Зависимости задач в /task: как настроить блокировки и не сорвать сроки

17.11.2026 · PDF-версия статьи с сайта cms.dolinais.ru

В /task зависимости задач — это не просто «пометки для красоты», а рабочий инструмент управления порядком работ. Если вы уже знаете, что такое блокирующие задачи, списки «Ждёт завершения» и «Блокирует», и зачем нужен замок на карточке, самое время перейти к практике: как настроить связи так, чтобы они действительно помогали, а не мешали. В этой статье — пошаговая настройка, типичные ошибки и приёмы, которые сэкономят время команде в 2026 году.

Живое демо /task

Ниже — интерактивная демонстрация /task: можно переключать виды, открывать задачи и пробовать действия с зависимостями прямо в браузере.

ДЕМО /TASK — КАНБАН: ПРИМЕР ДАННЫХ

Задачи в блоке не заданы.

Что именно настраиваем

Напомним термины коротко, чтобы дальше говорить предметно.

- **Блокирующая задача** — задача, без завершения которой нельзя считать готовой другую. В /task такая связь называется зависимостью.
- **Список «Ждёт завершения»** — блок на карточке, где перечислены задачи-блокираторы, которые ещё не закрыты.
- **Список «Блокирует»** — обратный список: какие задачи ждут завершения текущей.
- **Замок на карточке** — индикатор и одновременно ограничение: пока блокираторы открыты, задачу нельзя перевести в статус «Готово».

Практическая задача — сделать так, чтобы эти элементы отображали реальный порядок работ, а не формальные связи «для отчёта».

Шаг 1. Определите, что действительно блокирует

Перед настройкой ответьте на вопрос: «Можно ли начать или закончить эту работу, если блокиратор не закрыт?» Если ответ «да, но неудобно» — это не блокировка, а пожелание. Заведите отдельную задачу или комментарий. Настоящая зависимость — когда результат блокиратора является входом для следующей задачи: макет до вёрстки, доступ к API до интеграции, согласование текста до публикации.

> Правило: одна задача — один-два блокиратора. Если их больше трёх, скорее всего, вы описываете не зависимость, а этап проекта.

Шаг 2. Создайте связь в карточке

1. Откройте задачу, которую нужно закрыть позже.
2. В блоке зависимостей добавьте блокиратор: выберите задачу из проекта или найдите по названию.
3. Проверьте, что связь отобразилась в двух местах: у текущей задачи — в «Ждёт завершения», у блокиратора — в «Блокирует».
4. Если задача уже была в работе, убедитесь, что замок появился: до закрытия блокиратора статус «Готово» будет недоступен.

Обратная связь важна: именно список «Блокирует» показывает исполнителю, кого он держит. Это снижает риск ситуации «сделал и забыл».

Шаг 3. Настройте порядок в списках

Списки «Ждёт завершения» и «Блокирует» — не свалка, а рабочий навигатор. Договоритесь в команде о простых правилах:

- в «Ждёт завершения» держите только актуальные блокираторы, снятые связи удаляйте;
- в «Блокирует» проверяйте задачи перед сменой приоритета: возможно, вы блокируете чужой срок;
- если блокиратор сам кого-то ждёт, стройте цепочку последовательно, а не вешайте всё на одну задачу.

На практике удобно раз в день просматривать задачи с замком: это быстрый способ увидеть, где проект «встал».

Шаг 4. Проверьте запрет завершения

Замок — это не наказание, а защита от преждевременного закрытия. Проверьте сценарий: попробуйте перевести задачу в «Готово», пока блокиратор открыт. Система не даст это сделать и подскажет, какие задачи нужно закрыть. Если запрет не срабатывает, проверьте, что связь создана именно как блокирующая, а не как информационная.

Отдельно обсудите исключения: иногда блокиратор закрывают позже по объективным причинам. В таких случаях не снимайте зависимость молча — зафиксируйте причину в комментарии и при необходимости перенесите срок. Иначе замок теряет смысл, а команда перестаёт ему доверять.

Типичные ошибки настройки

- **Циклические зависимости.** Задача А ждёт В, В ждёт А. Замок в таком случае блокирует обе задачи навсегда. Проверяйте цепочки перед сохранением.
- **Слишком длинные цепочки.** Пять-шесть последовательных блокировок трудно контролировать. Дробите этапы.
- **Зависимости «на будущее».** Связь, которая понадобится через месяц, только создаёт шум. Добавляйте её, когда работа приблизилась.
- **Игнорирование списка «Блокирует».** Исполнитель не видит, кого держит, и затягивает закрытие.

Что это даёт команде

Настроенные зависимости превращают план в исполнимый порядок работ. Вы быстрее видите узкие места:

задача с замком — сигнал, что где-то застрял блокиратор. Списки «Ждёт завершения» и «Блокирует» дают двусторонний обзор, а запрет завершения не позволяет формально закрыть задачу, за которой ещё нет результата. В итоге меньше переделок, честнее статусы и понятнее сроки — особенно когда над проектом работают несколько человек или ИИ-агенты.

Начните с одной цепочки на ближайшую неделю: создайте блокирующую связь, проверьте замок и списки. Если инструмент помогает, расширяйте практику на весь проект.

Источник: https://cms.dolinais.ru/zavisimosti-zadach-v-task-kak-nastroit-blokirovki-i-ne-sorvat-sroki