

Зависимости задач в /task: блокировки, списки «Ждёт завершения» и «Блокирует», замок на карточке

24.10.2026 · PDF-версия статьи с сайта cms.dolinais.ru

В командной работе редко бывает так, что все задачи независимы. Часто одна задача не может быть выполнена, пока не завершена другая: дизайн до вёрстки, бэкенд до интеграции, тесты после сборки. В таск-менеджере **/task** для управления такими связями есть механизм зависимостей. В этой статье разберём, что такое блокирующие задачи, как работают списки «Ждёт завершения» и «Блокирует», зачем нужен замок на карточке и почему система запрещает завершить задачу, пока не закрыты блокираторы. Материал актуален для версии **/task** 2026 года.

Живое демо /task

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

ДЕМО /TASK — КАНБАН: ПРИМЕР ДАННЫХ

Задачи в блоке не заданы.

Что такое зависимости задач и зачем они нужны

Зависимость задач — это связь между двумя задачами, при которой одна из них не может быть начата или завершена, пока не выполнена другая. В **/task** такая связь настраивается вручную и отображается на карточке задачи в виде двух списков: «Ждёт завершения» (задачи, которые блокируют текущую) и «Блокирует» (задачи, которые текущая блокирует).

Зачем это нужно? Без явных зависимостей команда часто берётся за задачу, которая ещё не готова к работе, тратит время на переделки или ждёт, не понимая, чего именно. Зависимости формализуют порядок работ и делают узкие места видимыми. Например, если задача «Свёрстать страницу» ждёт завершения задачи «Утвердить макет», то исполнитель вёрстки сразу видит, что начинать нельзя, а менеджер — что задержка на этапе макета остановит весь фронт работ.

Как настроить зависимость в /task

Настройка занимает несколько секунд. Откройте карточку задачи, найдите блок «Зависимости» и добавьте нужную связь.

1. Откройте задачу, которую нужно заблокировать (например, «Интеграция с платёжным шлюзом»).
2. В блоке «Зависимости» нажмите «Добавить блокирующую задачу».
3. Выберите из списка или найдите по номеру задачу-блокиратор (например, «Получить API-ключи от заказчика»).
4. Сохраните. Задача-блокиратор появится в списке «Ждёт завершения», а в её карточке текущая задача отобразится в списке «Блокирует».

Связь двусторонняя: добавив зависимость с одной стороны, вы автоматически увидите её с другой. Это удобно для навигации: открыв блокирующую задачу, можно сразу оценить, что она держит.

Списки «Ждёт завершения» и «Блокирует»

Эти два списка — основа работы с зависимостями. **«Ждёт завершения»** показывает задачи, которые должны быть закрыты до текущей. **«Блокирует»** — задачи, которые ждут завершения текущей. Оба списка кликабельны: можно перейти к связанной задаче, проверить её статус, исполнителя и срок.

На практике удобно перед началом работы открыть свою задачу и посмотреть список «Ждёт завершения». Если там есть незакрытые пункты, браться за работу рано — сначала нужно разобраться с блокираторами. А если вы завершаете задачу и видите непустой список «Блокирует», стоит предупредить коллег, что их очередь подошла.

Замок на карточке: что он означает

Когда у задачи есть хотя бы один незавершённый блокиратор, на её карточке появляется **замок** — визуальный индикатор того, что задача заблокирована. Замок виден в списке задач и на канбан-доске, поэтому не нужно открывать каждую карточку, чтобы понять, что работа стоит.

Замок не мешает редактировать описание, менять исполнителя или добавлять комментарии — он сигнализирует именно о невозможности завершения. Это важно: задача может быть в работе, но её финал зависит от другой задачи.

Запрет завершения до закрытия блокираторов

Самый важный механизм — **запрет перевода задачи в статус «Завершена»**, пока не закрыты все блокираторы. Если попытаться завершить задачу с активным замком, /task покажет предупреждение и не даст сохранить статус. Это защищает от случайного закрытия задачи, которая фактически не готова.

Пример: разработчик завершил вёрстку страницы, но задача «Утвердить тексты» ещё в работе. Система не позволит закрыть вёрстку, пока тексты не утверждены, — и это правильно, потому что без финальных текстов страница не готова.

Запрет действует и при групповых операциях: если выбрать несколько задач и попытаться завершить их разом, система проверит зависимости у каждой и пропустит только те, у которых нет активных блокираторов.

Практические сценарии использования

- **Разработка сайта.** Задача «Вёрстка» ждёт «Дизайн-макет», «Дизайн-макет» ждёт «Согласование с клиентом». Цепочка видна целиком, и задержка на любом этапе сразу отражается на следующих.
- **Мобильное приложение.** «Релиз в стор» блокируется задачами «Тестирование на iOS» и «Тестирование на Android». Пока оба тестирования не закрыты, релиз не завершить.
- **IT-консалтинг.** «Аудит инфраструктуры» блокирует «Подготовку рекомендаций». Пока аудит не завершён, рекомендации не могут быть финализированы.

В каждом случае зависимости делают процесс прозрачным: видно, что именно тормозит работу и кто должен действовать следующим.

Плюсы механизма зависимостей

- **Порядок вместо хаоса.** Команда не берётся за задачи, которые ещё не готовы, и не тратит время на переделки.
- **Видимость узких мест.** Блокирующая задача с большим списком «Блокирует» — сигнал, что её нужно ускорить.
- **Защита от преждевременного закрытия.** Замок и запрет завершения не дают случайно закрыть незаконченную работу.
- **Прозрачная коммуникация.** Списки «Ждёт завершения» и «Блокирует» дают ответ на вопрос «почему задача стоит» без лишних обсуждений.
- **Работает на всех видах.** Зависимости отображаются в списке задач, на канбан-доске и в карточке — в любом представлении /task.

Частые ошибки и как их избежать

Одна из типичных ошибок — строить слишком длинные цепочки зависимостей, где каждая задача ждёт предыдущую. Это превращает работу в жёсткую последовательность и лишает гибкости. Ставьте зависимости только там, где они действительно критичны: когда результат одной задачи физически или логически необходим для другой.

Вторая ошибка — забывать снимать зависимость, если она потеряла актуальность. Например, блокиратор был закрыт в другом проекте или отменён. Регулярно просматривайте список «Ждёт завершения» и удаляйте неактуальные связи, чтобы замок не висел зря.

Третья ошибка — игнорировать замок при планировании сроков. Если задача заблокирована, её срок сдвигается вместе с блокиратором. Учитывайте это при расстановке дат в полях «Начало выполнения» и «Срок».

Коротко о главном

Зависимости задач в /task — это простой и надёжный способ выстроить порядок работ. Блокирующие задачи видны в списках «Ждёт завершения» и «Блокирует», замок на карточке сигнализирует о блокировке, а запрет завершения гарантирует, что задача не будет закрыта раньше времени. Настройка занимает секунды, а польза для команды — ощутимая: меньше переделок, больше прозрачности и предсказуемые сроки.

Источник: <https://cms.dolnais.ru/zavisimosti-zadach-v-task-blokirovki-spiski-zhdet-zaversheniya-i-blokiruet-zamok-na-kartoc>