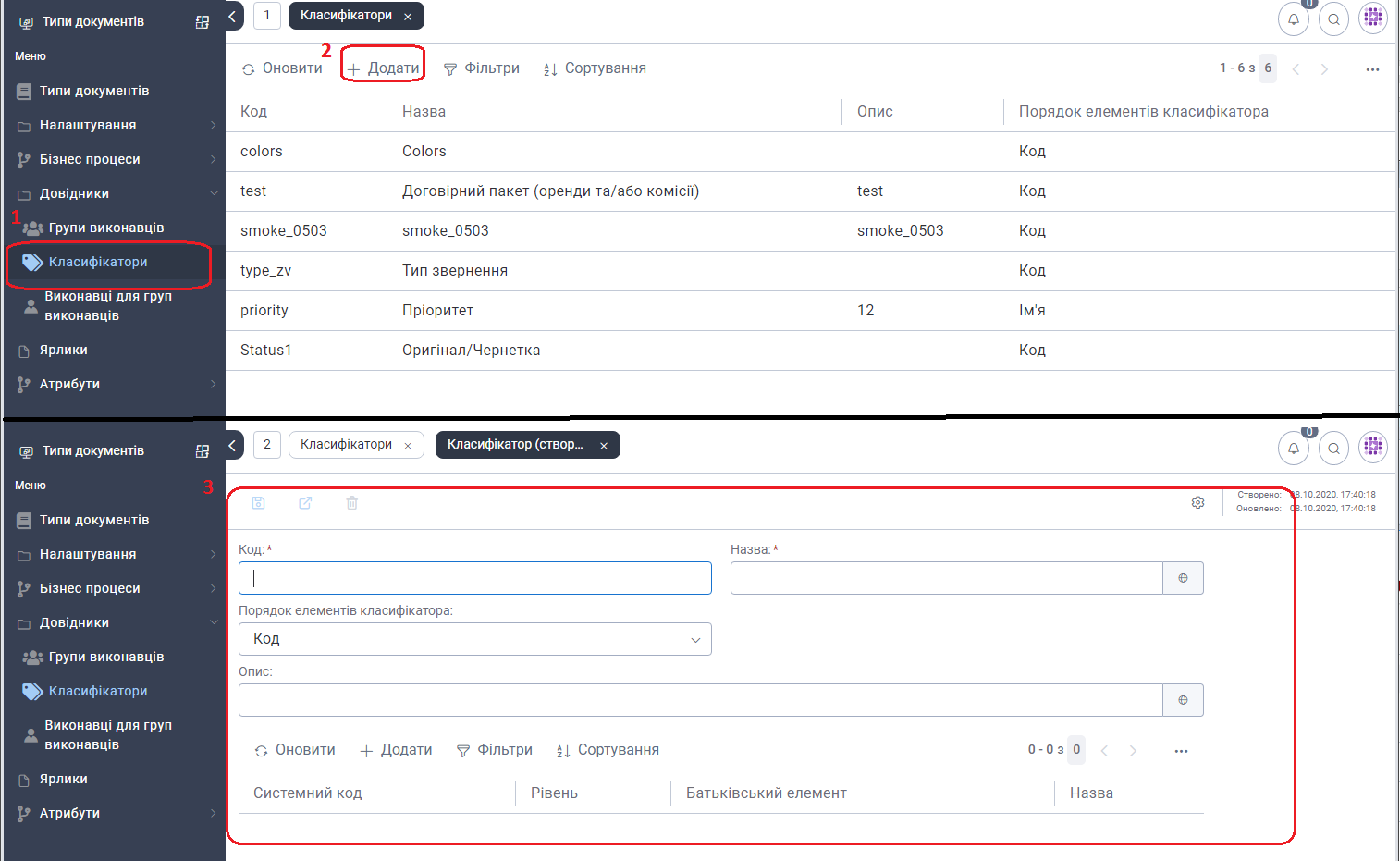


Рис. 4. 2. Створення класифікатора

Таблиця 17

|  |  |
| --- | --- |
| **Назва поля** | **Опис** |
| Код\* | Унікальний ідентифікатор даного класифікатору |
| Назва\* | Відображувана назва даного класифікатору |
| Порядок елементів класифікатора | Вказується порядок за яким відображатимуться елементи класифікатора. Можливі значення:   * Код * Назва |
| Опис | Короткий опис даного класифікатора |

Для додавання елементу класифікатора необхідно в нижній частині форми натиснути кнопкуC:\Users\daryna.karapuzova\Desktop\нет.png «Додати», та на формі, що з’явилась заповнити поля (Рис. 4. 3) Таблиця 18.

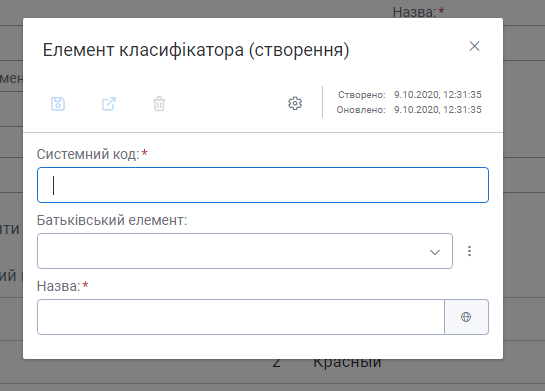


Рис. 4. 3. Параметри створення предмету класифікатора

Таблиця 18

|  |  |
| --- | --- |
| **Назва поля** | **Опис** |
| Системний код\* | Ідентифікатор предмету класифікатора |
| Батьківський елемент | Пов’язання предметів класифікатора між собою. Залежно від того, як вони пов’язуються між собою, так проставляється їх рівень (колонка «Рівень») |
| Назва\* | Відображувана назва предмету класифікатора |

## 4.3. Виконавці для груп виконавців

Вказуються виконавці (штатні одиниці), що належать до визначеної групи виконавців.

Щоб додати виконавця до групи необхідно в панелі навігації робочого столу «Типи документів» обрати ярлик «Довідники» далі «Виконавці для груп виконавців», натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» та у формі, що відкрилась (Рис. 4. 4) заповнити поля, що описані в Таблиця 19.

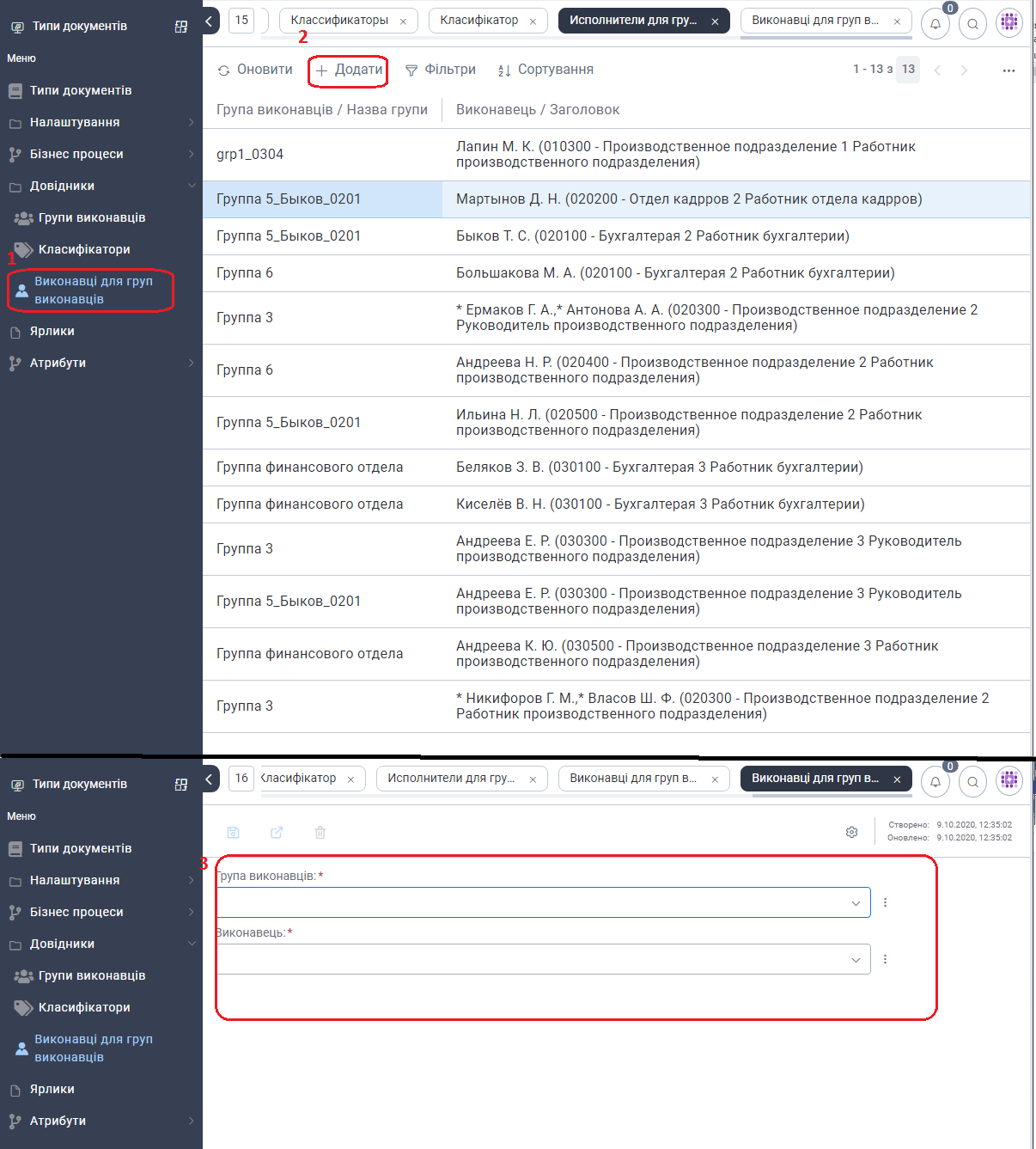


Рис. 4. 4. Додавання виконавців до груп виконавців

Таблиця 19

|  |  |
| --- | --- |
| **Назва поля** | **Опис** |
| Група виконавців\* | Перелік груп виконавців |
| Виконавець\* | Перелік штатних одиниць |

# 5. Ярлики

Ярлики відображають реєстри документів чи інших сутностей (завдання, контрольні точки). Адміністратор може:

* створювати паки ярликів, обмеженої вкладеності
* створювати ярлики в папках чи окремо від папок
* прив’язувати до ярликів реєстрів відповідних сутностей (завдання, контрольні точки) з можливістю обрання типу/типів сутностей та переліку атрибутів, які будуть відображатися у вигляді колонок реєстру
* використовувати атрибути облікового запису для фільтрації при формуванні реєстрів

## 5.1. Створення ярликів

Адміністратор має можливість створювати нові ярлики та здійснювати адміністрування прав на їх використання.

Для створення необхідно встановити курсор на будь-який існуючий ярлик активного робочого столу, викликати контекстне меню натиском правої клавіші миші та обрати «Додати ярлик» (Рис. 5. 1). Або в панелі навігації «Типи документів» обрати ярлик «Ярлики» далі натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» (Рис. 5. 2) та заповнити поля у формі, що відкрилась (Рис. 5. 3), які описано в Таблиця 20.

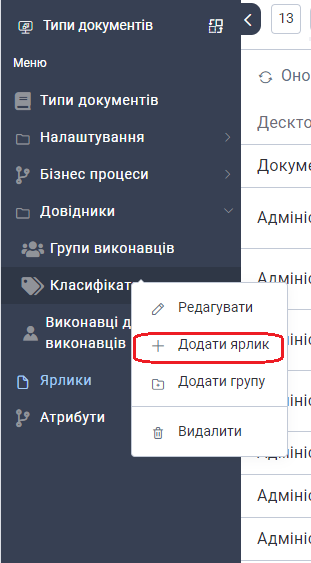


Рис. 5. 1. Створення ярлика через контекстне меню

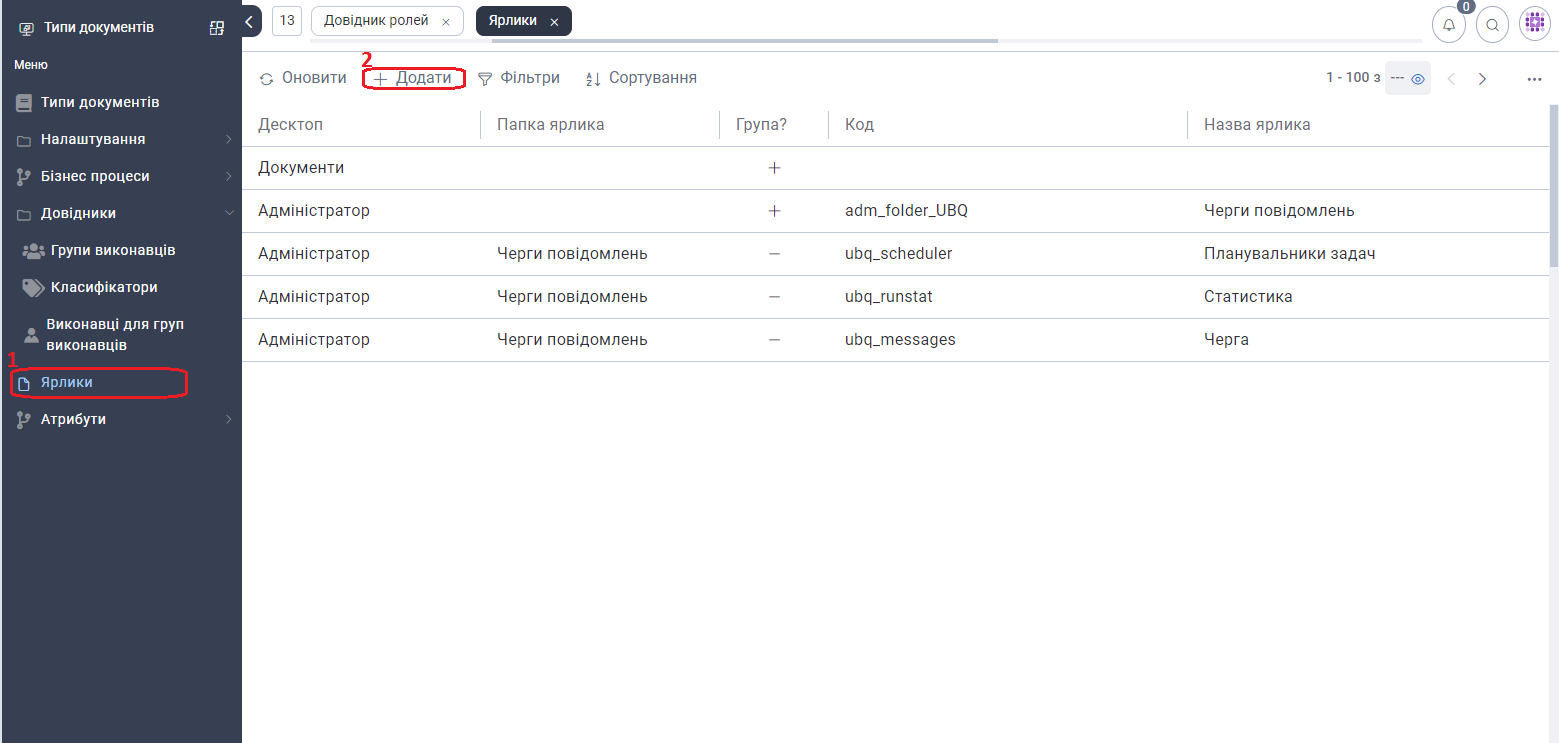


Рис. 5. 2. Створення ярлика через реєстр «Ярлики»

## 5.2. Створення групи ярликів

Адміністратор має можливість створювати нові групи ярликів та здійснювати адміністрування прав на їх використання. Для цього необхідно встановити курсор на будь-яку існуючу групу ярликів активного робочого столу, викликати контекстне меню та обрати «Додати групу» (Рис. 5. 1).

Відкриється форма властивостей нової групи ярликів (Рис. 5. 3), що аналогічні з властивостями ярлика Таблиця 20 (Рис. 2. 10).

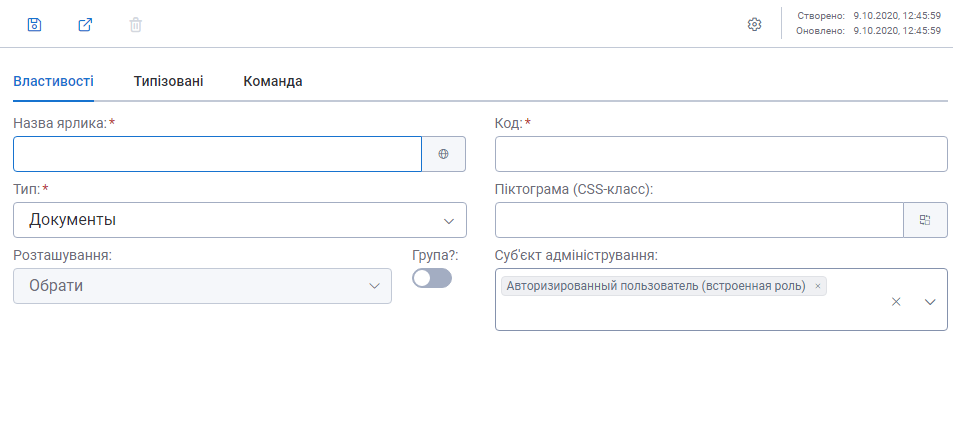


Рис. 5. 3. Параметри створення групи ярликів

Вкладка «Команда» містить програмний код.

Для адміністрування групи ярликів необхідно відкрити вкладку «Права на ярлик».

Спочатку таблиця, що розташована на вкладці, не містить жодного запису.

Після внесення всіх потрібних змін у налаштування нової групи ярликів адміністратор зберігає зміни та закриває форму налагодження групи ярликів. Група ярликів відображається у визначену робочому столі відразу, без перезавантаження серверу застосувань.

## 5.3. Редагування ярликів/груп ярликів

Адміністратор має можливість налаштування груп, що входять до активного робочого столу, та адміністрування прав на їх використання. Для цього необхідно встановити курсор на групу ярликів, викликати контекстне та обрати «Редагувати» (Рис. 2. 8).

Адміністратор завжди має дозвіл на ярлик, що створюється.

У формі, що відкрилась, заповнити необхідні поля. Форма складається з таких вкладок (Рис. 5. 3):

* Властивості
* Типізовані
* Команда

### 5.3.1. Властивості

Вкладка «Властивості» (Рис. 2. 10) містить опис властивостей даної групи. Поля вкладки описано в Таблиця 15.

Таблиця 20

| **Назва поля** | **Опис** |
| --- | --- |
| Назва ярлика\* | Назва, яка відображається в панелі навігації |
| Код\* | Унікальний ідентифікатор даного ярлику |
| Тип\* | Тип ярлику. Можливі варіанти:   * Документи * Завдання * Системний |
| Піктограма (CSS-клас) | Задає код піктограми, яка відображається перед назвою ярлика у панелі навігації, обрати зі списку необхідну піктограму |
| Розташування | Розташування ярлику в панелі навігації |
| Група? | Є ознакою, чи є ярлик логічним або групою |
| Суб’єкт адміністрування | Користувачі, які мають доступ до ярлика.  Адміністратор завжди має дозвіл на ярлик, що створюється |

### 5.3.2. Типізовані

На вкладці типізовані відображається список стандартних елементів та елементів групи атрибутів з яких можна створити реєстр ярлика (Рис. 5. 4). Поля описано в Таблиця 21.

Щоб створити відображуваний список ярлика необхідно перетягати зі списку елементів ліворуч необхідні елементи до необхідних полів праворуч.

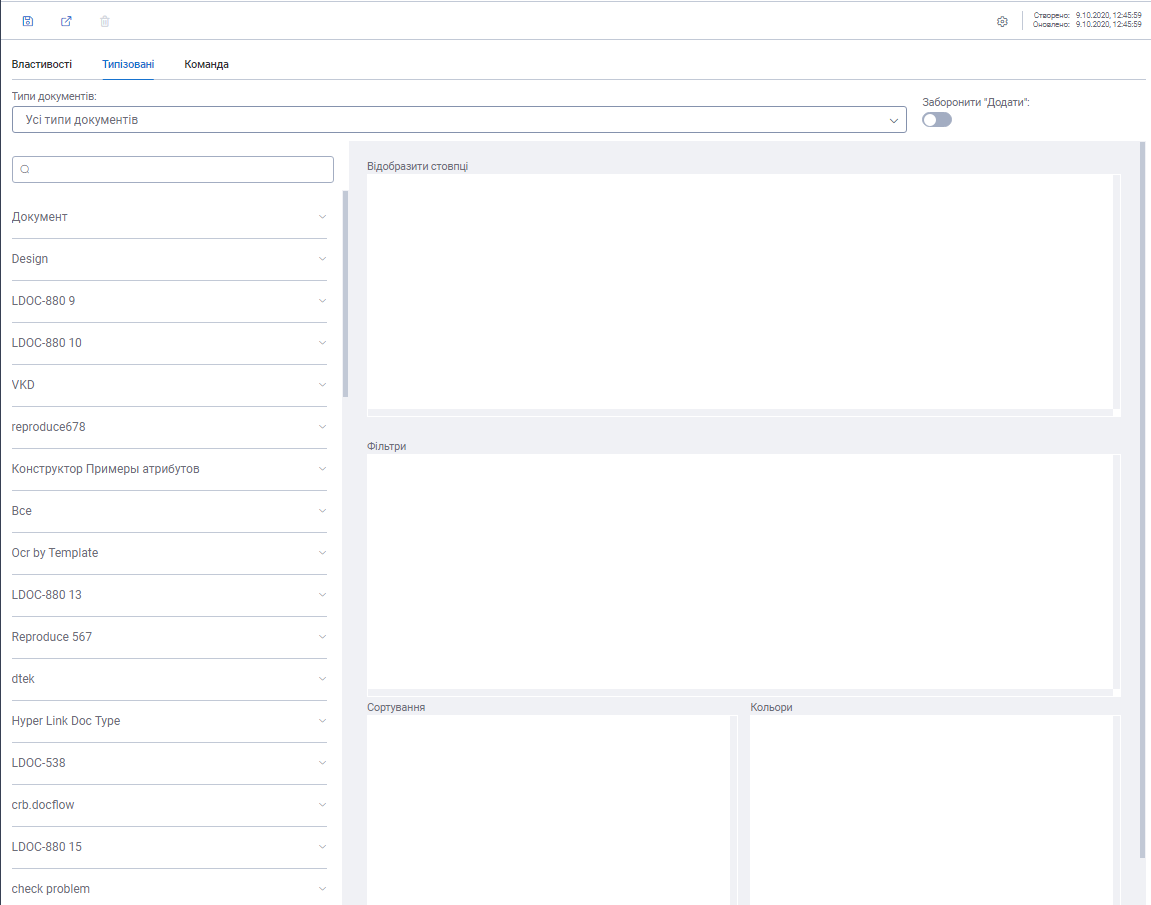


Рис. 5. 4. Параметри налаштування реєстру

Таблиця 21

| **Назва поля** | **Опис** |
| --- | --- |
| Типи документів | Перелік Типів документів, поля яких, використовуватимуться при формування реєстру |
| Заборонити «Додати» | Перемикач, що забороняє/дозволяє додавання документу одразу в реєстрі для якого створюється ярлик |
| Відобразити стовпці | Елементи, що мають відображуватися як стовбці, перетягуючи зі списку типів документів ліворуч |
| Фільтри | Елемент, за якими фільтруватиметься створений список та відображається згори списку |
| Сортування | Стовбці списку, які можуть сортуватися |
| Кольори | Елементи, що відділятимуться вказаним кольором |

Після внесення всіх потрібних змін у налаштування групи ярликів адміністратор зберігає зміни та закриває форму налагодження групи. Всі зміни до властивостей групи ярликів застосуються відразу, без перезавантаження серверу застосувань.

### 5.3.3. Команда

Вкладка «Команда» містить програмний код, який і задає умови відображення даних у реєстрі.

## 5.4. Видалення ярликів/груп ярликів з панелі навігації

Адміністратор має можливість видаляти ярлика та групи ярликів активного робочого столу з панелі навігації.

Для цього необхідно встановити курсор на будь-який ярлик чи групу ярликів активного робочого столу та в контекстному меню обрати «Видалити».

Відкриється діалогове вікно підтвердження видалення з кнопками «Так», «Ні».

При виборі «Так» система видаляє ярлик/групу ярликів з робочого столу.

# 6. Атрибути

В Системі наявні стандартні (базові) атрибути та користувацькі (атрибути, що створюються користувачем). Нижче описано створення користувацьких атрибутів.

## 6.1. Бібліотека атрибутів

Адміністратор може створювати користувацькі атрибути документів для використання в Типах документів окрім стандартних атрибутів.

Атрибути можна створювати, редагувати та видаляти. Управління атрибутами, тобто налаштування параметрів, здійснюється в бібліотеці атрибутів (Рис. 6. 1).

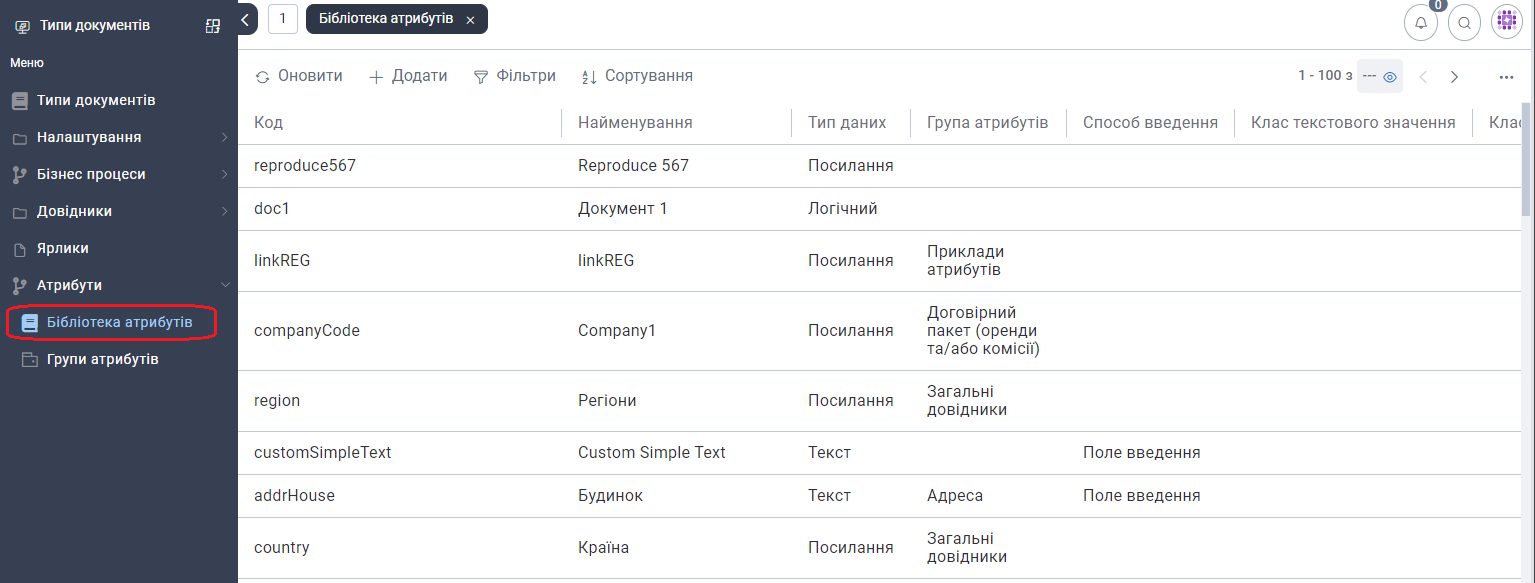


Рис. 6. 1. Бібліотека атрибутів

Системою підтримуються наступні види стандартних атрибутів:

* Числові атрибути (цілі, речові, грошові)
* Логічні атрибути (True/False)
* Атрибути типу Datetime (дата, дата та час)
* Текстові атрибути (рядки та багаторядковість)
* Колекції (набір об’єктів, що мають атрибутивний стан)
* Класифікатори (посилання на класифікатор)
* Довідники (посилання на довідник)
* Гіперпосилання (посилання на web-сторінку)

## 6.2. Групи атрибутів

Групи атрибутів призначені для структурування користувацьких атрибутів у певні групи, що відображатимуться на вкладці «Конструктор» Типу документів.

Адміністратор може створювати нові, редагувати або видаляти існуючі групи.

Редагувати поле «Код» можливо якщо група не використовується ні в параметрах користувацьких атрибутів, ні в налаштуваннях Типів документів.

Редагувати поле «Найменування» дозволено без обмежень.

## 6.3. Створення атрибутів та груп атрибутів

Для створення атрибуту (групи атрибутів) необхідно виконати таку послідовність дій:

1. В панелі навігації «Типи документів» обрати папку «Атрибути» далі довідник «Групи атрибутів». Створити групу атрибутів натиснувши кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» та заповнивши поля коду та найменування (Рис. 6. 2).

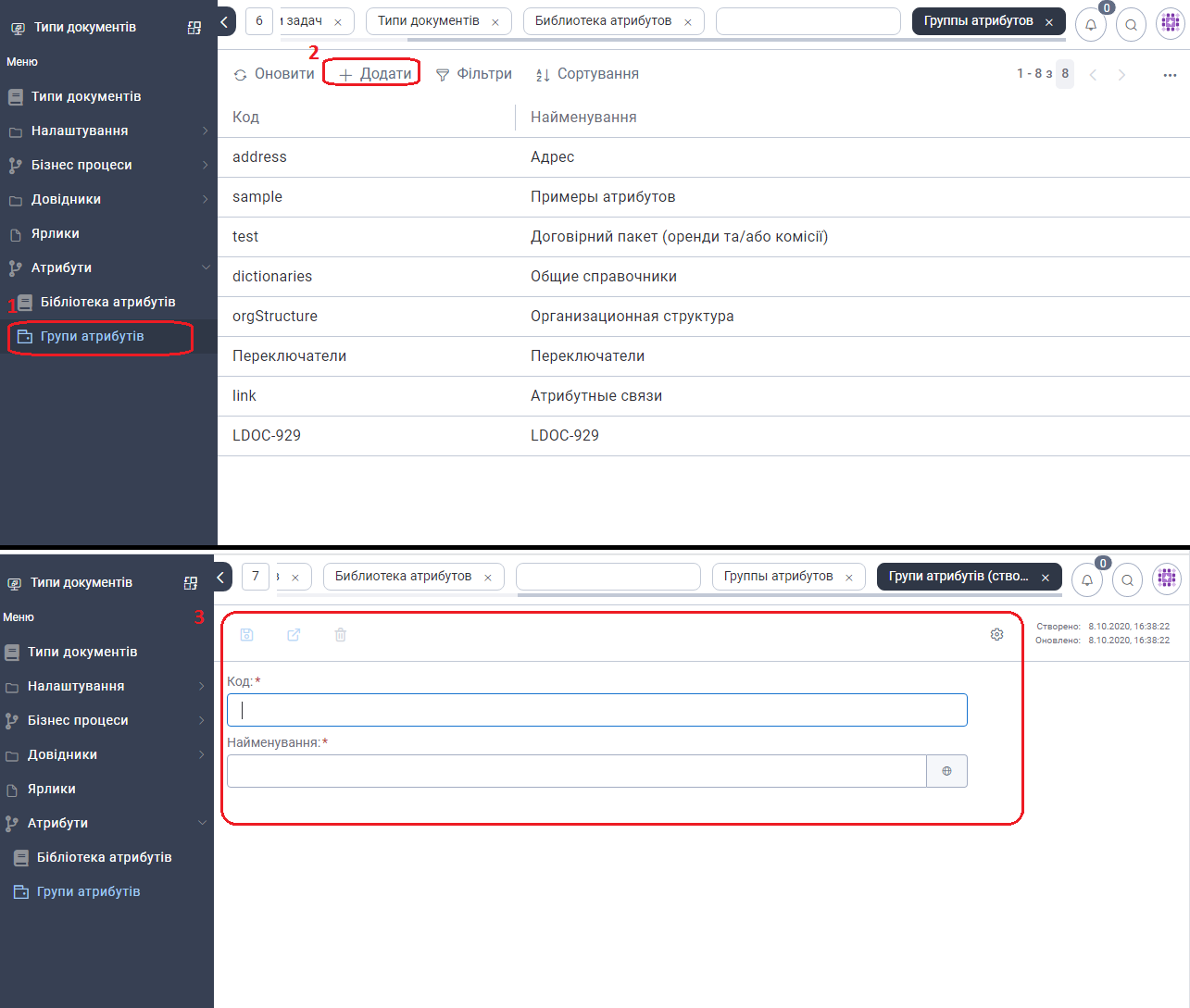


Рис. 6. 2. Групи атрибутів

1. В довіднику «Бібліотека атрибутів» натиснути кнопку C:\Users\daryna.karapuzova\Desktop\нет.png «Додати» та далі заповнити картку атрибута (Рис. 6. 3). Поля картки атрибуту описано в Таблиця 22. Створений атрибут має бути збереженим.

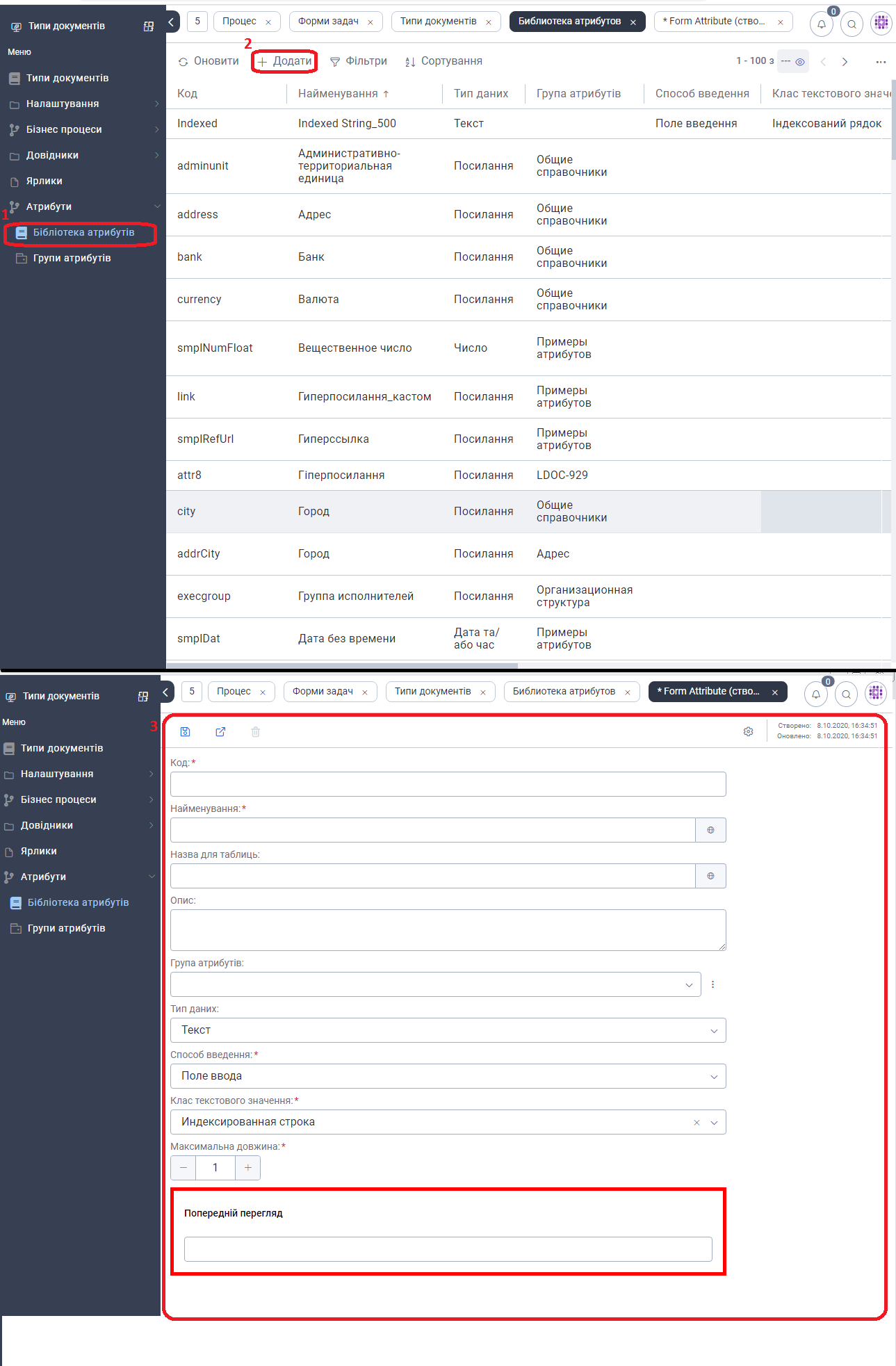


Рис. 6. 3. Створення атрибутів

Таблиця 22

| **Назва поля** | **Опис** |
| --- | --- |
| *Основні поля* | |
| Код\* | Унікальний ідентифікатор даного атрибуту |
| Найменування\* | Відображувана назва атрибута |
| Назва для таблиць | Найменування та назва для стовбців реєстрів даних |
| Опис | Короткий опис даного атрибута |
| Група атрибутів | Назва групи, до якої буде належати атрибут |
| Тип даних\* | Доступний тип даних:   * Текст * Число * Дата та/або час * Посилання * Логічний * Таблиця/колекція * Файл   Залежно від вибору типу додаються нові або змінюються поля форми. |
| Попередній перегляд | Графічне відображення створеного атрибута |
| *Змінні поля* | |
| Спосіб введення\* | Додаткове поле типу даних «Текст».  Встановлюється спосіб введення тексту атрибуту.  Можливі варіанти:   * Поле введення * Багаторядкове поле введення |
| Клас текстового значення\* | Додаткове поле типу даних «Текст».  Можливі варіанти:   * Індексований рядок * Розширений рядок |
| Максимальна довжина\* | Додаткове поле типу даних «Текст».  Максимальна кількість символів в даному атрибуті (максимальна довжина – 4000 символів) |
| Кількість рядків | Додаткове поле типу даних «Текст», з’являється в разі обрання способом введення «Багаторядкове поле введення».  Додається кількість необхідних рядків даному атрибуту |
| Клас числа\* | Додаткове поле типу даних «Число».  Встановлюється можливість вказання діапазону значень цілого, числа з плаваючою точкою та грошового атрибутів  Можливі варіанти:   * Число з плаваючою точкою * Ціле число * Гроші |
| Мінімальне значення | Додаткове поле типу даних «Число»  Встановлюється мінімальне значення для атрибуту |
| Максимальне значення | Додаткове поле типу даних «Число»  Встановлюється максимальне значення для атрибуту |
| Клас часового значення\* | Додаткове поле типу даних «Дата та/або час».  Встановлюється атрибут дати та/або часу.  Можливі варіанти:   * Дата * Дата та час |
| Тип посилання | Додаткове поле типу даних «Посилання»  Встановлюється посилання для атрибуту.  Можливі варіанти:   * Гіперпосилання * Системний довідник * Класифікатор |
| Гіперпосилання | Додаткове поле типу даних «Посилання» |
| Системний довідник\* | Додаткове поле типу даних «Посилання», з’являється в разі обрання типом посилання «Системний довідник».  Обирається посилання на довідник з випадаючого списку довідників Системи. |
| Класифікатор\* | Додаткове поле типу даних «Посилання», з’являється в разі обрання типом посилання «Класифікатор»  Вказується посилання на класифікатор |
| Дочірні атрибути таблиці/колекції\* | Додаткове поле типу даних «Таблиця/колекція».  Встановлюється набір стовбців та їх порядок відображення. |
| Розширення файлу | Додаткове поле типу даних «Файл»  Перелік доступних форматів для файлів з можливістю обрання декількох розширень. Можливі значення:   * .doc * .docx * .xls * .xlsx * .pdf * .txt |
| Максимальний розмір | Додаткове поле типу даних «Файл»  Встановлюється максимальне значення розміру файлу |

# 7. Спеціальні атрибути

## 7.1. Образ документа

Адміністратор може використовувати при налаштуванні Типу документа спеціальний атрибут «Образ документа» (Рис. 7. 1). Даний атрибут відображає образ (змістовне) документа на формі документа з можливістю його масштабування якщо образ документа збережено в форматі .pdf.

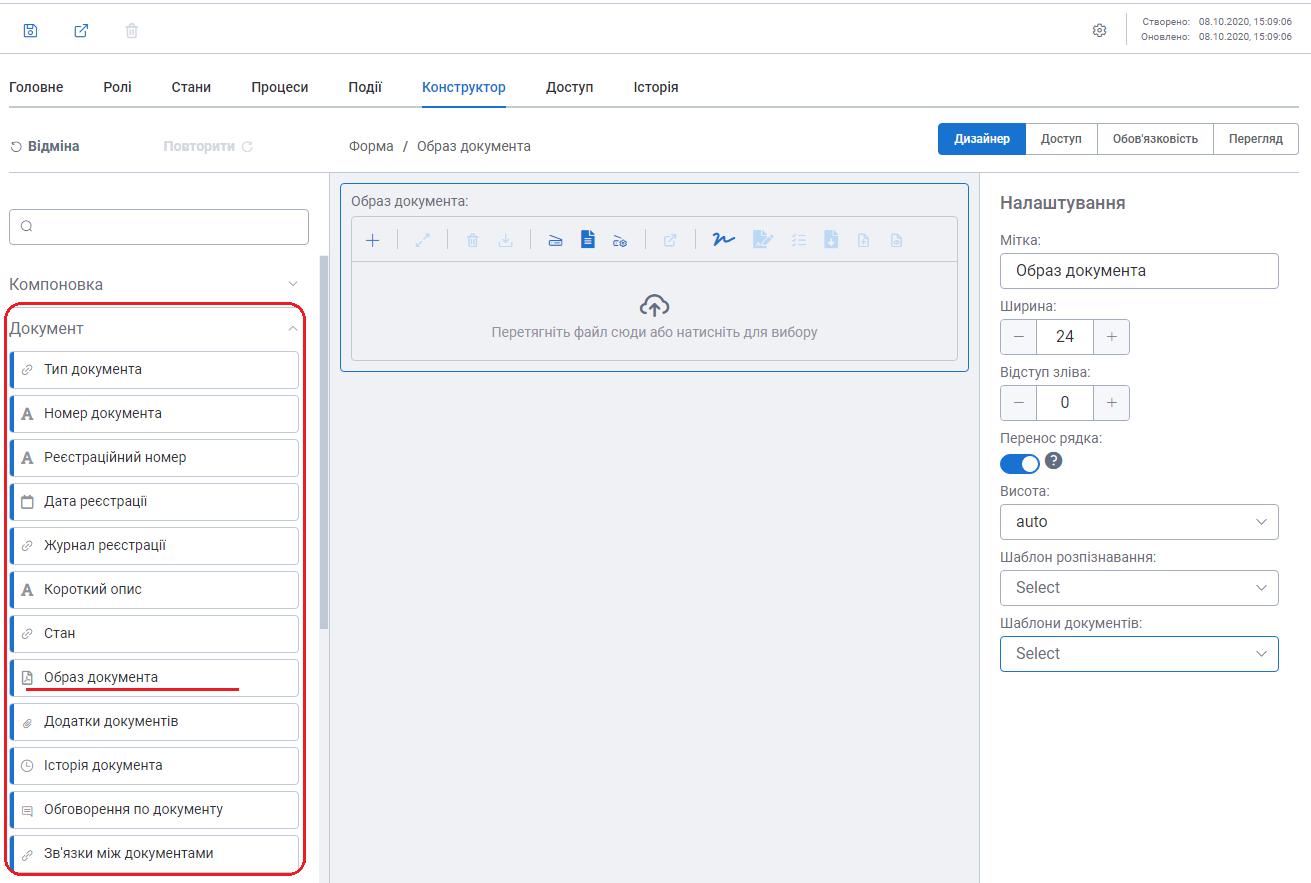


Рис. 7. 1. Атрибут «Образ документу»

Опис можливих налаштувань атрибуту на вкладці «Дизайнер» описано в Таблиця 23.

Таблиця 23

| **Назва поля** | **Опис** |
| --- | --- |
| Мітка | Відображуване найменування поля |
| Ширина | Допустимі розміри мітки |
| Відступ зліва | Розмір відступу зліва |
| Перенос рядка | Перемикач для переносу значення в рядку |
| Висота | Вказується висота рядка. Можливі варіанти:   * auto * px * % |
| Шаблон розпізнавання | Вказується перелік шаблонів розпізнавання (2.5. Шаблон розпізнавання) для атрибуту |
| Шаблони документів | Вказується перелік шаблонів документів (2.6. Шаблон документа) |

Адміністратор може налаштовувати обмеження доступу користувачів по аналогії з користувацькими атрибутами (доступ на перегляд/зміну атрибуту в залежності від стану документа та ролі користувача) (1.6. Конструктор).

### Завантаження образу документа з файлу

Система дозволяє користувачу завантажувати образ документа з файлу формату .pdf, а також з формату .doc, .docx, .rtf, .txt  з подальшою конвертацією його у формат .pdf з текстовим слоєм.

Завантажений файл зберігається обох форматах (початковий та .pdf). Скачування доступне для початкового формату.

Адміністратор може встановити право на редагування атрибуту загалом, в цьому випадку користувач може створювати нові версії образу.

### Сканування образу документа

Адміністратор може встановити право на редагування атрибуту загалом, в цьому випадку користувач може створювати нові версії образу.

Окремо виконується сканування файлу. Після, у випадку необхідності, примусово виконується створення версії. В такому випадку змістовне завантаженого файлу очищається та файл стає версією.

### Підпис образу документа КЕП

Система дозволяє підписувати образ за допомогою файлу на пристрої користувача або за допомогою апаратного носія.

Примітка: Для підписів нанесених на образ та додатки документа встановлюється ознака *internal/external* залежно від того як було нанесено підпис на додаток безпосередньо в Системі чи завантажено з-зовні. Підписи, що нанесені в Системі, заборонено видаляти в будь якому випадку.

### Версії образу документа

Атрибут «Образ документа» дозволяє створення нових версій образу документа, що ініціюються або користувачем або автоматично в процесі обробки документу. Атрибут зберігає попередньої версії образу документа разом з нанесеними на них ЕЦП. А також дозволяє переглянути попередні версії образу та нанесених на них ЕЦП безпосередньо з картки документа.

Версії можна видаляти або скачувати. Редагування чи повернення до попередньої версії неможливе.

Адміністратор може встановити право на редагування атрибуту загалом, в цьому випадку користувач може створювати нові версії образу.

Окремо виконується завантаження/сканування файлу. Після, у випадку необхідності, примусово виконується створення версії. В такому випадку змістовне завантаженого файлу очищається та файл стає версією.

Для кожної версії відображається номер версії, дата та час створення версії, хто створив версію та його посада.

Для окремо обраної версії відображається образ, підписи нанесені на версію без можливості їх видалення.

## 7.2. Додатки документів

Адміністратор може використовувати при налаштуванні Типу документа спеціальний атрибут «Додатки документа» (Рис. 7. 2). Даний атрибут відображає перелік додатків документа у вигляді набору іконок з назвами додатків.

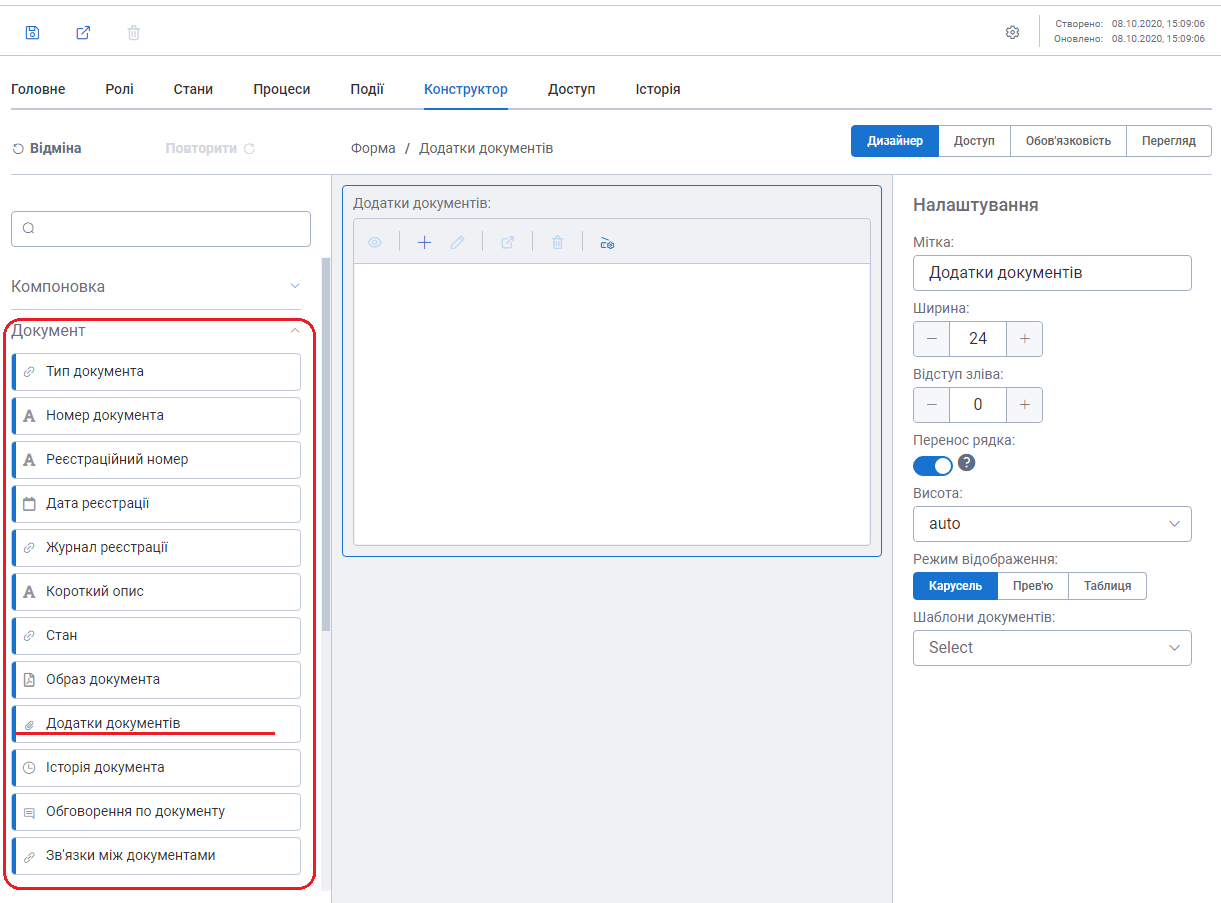


Рис. 7. 2. Атрибут «Додатки документів»

Опис можливих налаштувань атрибуту на вкладці «Дизайнер» описано в Таблиця 24.

Таблиця 24

| **Назва поля** | **Опис** |
| --- | --- |
| Мітка | Відображуване найменування поля |
| Ширина | Допустимі розміри мітки |
| Відступ зліва | Розмір відступу зліва |
| Перенос рядка | Перемикач для переносу значення в рядку |
| Висота | Вказується висота рядка. Можливі варіанти:   * auto * px * % |
| Режим відображення | Обирається вигляд для відображення переліку додатків:   * Карусель * Прев’ю * Таблиця |
| Шаблони документів | Вказується перелік шаблонів документів (2.6. Шаблон документа) |

Адміністратор може налаштовувати варіанти відображення атрибуту для конкретного Типу документа.

Адміністратор може налаштовувати обмеження доступу користувачів по аналогії з користувацькими атрибутами (доступ на перегляд/зміну атрибуту в залежності від стану документа та ролі користувача) (6. Атрибути).

Завантаження відбувається по аналогії із завантаженням образу

## 7.3. Зв’зки документів

Адміністратор може використовувати при налаштуванні Типу документа спеціальний атрибут «Зв’язки між документами» (Рис. 7. 3). Даний атрибут відображає зв’язок поточного документа з іншими документами.

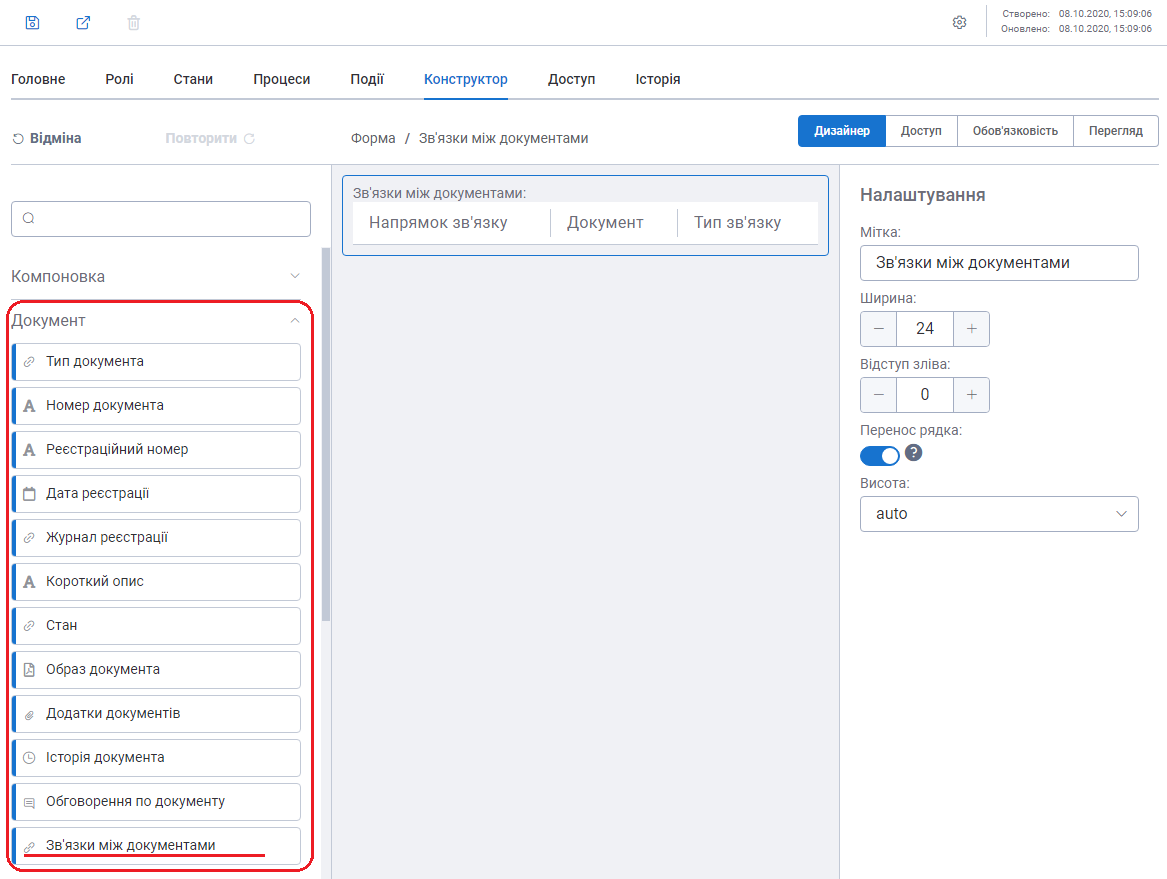


Рис. 7. 3. Атрибут «Зв’язки документів»

Опис можливих налаштувань атрибуту на вкладці «Дизайнер» описано в Таблиця 25.

Таблиця 25

| **Назва поля** | **Опис** |
| --- | --- |
| Мітка | Відображуване найменування поля |
| Ширина | Допустимі розміри мітки |
| Відступ зліва | Розмір відступу зліва |
| Перенос рядка | Перемикач для переносу значення в рядку |
| Висота | Вказується висота рядка. Можливі варіанти:   * auto * px * % |

Адміністратор може налаштовувати обмеження доступу користувачів по аналогії з користувацькими атрибутами (доступ на перегляд/зміну атрибуту в залежності від стану документа та ролі користувача) (6. Атрибути).

## 7.5. Обговорення по документу (чат)

Адміністратор може використовувати при налаштуванні Типу документа спеціальний атрибут «Обговорення по документу» (Рис. 7. 4). Даний атрибут відображає у вигляді чату список повідомлень користувачів, що беруть участь в обробці документа.

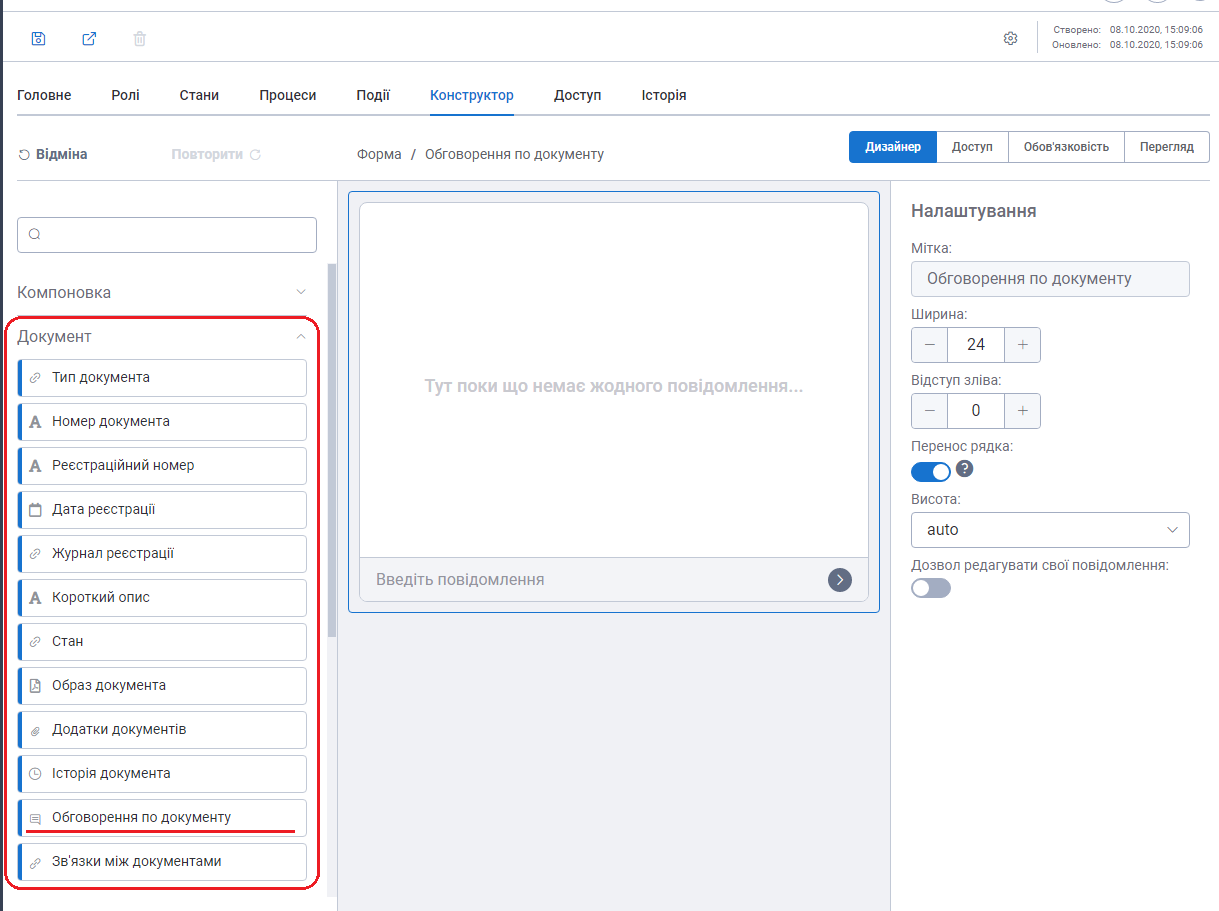


Рис. 7. 4. Атрибут «Обговорення по документу»

Опис можливих налаштувань атрибуту на вкладці «Дизайнер» описано в Таблиця 26.

Таблиця 26

| **Назва поля** | **Опис** |
| --- | --- |
| Мітка | Відображуване найменування поля |
| Ширина | Допустимі розміри мітки |
| Відступ зліва | Розмір відступу зліва |
| Перенос рядка | Перемикач для переносу значення в рядку |
| Висота | Вказується висота рядка. Можливі варіанти:   * auto * px * % |
| Дозвіл редагувати свої повідомлення | Перемикач, що дозволяє користувачу редагувати власні повідомлення |