

PDF В TASK: Извлечение Реквизитов В Таблицу

Дата актуализации: 2026-05-14

TASK может читать PDF как read-only источник через PyMuPDF. Это подходит для векторных PDF с текстовым слоем, когда нужно извлечь реквизиты в DOCX/XLSX отчёт. PDF не редактируется, точные замены для PDF не применяются.

Куда Класть PDF

Положите PDF в:

input\

В GUI используйте:

ЗАДАЧА ДЛЯ OPENAI
ЗАДАЧА ДЛЯ GEMINI

или:

Специализированные команды -> Быстрая задача

Поле **Страниц PDF** в **Дополнительно** ограничивает чтение первых страниц PDF. По умолчанию читаются первые 5 страниц.

Если нужен чистый табличный результат по готовому образцу, положите DOCX/XLSX шаблон таблицы в **input**, включите поле **Чистая таблица по шаблону** и выберите файл в поле **Шаблон таблицы**. Пустое поле выбирает шаблон автоматически только в простом случае: во всём **input** есть ровно один DOCX/XLSX-файл и есть другие исходные файлы, например PDF.

Когда PDF Подходит

Подходит:

- PDF содержит выделяемый текст;
- нужно извлечь таблицу реквизитов, а не править документ;
- реквизиты находятся на первых страницах, титульных листах или в штампах.

Не подходит без OCR:

- PDF сделан из сканов;
- текст превращён в кривые/контуры;
- реквизиты видны только как изображение.

Чистый Экспорт По Шаблону

Режим нужен, когда полный TASK-отчёт слишком технический, а на выходе нужна простая таблица, повторяющая пользовательский образец.

Как Подготовить Файлы

Положите исходные документы в `input\` или во вложенные папки:

```
input\2019\...  
input\2024\...
```

Положите шаблон таблицы тоже в `input\`, например:

```
input\Таблица_СТП_ГП _П33 с реквизитами.docx
```

Шаблон может быть DOCX или XLSX:

- DOCX: используется первая таблица документа;
- XLSX: используется первый лист книги.

Как TASK Определяет Шаблон

TASK создаёт чистую таблицу только если включено поле:

Чистая таблица по шаблону

После этого шаблон определяется так:

1. Если в поле **Шаблон таблицы** выбран конкретный DOCX/XLSX, используется он.
2. Если поле **Шаблон таблицы** пустое, TASK ищет автошаблон среди входных файлов.
3. Автошаблон выбирается только когда во всём **input** найден ровно один DOCX/XLSX и при этом есть другие исходные файлы для обработки.
4. Если DOCX/XLSX несколько, пустой выбор не угадывает шаблон: чистый экспорт пропускается, чтобы не взять неправильную таблицу.
5. Если выбранный шаблон не найден или имеет не DOCX/XLSX-расширение, запуск завершится ошибкой.

Выбранный шаблон исключается из корпуса источников до LLM-запроса. То есть модель извлекает данные из PDF/DOCX/PPTX/XLSX-источников, но не из самой таблицы-образца.

Как Шаблон Превращается В Колонки

Для DOCX:

- берётся первая таблица;
- первая строка таблицы становится списком колонок;
- вторая строка считается примером формата значений.

Для XLSX:

- берётся первый лист;
- первая непустая строка становится списком колонок;
- следующая непустая строка считается примером формата значений.

Пример DOCX-шаблона:

Наименование	Вид градостроительной документации	Кем утверждено	Дата	номер
Ильинское сельское поселение	Об утверждении правил землепользования и застройки Ильинского сельского поселения	решение Совета Ильинского сельского поселения	от 26.06.2024	№ 94

Из этого TASK поймёт:

- на выходе должно быть ровно 5 колонок;
- ключи в `values` должны называться точно так же;
- дата должна выглядеть как `от ДД.ММ.ГГГГ`;
- номер должен выглядеть как `% N`.

Лишние поля из ответа модели, например `source_file`, `confidence` или `missing_fields`, не попадают в чистую таблицу, если таких колонок нет в шаблоне. Они остаются в полном TASK-отчёте.

Что Должна Вернуть Модель

В запросе нужно явно попросить модель вернуть значения в `values` с ключами, которые совпадают с колонками шаблона.

Минимальный пример:

Извлеки реквизиты для заполнения таблицы.

Верни одну строку на каждый исходный документ.

В `values` заполни колонки ровно с такими названиями:

- Наименование
- Вид градостроительной документации
- Кем утверждено
- Дата
- номер

Не сокращай значения основных колонок.

Если значение не найдено, оставь пустым.

В `quote` дай короткую цитату-основание.

В `notes` укажи сомнения и OCR-искажения.

`Replacements` не возвращай.

Если модель вернёт ключ `Дата`, он попадёт в колонку `Дата`. Если вернёт `date` вместо `Дата`, чистый экспорт эту колонку не заполнит, потому что имена должны совпадать с шаблоном.

Как TASK Экспортирует Чистый Результат

После LLM-прохода всегда остаются обычные технические результаты:

```
output\_doc_tasks\<timestamp>__doc_task.json
output\_doc_tasks\<timestamp>__doc_task.xlsx
output\_doc_tasks\<timestamp>__doc_task.docx
output\_doc_tasks\<timestamp>__doc_task.md
```

Если включён чистый экспорт и шаблон определён, дополнительно создаются:

```
output\_doc_tasks\<timestamp>__doc_task_clean.json
output\_doc_tasks\<timestamp>__doc_task_clean.xlsx
output\_doc_tasks\<timestamp>__doc_task_clean.docx
```

DOCX-копия создаётся только для DOCX-шаблона. Если шаблон XLSX, TASK создаст чистые **JSON** и **XLSX**.

Чистый XLSX создаётся заново с колонками шаблона. Чистый DOCX создаётся как копия DOCX-шаблона: первая таблица очищается до заголовка и заполняется новыми строками.

Пустые строки, где все колонки пустые, не попадают в чистый экспорт.

Что Постобработка Исправляет, А Что Нет

Постобработка делает только форматные операции:

- добавляет **от** к дате, если образец во второй строке начинается с **от**;
- приводит номер к виду **№ N**;
- убирает хвост **от ... № ...** из соседних колонок, если дата и номер уже вынесены в отдельные колонки.

Постобработка не исправляет смысловые ошибки:

- не угадывает правильное название поселения;
- не исправляет OCR-искажения букв;
- не выбирает между несколькими похожими объектами;
- не заполняет отсутствующие значения из внешней памяти.

Такие случаи должны оставаться в **notes**, **confidence**, **missing_fields** или обычном DOCX/XLSX/JSON отчёте TASK. Чистая таблица - это аккуратный экспорт уже извлечённых значений, а не второй скрытый LLM-редактор.

Практический Порядок В GUI

1. Очистите или подготовьте `input\`.
2. Положите исходные документы в `input\2019\`, `input\2024\` или другие подпапки.
3. Положите DOCX/XLSX-шаблон таблицы в `input\`.
4. Откройте `ЗАДАЧА ДЛЯ OPENAI` или `ЗАДАЧА ДЛЯ GEMINI`.
5. Выберите инструкцию TASK или используйте `Специализированные команды -> Быстрая задача`.
6. В запросе перечислите колонки ровно как в первой строке шаблона.
7. Включите `Чистая таблица по шаблону`.
8. В поле `Шаблон таблицы` выберите DOCX/XLSX. Пустой выбор оставляйте только если в `input\` заведомо один DOCX/XLSX-файл и есть другие исходные файлы.
9. При PDF-источниках настройте `Страниц PDF`, если реквизиты находятся дальше первых 5 страниц.
10. Запустите TASK и заберите чистые файлы из `output\_doc_tasks\`.

CLI-Пример

То же самое можно запустить из CLI:

```
runtime\python.exe system_core\doc_task_runner.py run --recursive --provider openai -  
-resume --clean-table --clean-table-template "Таблица_СТП_ГП _П33 с реквизитами.docx"
```

`--clean-table-template` принимает путь относительно `input\` или абсолютный путь к DOCX/XLSX-шаблону.

Шаблон: Реквизиты Проекта

Текст можно вставить в `Быстрая задача` и при необходимости сохранить в кэш.

Извлеки реквизиты проекта с первых страниц документов.

Верни одну строку на каждый исходный файл или на каждый явно отдельный проект, если в одном файле их несколько. Ничего не выдумывай: если поле не найдено, оставь пустым и укажи это в `missing_fields`.

В `values` заполни колонки:

- `source_file`
- `project_name`
- `project_code`
- `object_name`
- `object_address`
- `customer`
- `developer`
- `designer`
- `contractor`
- `document_stage`
- `document_set_or_volume`
- `contract_number`
- `date`
- `source_pages`
- `confidence`
- `missing_fields`

В `quote` дай короткую цитату-основание. В `notes` укажи неоднозначности.

`Replacements` не возвращай.

Шаблон: Реквизиты Документа

Извлеки реквизиты документа с первых страниц.

Верни одну строку на каждый документ. Не извлекай реквизиты проекта, если они не относятся к самому документу. Если поле не найдено, оставь пустым.

В values заполни колонки:

- source_file
- document_title
- document_code
- document_number
- revision
- version
- approval_status
- organization
- author_or_responsible
- date
- pages_or_volume
- source_pages
- confidence
- missing_fields

В quote дай короткую цитату-основание. В notes укажи неоднозначности.
Replacements не возвращай.

Шаблон: По Пользовательской Таблице

Если таблица нужна только один раз, перечислите колонки прямо в запросе. Если таблица повторяемая, положите DOCX/XLSX-шаблон в `input\` рядом с PDF, включите `Чистая таблица по шаблону` и выберите этот шаблон. Подробный порядок описан выше в разделе `Чистый Экспорт По Шаблону`.

Извлеки реквизиты по заданной ниже структуре таблицы.

Колонки результата:

[вставить список колонок]

Правила:

- ключи в `values` должны точно совпадать с названиями колонок шаблона;
- одна строка на каждый найденный объект/документ/проект;
- значения брать только из предоставленных блоков;
- если значение не найдено, оставить пустым;
- спорные случаи помечать в `confidence` и `notes`;
- в `quote` привести короткую цитату-основание;
- в `source_pages` указать страницы или блоки, откуда взяты значения;
- `replacements` не возвращать.

Шаблон: Строго По Образцу

Этот вариант полезен, когда модель начинает сокращать значения. Вставьте свои колонки и пример формата.

Извлеки реквизиты для заполнения таблицы. Таблица/описание колонок является ОБРАЗЦОМ точного стиля результата.

Верни одну строку на каждый исходный документ.

ОБЯЗАТЕЛЬНЫЕ КОЛОНКИ values, ровно с такими названиями:

- [колонка 1]
- [колонка 2]
- [колонка 3]

КРИТИЧЕСКИ ВАЖНО: не сокращай значения основных колонок.

Формат значений должен быть как в образце:

- [колонка 1]: "[пример полного значения]"
- [колонка 2]: "[пример полного значения]"
- [колонка 3]: "[пример полного значения]"

Правила:

- бери значения только из предоставленных блоков;
- если OCR искажает отдельные буквы, но реквизиты читаются, извлекай значение и отметь сомнение в notes;
- в quote дай короткую цитату-основание;
- replacements не возвращай.