Создание робота по распознаванию PDF-документа

1. Базовые понятия

В данном разделе определяется ряд базовых понятий, которые будут использоваться в дальнейшем (рисунок 1.1):

- *Рабочая область* – центральное окно в программе PIX Studio после создания проекта
- *Окно активностей* – левое окно в программе PIX Studio после создания проекта
- *Активности* – базовые функции, из которых конструируется робот
- *Группа активностей* – активности, объединённые единой идей своего функционала. В дальнейшем для ясности активности будут именоваться следующим образом «Группа активностей/Название активности»
- *Использовать/применить активность* – перетащить активность из окна активностей в рабочую область. Для выполнения данного действия требуется нажать ЛКМ на активность и, зажав ЛКМ, перенести курсор в рабочую область, а затем отпустить ЛКМ

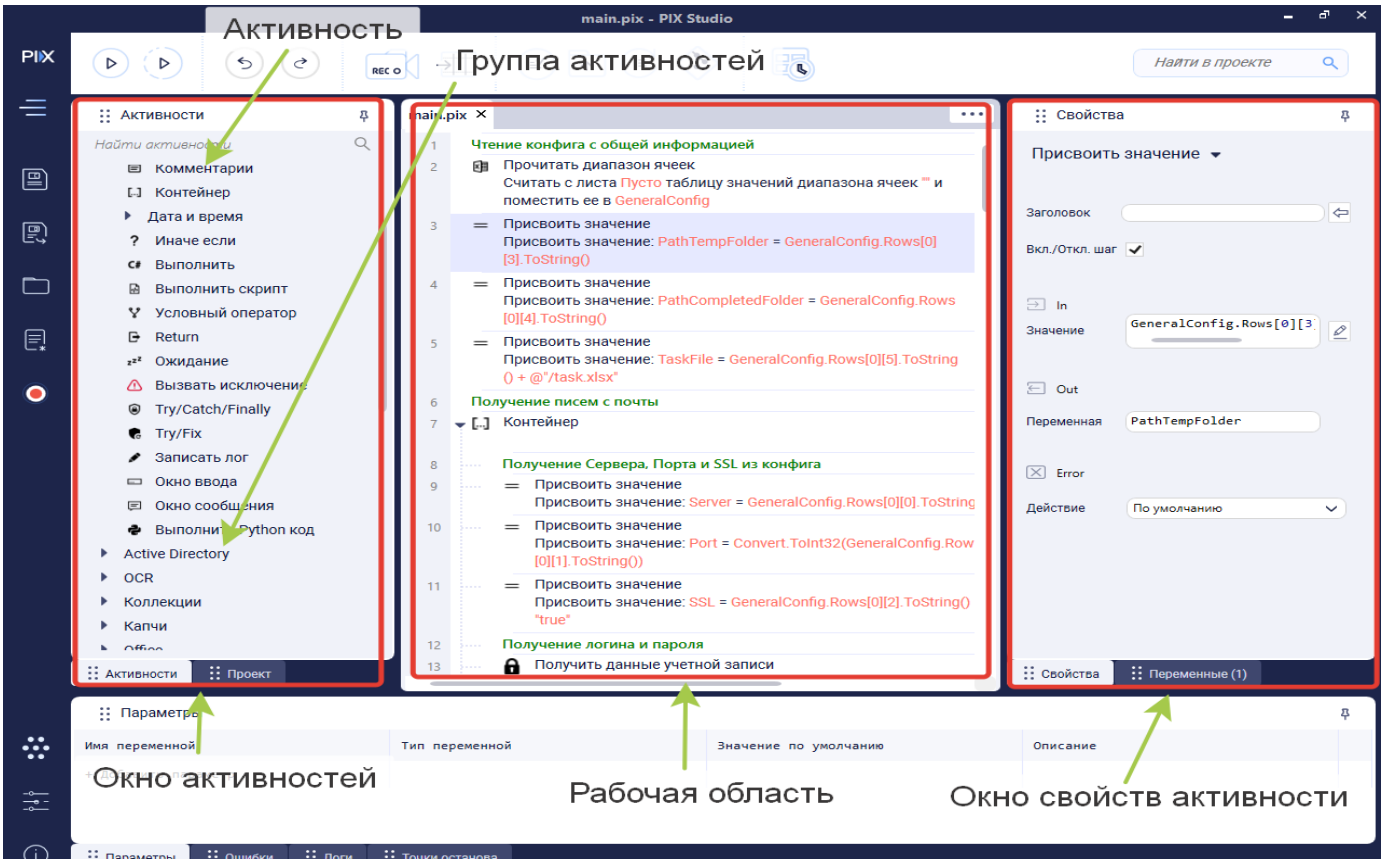


Рисунок 1.1 Окно редактора PIX Studio

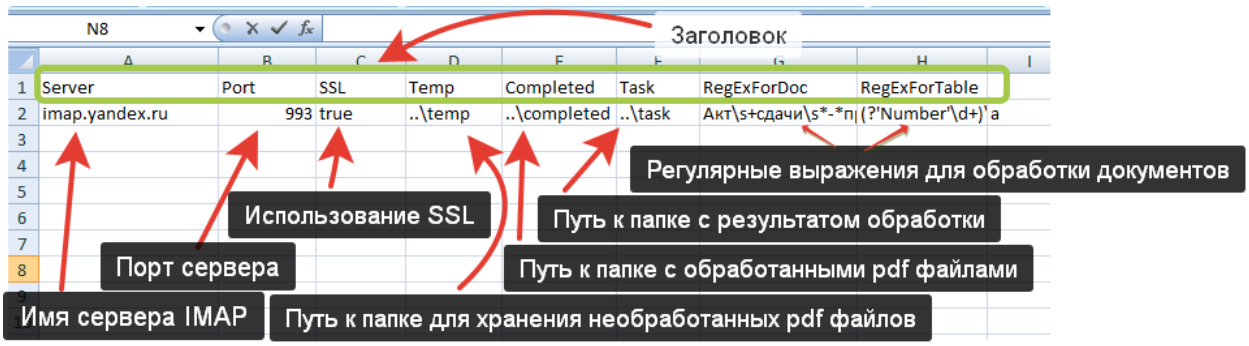
2. Постановка задачи

Робот должен проверить почту, в случае наличия на почте непрочитанных писем, робот скачивает вложения непрочитанных писем, в теме которых присутствует опознавательное слово, например «договор». Затем робот анализирует полученные PDF документы и считывает из них необходимые поля и содержимое таблиц. Затем считанные данные заносятся в excel таблицу в соответствующие поля.

3. Предварительные действия

Прежде чем приступить к созданию робота, желательно выполнить ряд подготовительных действий. В первую очередь, создаётся дополнительный excel документ, который в дальнейшем будет именоваться, как «*конфиг*». В данные документ заносится информация, которая необходима для работы робота, но может быть изменена сторонними действиями. Для данной задачи конфиг может содержать следующую информацию (Рисунок 3.1):

- Параметры для подключения к почте по протоколу IMAP (имя сервера, порт, использование SSL). Данные параметры зависят от использующейся почты и могут быть без особо труда найдены на страницы поддержки почтового сервиса или с помощью запроса в поисковые системы: «Название почтового сервиса IMAP». Так же необходимо убедится, что в настройках почтового адреса включена поддержка IMAP.
- Относительные пути к необходимым папкам (путь к папке с необработанными файлами, путь к паке с результатом обработки и т.д.)
- Регулярные выражения для обработки текста PDF-документов.



3.1. Пример конфига для робота.

Помимо конфига необходимо убедится в наличие шаблона excel таблицы, куда будут занесены считанные данные. Пример такой таблицы можно увидеть на рисунке 3.2.

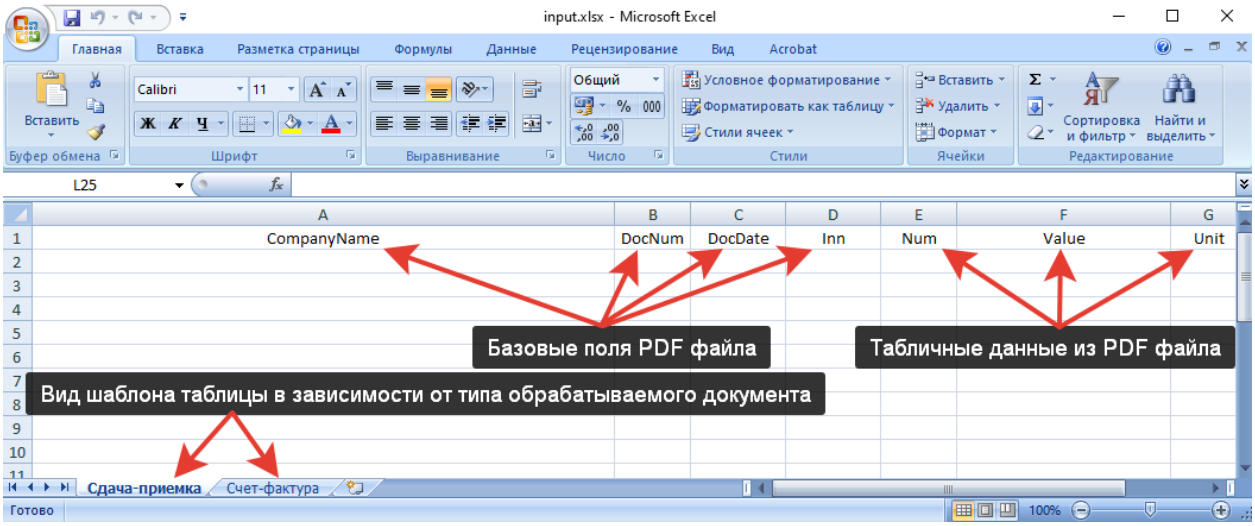


Рисунок 3.2. Пример шаблона excel таблицы.

Так же стоит написать регулярные выражения для обработки считанного из PDF текста. Примеры регулярных выражений для обработки текста и таблицы из PDF документа представлены на рисунке 3.3.

Для написания регулярных выражений можно использовать сайт «regex101.com» или какой-нибудь подобный.

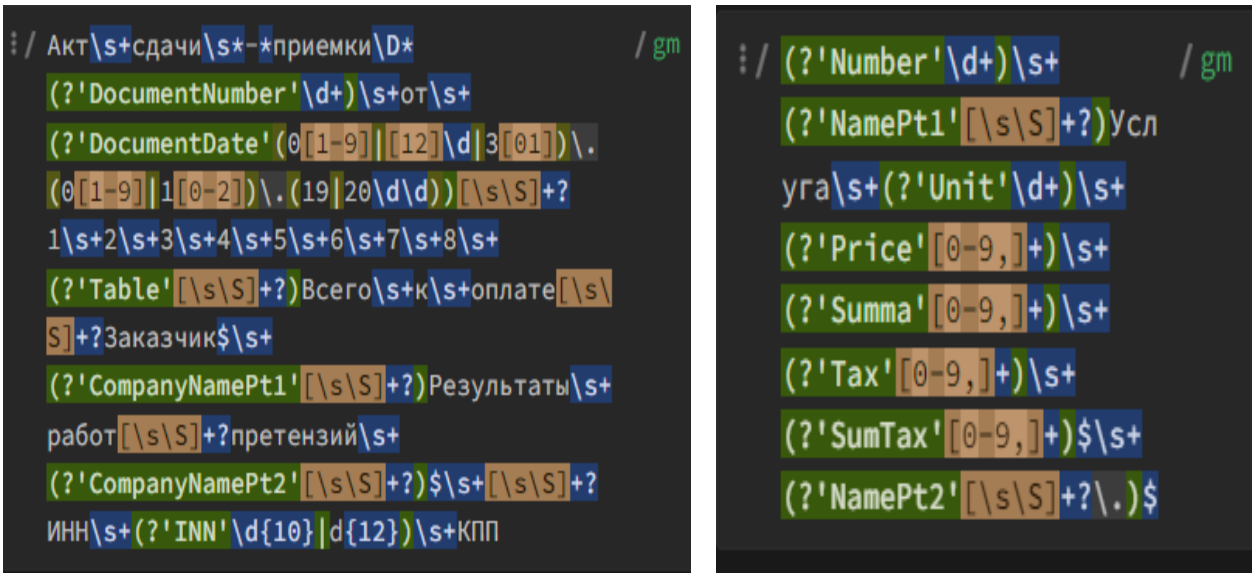


Рисунок 3.3 Пример регулярных выражений для обработки текста (слева) и таблицы (справа)

4. Работа с PIX Studio

Теперь необходимо запустить PIX Studio и создать новый проект. Для этого сначала необходимо нажать на панели слева на иконку блокнота со звёздочкой в левом нижнем углу. В появившемся окне выбирается пункт проект (Рисунок 4.1). PIX Studio попросит ввести название проекта и его расположение.

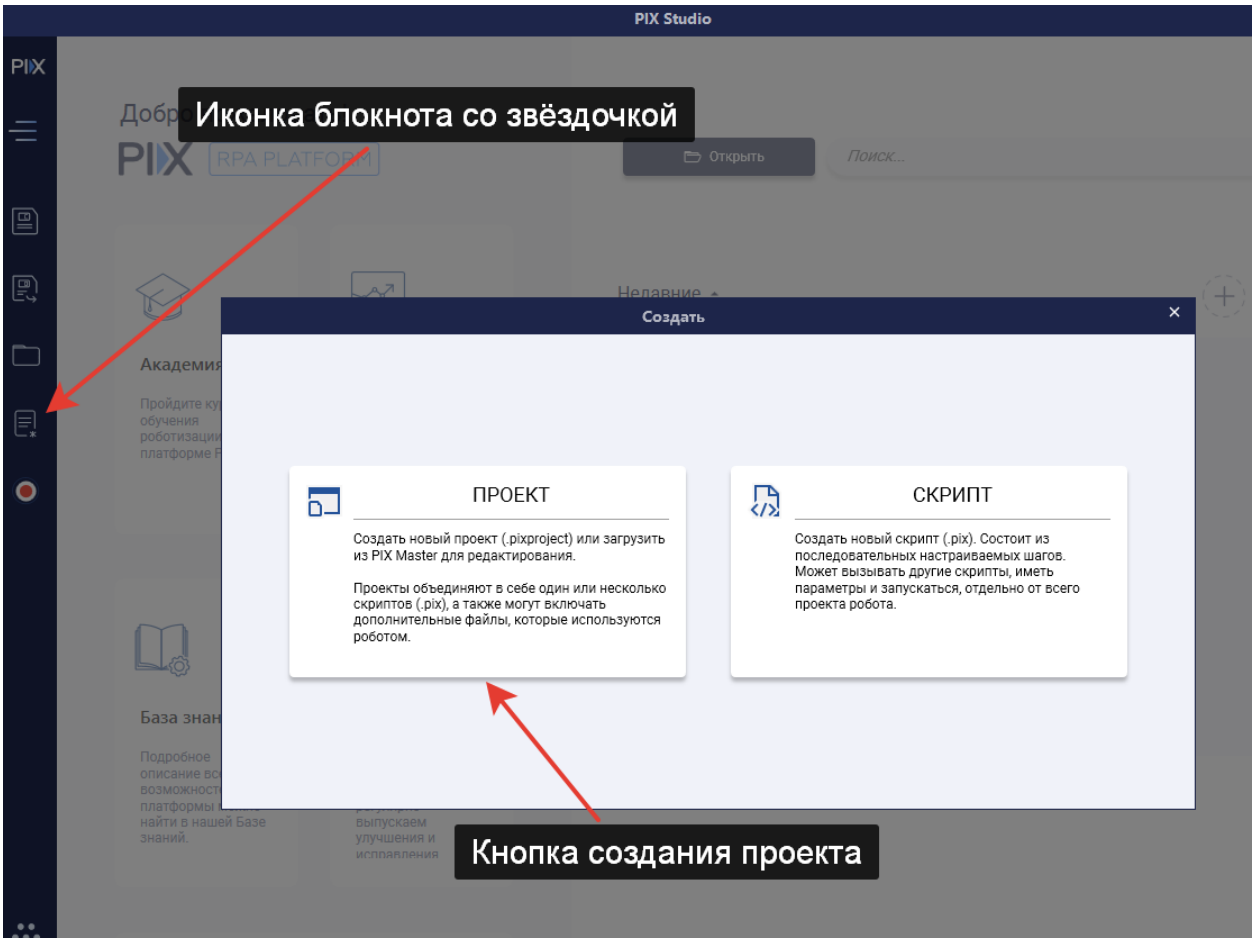


Рисунок 4.1. Создание проекта

4.1. Чтение конфига

В первую очередь необходимо прочитать конфиг. Для этого необходимо использовать активность *«Office/Excel/Прочитать диапазон ячеек»*. В окне свойств данной активности заполняется поля (рисунок 4.1.1):

- *Путь к файлу* – строка, содержащая полный или относительный путь до файла с конфигом.

- *Диапозон* – строка, которая указывает диапазон ячеек, которые необходимо считать, если необходимо прочитать все непустые ячейки с листа используется строка «"»».
- *Таблица* – Имя переменной, в которую будет помещён результат.

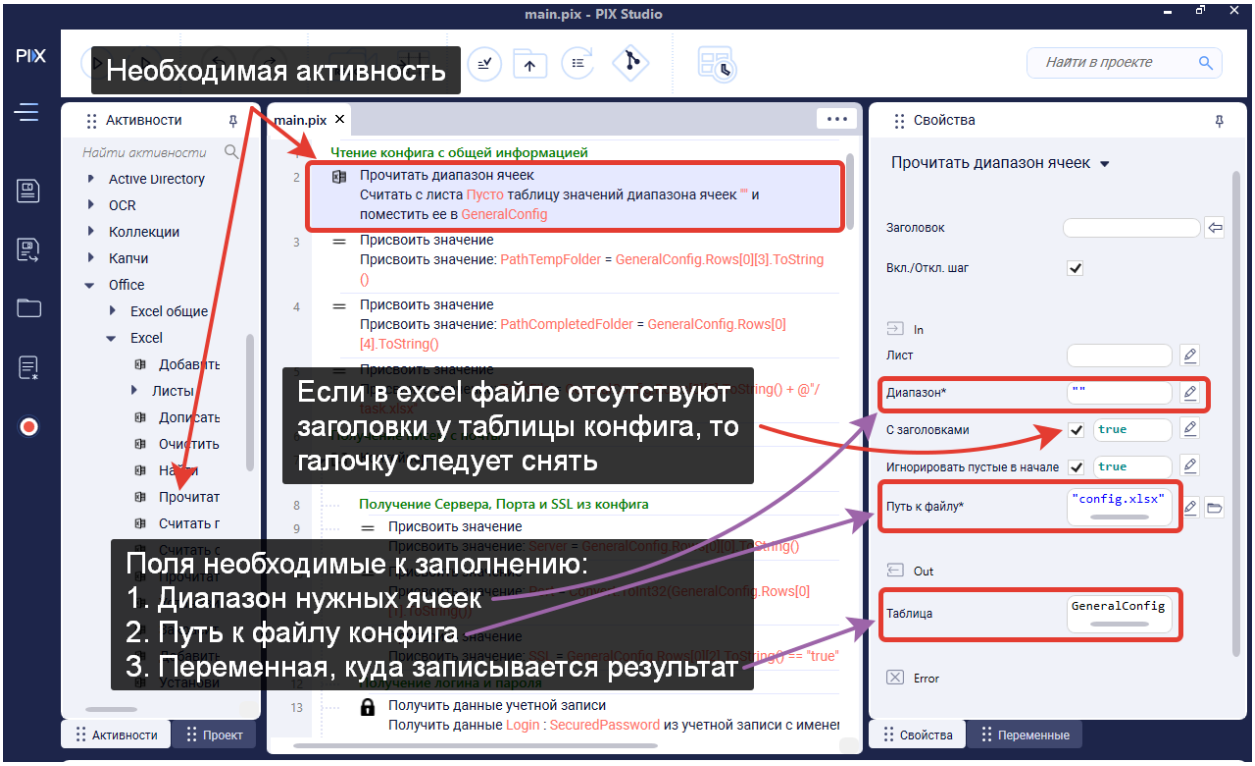


Рисунок 4.1.1. Считывание конфига.

Затем с помощью активностей *«Базовые/Присвоить значение»* данные из таблицы конфига помещаются в отдельные переменные. Для доступа к соответствующей ячейке таблицы используется следующие C# выражение: *«Переменная_с_таблицей.Rows[Индекс_Строки][Индекс_Столбца].ToString()»*. Причём нумерация строк и столбцов начинается с 0. Так как значение порта требуется хранить в численном виде, то дополнительно к переменной со значением порта применяется C# функция *«Convert.ToInt32()»*.

В окне свойств данной активности заполняются поля (Рисунок 4.1.2):

- *Значение* – значение, которое необходимо сохранить в переменной.
- *Переменная* – имя переменной, куда необходимо сохранить результат.

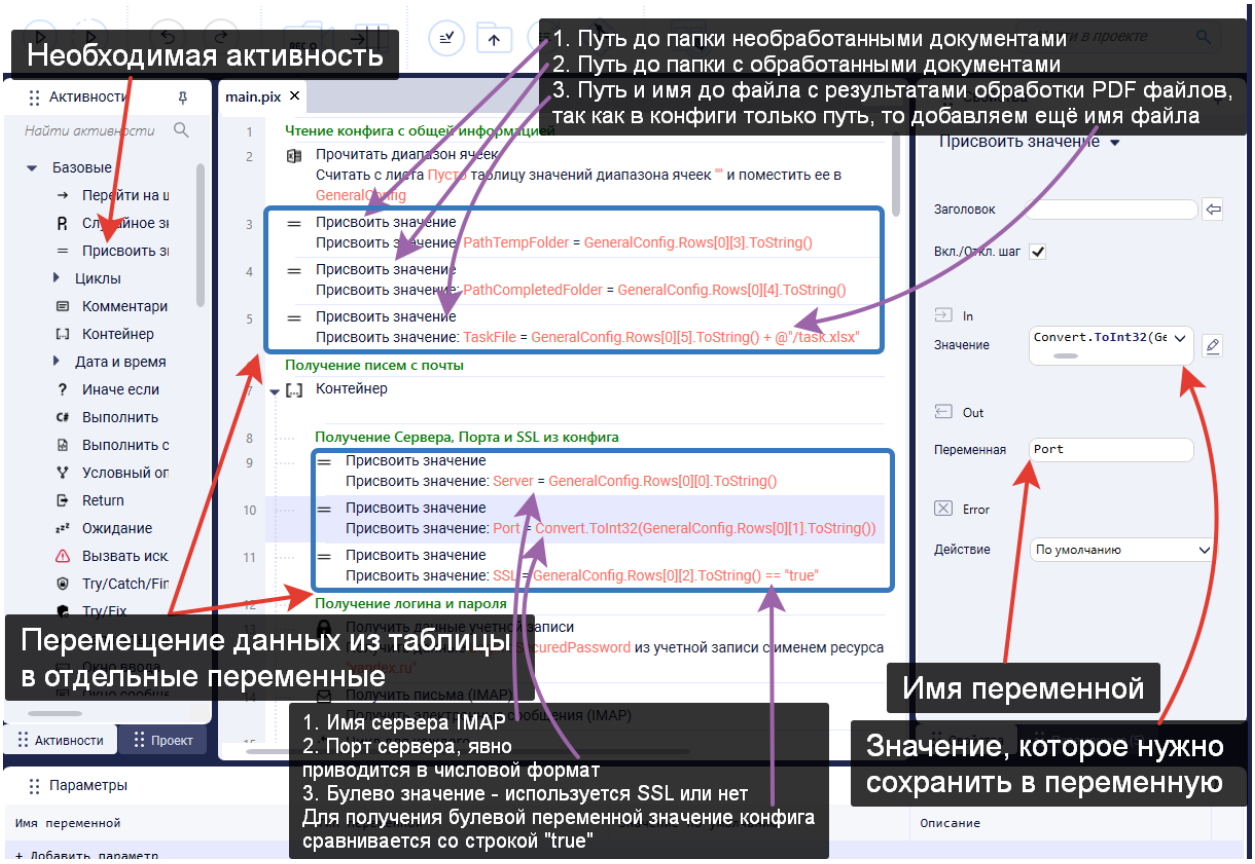


Рисунок 4.1.2. Сохранение значений в отдельные переменные.

4.2. Получение писем с почты и сохранение вложений

Изначально нужно сохранить логин и пароль от почтового ящика в Windows Credentials. Для используется активность «*Windows Credentials/Создать учетную запись*». Данная активность не войдёт в конечного робота, поэтому её можно создать и активировать в отдельном проекте, либо создать в текущем, но после первого запуска работа её можно удалить. В свойствах активности необходимо заполнить следующие поля (рисунок 4.2.1):

- *Имя ресурса* – строка, по которой в дальнейшем можно будет получить данный логин и пароль.
- *Логин* – строка с логином почтового ящика.
- *Пароль* – строка с паролем от почтового ящика.

Для получения сохраненных логина и пароля используется активность *«Windows Credentials/Получить данные учетной записи»*. В свойствах активности необходимо заполнить аналогичные поля (рисунок 4.2.2):

- *Имя ресурса* – строка, по которой были сохранены необходимый логин и пароль.
- *Логин* – переменная куда будет сохранён логин.
- *Пароль* – переменная, куда будет сохранён пароль в защищённом виде.

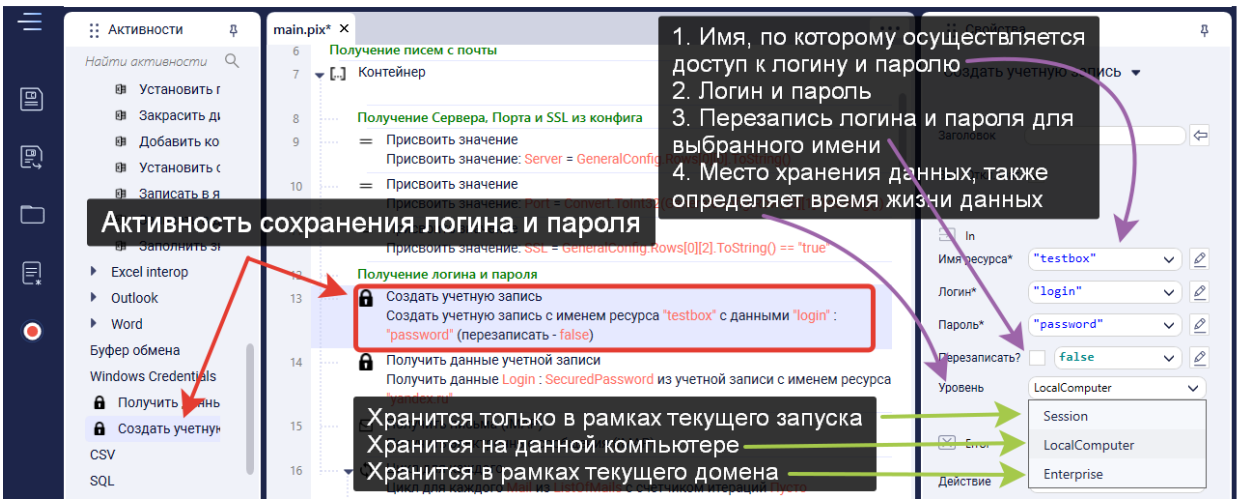


Рисунок 4.2.1. Сохранения логина и пароля в память устройства.

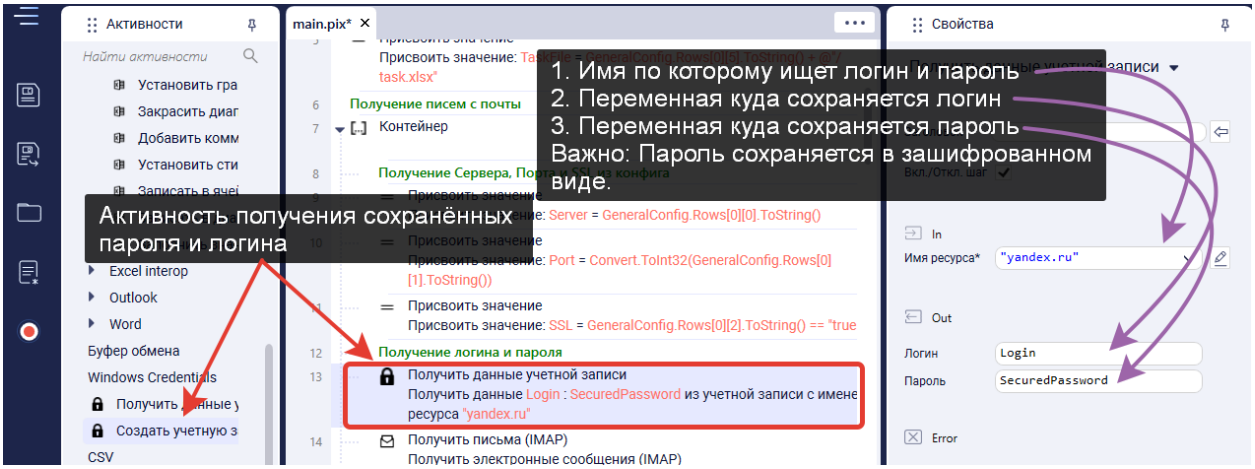


Рисунок 4.2.2. Получение логина и пароля из памяти устройства.

После получения логина и пароля, с помощью активности *«Email/Получить письма (IMAP)»* считываются письма из почтового ящика. В свойствах активности необходимо заполнить (рисунок 4.2.3):

- Поля, связанные с сервером IMAP (*Порт*, *Сервер*, *SSL*).
- Данные для входа в почтовый ящик (*Логин* и либо *Пароль* (в явном виде), либо *Безопасный строковый пароль* (в защищённом)).
- *Название папки*, какие письма читать, что делать с прочитанными.
- *Письма* – переменная, в которую записывается результат.

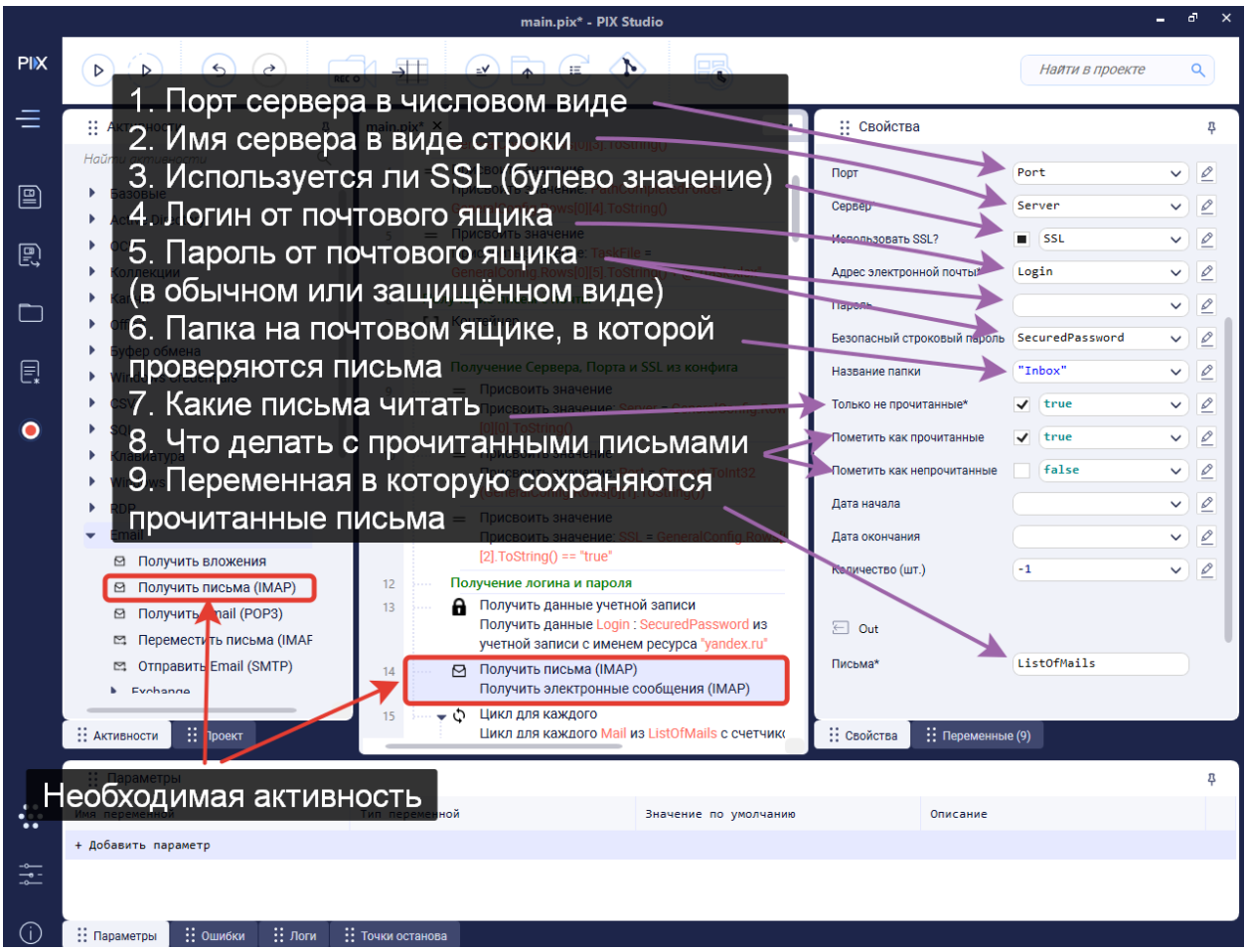


Рисунок 4.2.3. Получение писем.

Следующим шагом запускается цикл по полученным письмам с помощью активности *«Базовые/Циклы/Цикл для каждого»*. Внутри цикла каждое письмо проверяется на содержание в теме письма опознавательного слова. Для этого используется активность *«Базовые/Условный оператор»* и C# функция:

«Письмо.Subject.ToLower().Contains(“Слово_в_нижнем_регистре”)»

В случае если письмо содержит необходимое опознавательное слово в теме письма, то вложения из данного письма сохраняются в необходимую папку с помощью активности «*Email/Получить вложения»*. В свойствах активности необходимо заполнить поля: *Письмо* – переменная с текущим письмом, *Папка* – путь до папки, в которую нужно сохранить вложения, а именно папка с необработанными PDF файлами (рисунок 4.2.4).

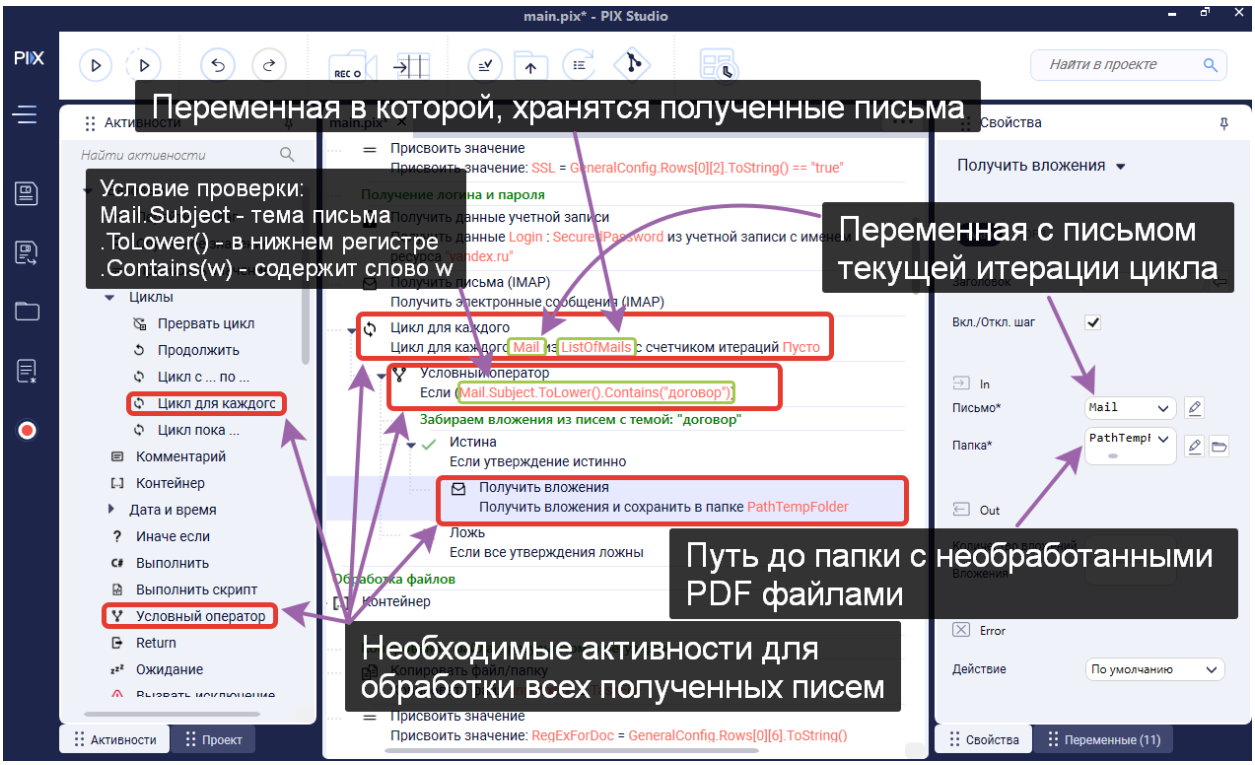


Рисунок 4.2.4. Сохранение вложений.

4.3. Обработка PDF файлов.

Теперь с помощью активности *«Файлы/Копировать файл/папку»* файл шаблона excel таблицы копируется в нужное место. Данная активность применяется на случай, если конечного файла excel таблицы с обработанными результатами ещё не создано. В свойствах активности указывается:

- *Путь откуда* – путь до шаблона, в том числе и название файла шаблона.
- *Путь куда* – путь до конечного файла, в том числе и название.
- *С удалением файла/паки откуда* – должно быть false, так как шаблон переносить не надо.
- *Перенос с заменой* – должно быть false, так как не нужно заменять конечный файл, в случае его существования.

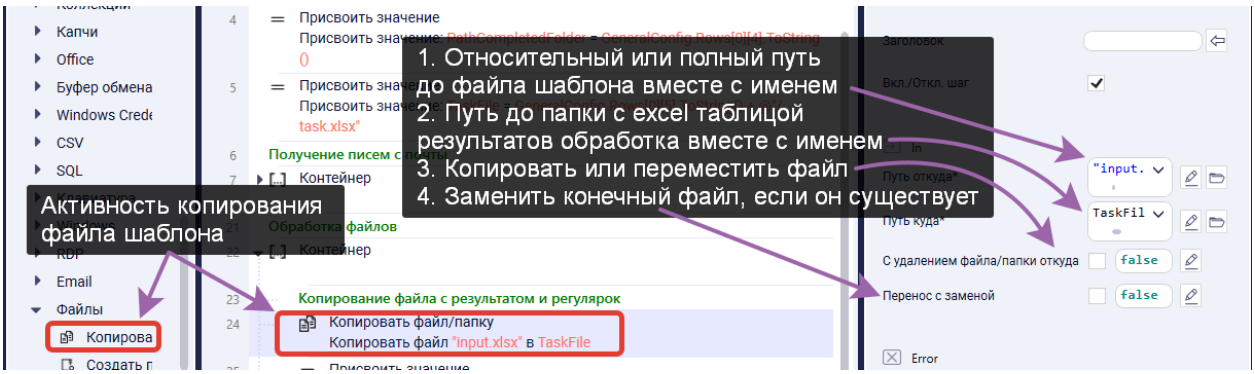


Рисунок 4.3.1. Копирования шаблона excel таблицы.

Далее с помощью активности *«Базовые/Присвоить значение»* сохраняются регулярные выражения из таблицы конфига в отдельные переменные. Одно регулярное выражения для обработки всего текста PDF файла, другое для обработки табличных значений из PDF файла.

Затем с помощью активности *«Файлы/Получить пути к файлам/каталогам»* получаются пути к необработанным PDF файлам. В свойствах активности указывается: *Путь* – путь до папки, в которой хранятся необработанные PDF файлы, *Список* – переменная, в которую будут сохранены пути ко всем PDF файлам (Рисунок 4.3.2).

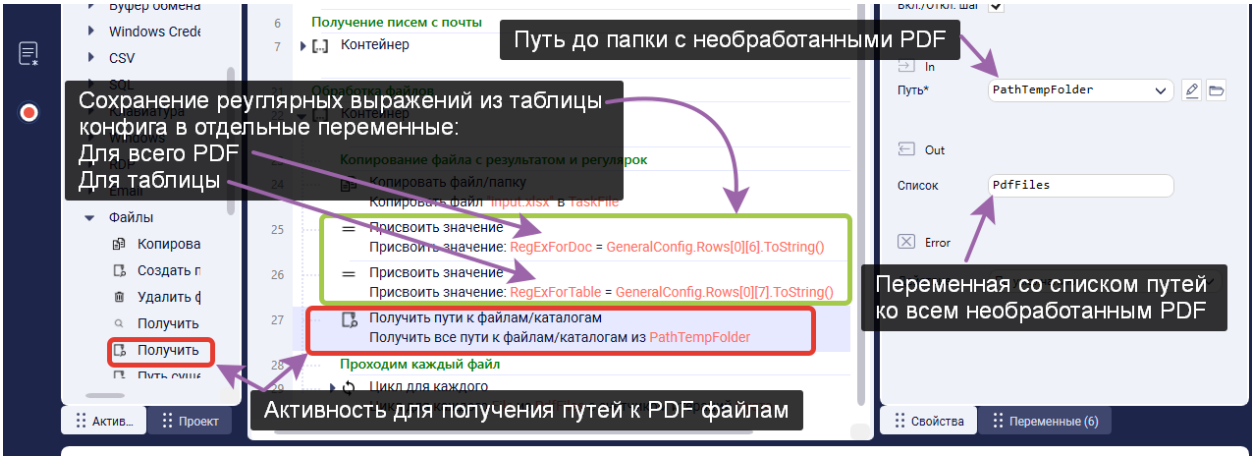


Рисунок 4.3.2. Получение путей ко всем необработанным PDF.

Следующим шагом запускается цикл по полученным путям файлов PDF. Для каждого файла с помощью активности *«PDF/Текст из PDF»* считывается весь текст из PDF файла. В свойствах активности необходимо указать: *Файл PDF* – путь до текущего файла, *Результат* – переменная, в которую будет сохранён весь текст. А также создаётся с помощью активности *«Коллекции/Словарь/Создать словарь»* словарь для хранения

необходимых данных из текста PDF. В свойствах активности необходимо заполнить поле: *Словарь* – переменная, в которой хранится сам словарь (Рисунок 4.3.3).

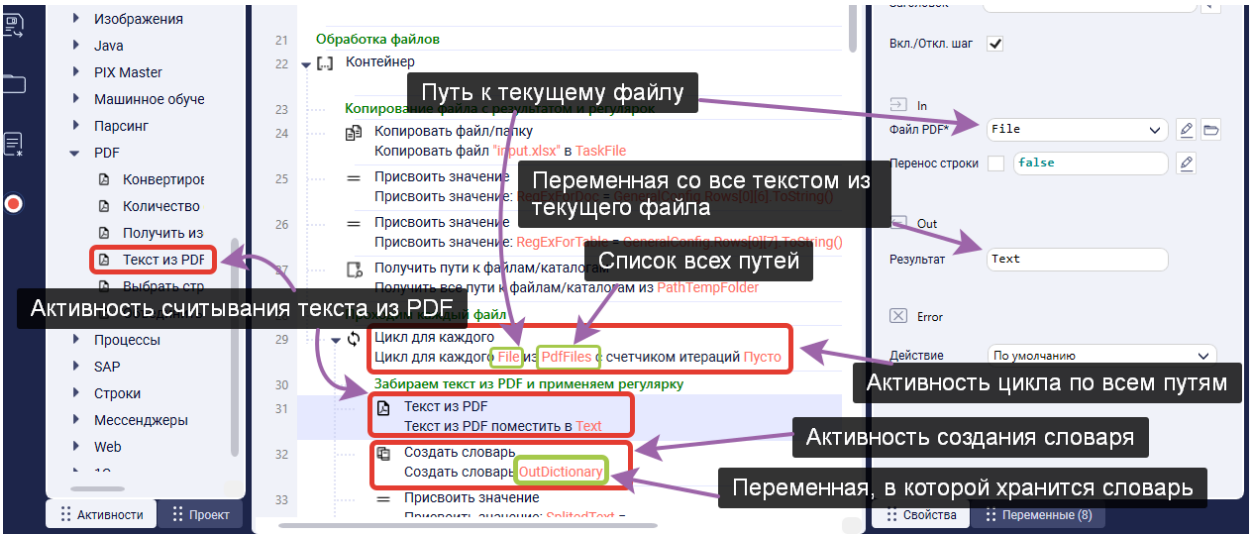


Рисунок 4.3.3. Считывание текста из PDF файла

Далее с помощью активности «Базовые/Присвоить значение» и C# функции *«System.Text.RegularExpression.Regex.Match(Текст, Регулярное выражения, Опции_регулярного_выражения)»* применяется регулярное выражения к считанному тексту. А также с помощью всё той же активности *«Базовые/Присвоить значение»* создаётся переменные для хранения текста таблицы из PDF и переменная для хранения типа текущего документа (Рисунок 4.3.4). А также в случае если регулярное выражение способно обрабатывать несколько типов файлов с помощью активности *«Базовые/Присвоить значение»* и C# функции *«Результат_регулярного_выражения.Groups[“Уникальное_имя_группы”]»* создаётся булева переменная для проверки какой тип файла обрабатывается.

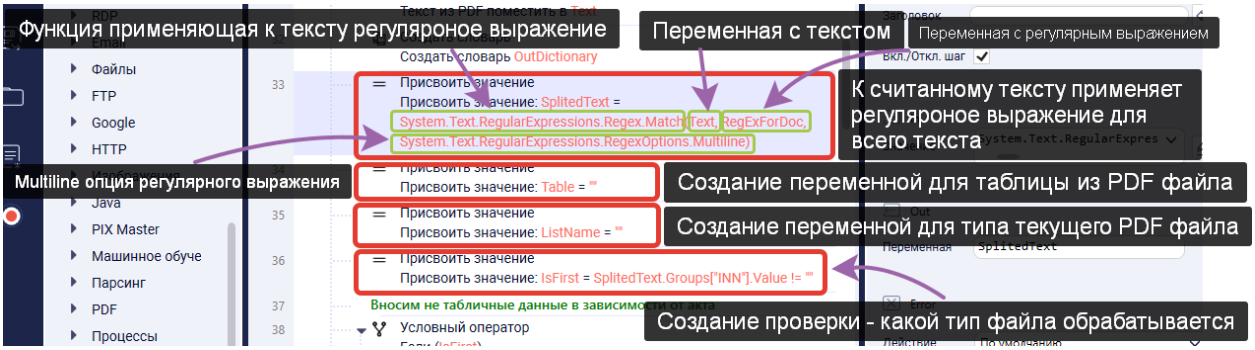


Рисунок 4.3.4. Создание переменных и применение регулярного выражения.

Причём создания именно булевой переменной C# функция нужно сравнить со строкой «””», так как C# функция вернёт строку «””», если данной группы нет в результате регулярного выражения, а значит тип файла относится к противоположному типу, который содержит в себе «Уникальное_имя_группы».

Дальше с помощью активностей *«Базовые/Условный оператор»*, *«Базовые/Присвоить значения»* и активности *«Коллекции/Словарь/Задать значение для ключа»* в зависимости от типа документа сохраняем таблицу из документа в отдельную переменную, название типа документа и заносим необходимые данные в словарь каждый со своим ключом. В свойствах активности *«Коллекции/Словарь/Задать значение для ключа»* необходимо указать (Рисунок 4.3.5):

- *Ключ* – строка, которой именуется нужное значение из документа.
- *Значение* – значение соответствующей группы регулярного выражения.
- *Словарь* – переменная словаря.

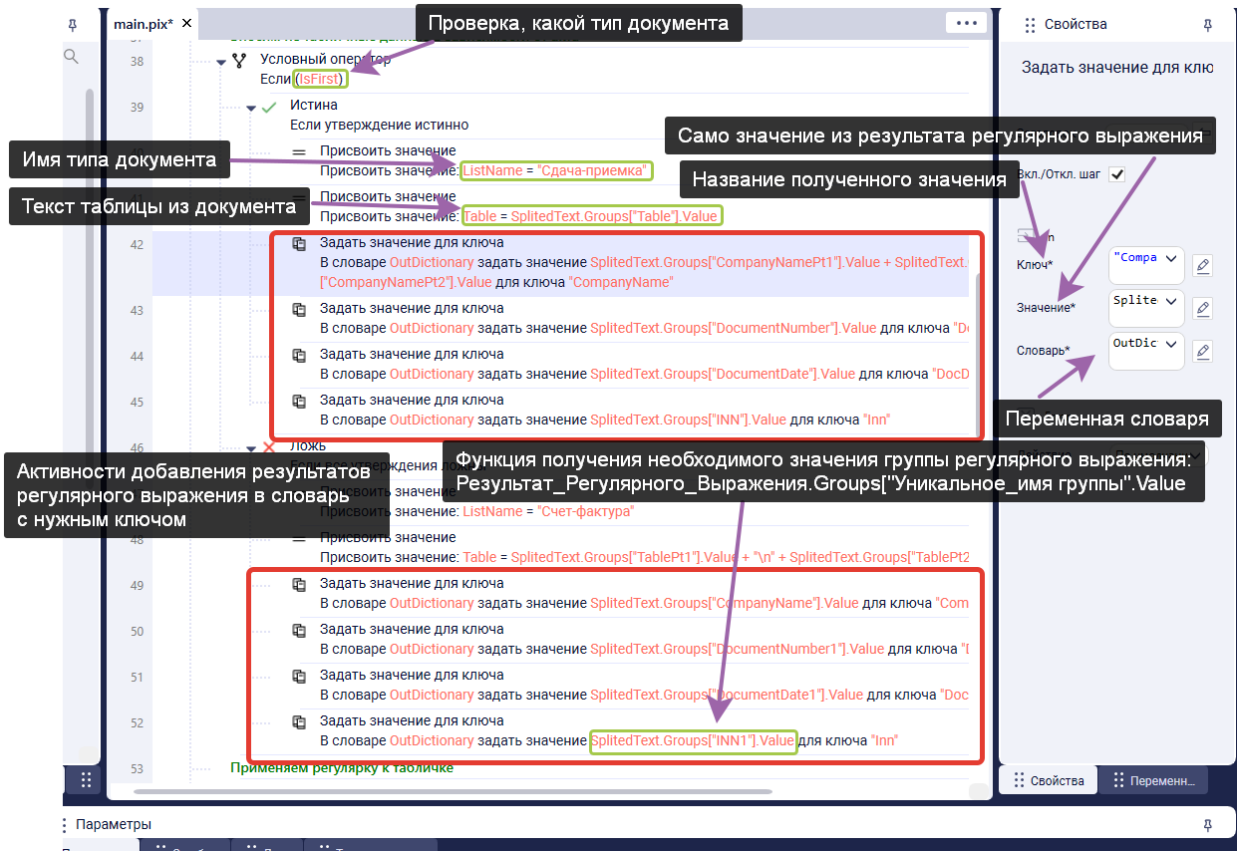


Рисунок 4.3.5. Занесение результатов регулярного выражения в словарь

Дальше применяется регулярное выражение для текста таблицы с помощью активности *«Базовые/Присвоить значение»* и C# функции *«System.Text.RegularExpression.Regex.Matches(Текст, Регулярное_выражения, Опции_регулярного_выражения)»*. А также создаётся пустая таблица с помощью активности *«Коллекции/Таблица/Создать таблицу»*, а в свойствах активностей необходимо заполнить: Таблица – переменная, в которой будет храниться таблица (Рисунок 4.3.6).

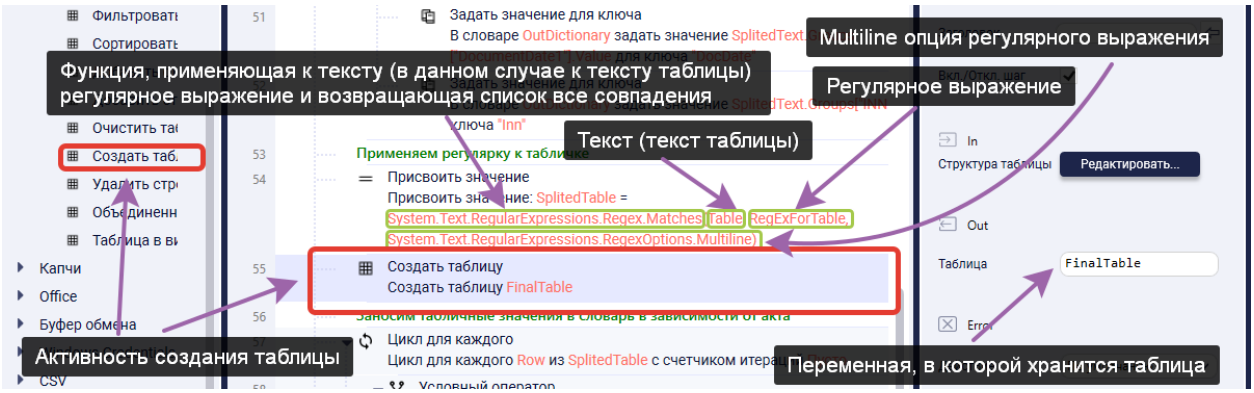


Рисунок 4.3.6. Применение регулярного выражения к тексту таблицы

Затем запускается цикл по результатам применения регулярного выражения к тексту таблицы из PDF. В зависимости от типа файла, нужные значения на каждой итерации заносятся в словарь. А в конце итерации все значения словаря записываются в конец созданной ранее пустой таблицы с помощью активности *«Коллекции/Таблица/Добавить строку»*. В свойствах активности необходимо указать:

- *Строка* – переменная, в которой хранится словарь
- *Позиция* – куда записывать данные, в начало или в конец. В данном случае должна быть равна End.
- *Создать столбцы* – автоматическое создание недостающих столбцов. В данном случае должно быть true.
- *Таблица* – переменная, в которой хранится таблица.

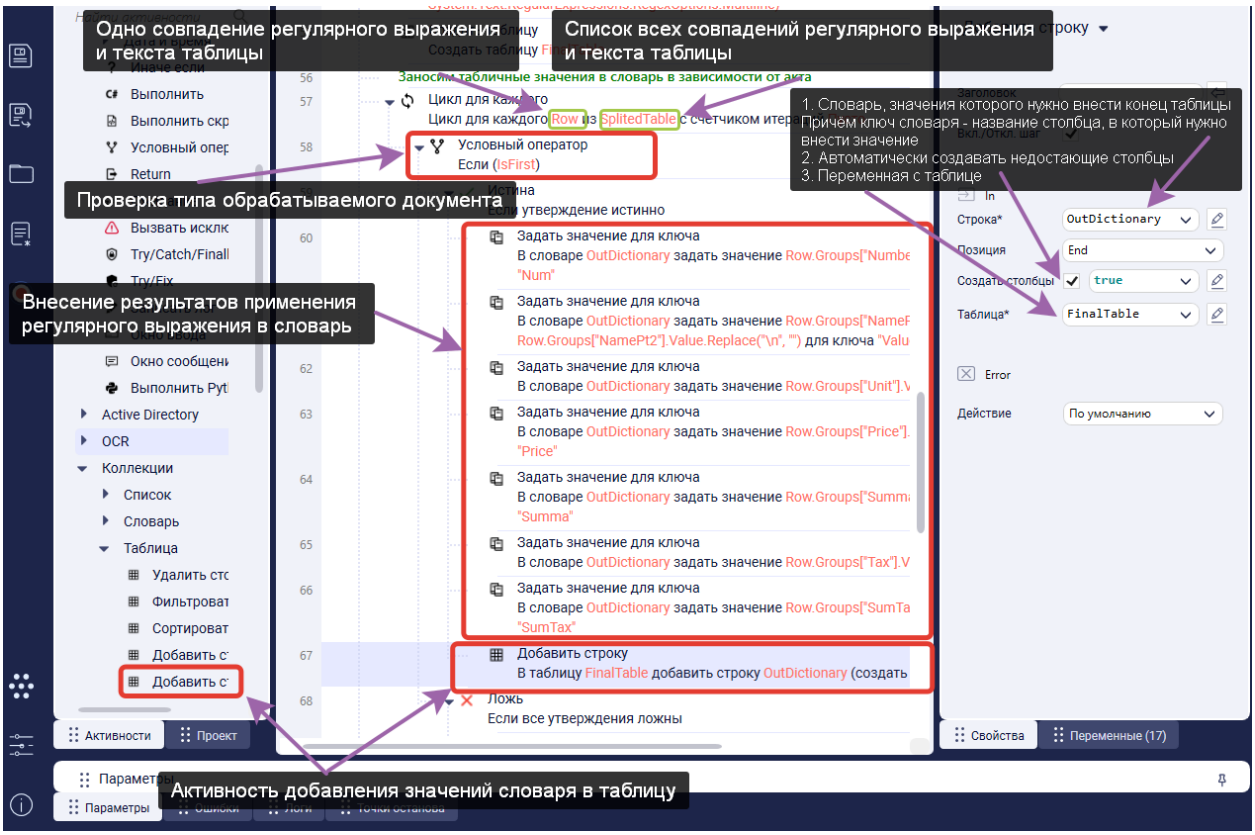


Рисунок 4.3.7. Занесение результатов обработки таблицы в словарь

После завершения цикла с помощью активности *«Office/Excel/Дописать диапазон»* полученная таблица с результатами обработки файла записывается в конец excel таблицы с результатами обработки PDF файлов, а сам файл переносится из папки с необработанными PDF в папку с обработанными PDF с помощью активности *«Копировать файл/папку»* и робот переходит к обработке следующего файла.

В свойствах активности *«Office/Excel/Дописать диапазон»* необходимо указать: *Лист*, *Таблица*, *Путь к файлу* (Рисунок 4.3.8).

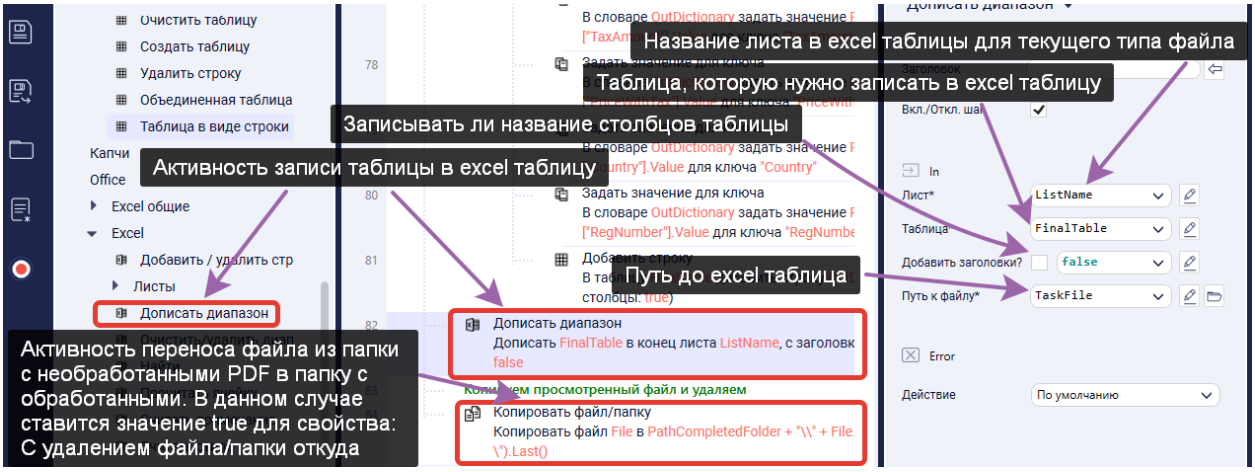


Рисунок 4.3.8. Запись данных в excel и перенос файла.

Литература

1. https://knowledgebase.pixrpa.ru/actions/Excel/ReadRange
2. https://knowledgebase.pixrpa.ru/actions/Base/Assign
3. https://knowledgebase.pixrpa.ru/actions/Base/Assign
4. https://knowledgebase.pixrpa.ru/actions/Email/GetIMapMailMessages
5. https://knowledgebase.pixrpa.ru/actions/Base/LoopForEach
6. https://knowledgebase.pixrpa.ru/actions/Base/If
7. https://knowledgebase.pixrpa.ru/actions/Email/GetAttachments
8. https://knowledgebase.pixrpa.ru/actions/Files/CopyFileCatalog
9. https://knowledgebase.pixrpa.ru/actions/Files/GetListFilesOrCatalogs
10. https://knowledgebase.pixrpa.ru/actions/PDF/ReadPDF
11. https://knowledgebase.pixrpa.ru/actions/Dictionary/CreateDictionary
12. https://knowledgebase.pixrpa.ru/actions/Dictionary/AddKey
13. https://knowledgebase.pixrpa.ru/actions/DT/CreateTable
14. https://knowledgebase.pixrpa.ru/actions/DT/AddRow
15. https://knowledgebase.pixrpa.ru/actions/Excel/AppendRange
16. https://regex101.com/