

Пользовательские поля в /task: как настроить свои атрибуты задач

26.10.2026 · PDF-версия статьи с сайта cms.dolinais.ru

Стандартные атрибуты задачи — исполнитель, срок, приоритет, статус — покрывают базовый сценарий, но почти в каждой команде появляется что-то своё: код клиента, номер договора, тип услуги, бюджет, стенд или ответственный за приёмку. Заводить это в описание или комментарии — значит терять структуру и возможность фильтрации. Пользовательские поля в /task решают задачу иначе: вы сами определяете набор атрибутов, а система хранит их как полноценные данные задачи. Ниже — что это такое, зачем нужно, как настроить и какие выгоды это даёт в 2026 году.

Живое демо /task

Ниже — интерактивная демонстрация /task: можно переключать виды, открывать задачи и пробовать действия. Это лучший способ увидеть, как пользовательские поля выглядят в карточке и списке.

ДЕМО /TASK — КАНБАН: ПРИМЕР ДАННЫХ

Задачи в блоке не заданы.

Что такое пользовательские поля и зачем они нужны

Пользовательское поле — это настраиваемый атрибут задачи, который команда добавляет сама. В отличие от встроенных полей, оно не является частью ядра системы: вы создаёте его под свой процесс. Технически это метаданные задачи с типом (текст, число, дата, список, флаг, ссылка) и правилами отображения.

Зачем это нужно на практике:

- **Единый формат данных.** Вместо «клиент: Ромашка» в одном месте и «ООО Ромашка» в другом — выпадающий список с фиксированными значениями.
- **Фильтры и группировки.** По пользовательскому полю можно отбирать задачи в списке и на канбан-доске, строить отчёты.
- **Автоматизация.** Поле становится условием для правил: например, задачи с типом «Баг» и критичностью «Высокая» автоматически попадают в отдельную колонку.
- **Прозрачность для смежников.** Менеджеру не нужно читать переписку — достаточно открыть карточку и увидеть нужные атрибуты.

Важно не превращать это в свалку: каждое поле должно отвечать на вопрос «какое решение я приму на основе этого значения?». Если ответа нет — поле лишнее.

Отдельная страница настройки: где и как конфигурировать

Создание и редактирование атрибутов вынесено на отдельную страницу настроек, чтобы не смешивать повседневную работу с администрированием. Типовой сценарий выглядит так:

1. Откройте раздел настроек /task и перейдите на вкладку пользовательских полей.
2. Нажмите «Добавить поле» и задайте имя — короткое и понятное команде, без внутреннего жаргона.
3. Выберите тип: текст, число, дата, список значений, флаг или ссылка.
4. Для списков задайте варианты и, если нужно, значение по умолчанию.
5. Укажите область действия: все проекты или конкретный проект.
6. Настройте обязательность и видимость — например, показывать поле только после перехода задачи в определённый статус.
7. Сохраните и проверьте результат на тестовой задаче.

Отдельная страница удобна тем, что изменения не затрагивают рабочие карточки до сохранения, а список полей остаётся обзорным. Если поле перестало использоваться, его можно скрыть, не удаляя исторические значения.

Типы полей и когда какой выбирать

- **Текст** — для свободных комментариев, номеров договоров, внешних ссылок на макеты.
- **Число** — бюджет, оценка в часах, количество правок.
- **Дата** — дата согласования с клиентом, дата релиза, дедлайн приёмки.
- **Список значений** — тип услуги, клиент, стенд (dev/stage/prod), критичность.
- **Флаг** — бинарные признаки: «требуется ревью», «нужна публикация в блог».
- **Ссылка** — URL на тикет в другой системе, документацию или дизайн-макет.

Правка значений прямо в задаче

Ключевое преимущество — не нужно уходить в настройки, чтобы изменить значение. Пользовательские поля отображаются в карточке задачи и доступны для редактирования в том же окне. На практике это означает:

- значения можно менять в один клик — как и стандартные атрибуты;
- изменения сразу видны в списке и на доске, если поле выведено в представление;
- история изменений фиксируется, поэтому можно понять, кто и когда поменял значение.

Хорошая практика — выводить в карточку только 3–5 самых важных полей, а остальные прятать в блок «Дополнительно». Иначе карточка превращается в длинную анкету, и её перестают заполнять.

Правило: если поле не влияет на решение или отчёт, оно не должно занимать место в основной части карточки.

Пример из практики

Команда веб-студии добавила три поля: «Клиент» (список), «Тип работ» (список: вёрстка, интеграция, контент) и «Стенд» (список: dev, stage, prod). В результате менеджер фильтрует задачи по клиенту, а разработчик видит, на каком стенде воспроизводить баг, не задавая вопросов в чате. Ни одно из этих значений не пришлось менять в описании задачи.

Контекстные подсказки по полям

Контекстная подсказка — это краткое пояснение рядом с полем: что вводить, в каком формате, почему это

важно. Подсказки снижают количество ошибок и вопросов к администратору. Их стоит добавлять для полей, где возможна неоднозначность:

- для поля «Бюджет» — «в рублях, без НДС»;
- для поля «Срок согласования» — «дата, когда клиент подтвердил макет»;
- для поля «Стенд» — «выберите окружение, где воспроизводится проблема».

Подсказки отображаются при наведении или рядом с полем и не мешают опытным пользователям. Это особенно полезно в командах, где состав участников меняется: новый сотрудник быстрее разберётся, что от него требуется.

В чём плюсы такого подхода

- **Гибкость без доработок.** Атрибуты меняются через интерфейс, а не через код.
- **Порядок в данных.** Единые списки значений исключают разночтения.
- **Скорость работы.** Правка в карточке экономит переходы между экранами.
- **Меньше коммуникаций.** Подсказки и структурированные поля снимают типовые вопросы.
- **Основа для отчётности.** По пользовательским полям можно строить выборки и оценивать нагрузку по клиентам или типам работ.

Пользовательские поля в /task — это не «ещё одна настройка», а способ привести процесс к единому стандарту. Начните с двух-трёх полей, которые действительно влияют на решения, добавьте короткие подсказки и постепенно расширяйте набор по мере необходимости.

Источник: <https://cms.dolinais.ru/polzovatelskie-polya-v-task-kak-nastroit-svoi-atributy-zadach>