Зависимости задач в /task: блокирующие связи, списки «Ждёт завершения» и «Блокирует»

11.11.2026 · PDF-версия статьи с сайта cms.dolinais.ru

Что такое зависимости задач и зачем они нужны

Зависимость задач — это связь между двумя задачами, при которой одна не может быть завершена, пока не выполнена другая. В /task такая связь оформляется как **блокировка**: задача А блокирует задачу В, а В находится в ожидании завершения А. Это не просто пометка в описании, а рабочее ограничение, которое влияет на статусы, карточку и списки.

Представьте типичный сценарий: бэкенд-разработчик должен дождаться, пока дизайнер закроет макет, а тестировщик — пока разработчик не завершит реализацию. Без формальной зависимости такие задачи легко закрыть «на глазок», а потом выясняется, что релиз собрать нельзя. Блокировки решают три проблемы: показывают, кто кого ждёт, не дают завершить задачу раньше времени и делают план реалистичным.

Зависимости работают поверх обычных статусов. У задачи по-прежнему есть «В работе», «На проверке», «Готово», но появляется дополнительное условие: если есть открытый блокиратор, завершение запрещено. Это особенно полезно в командах, где задачи переходят между ролями и легко потерять, что именно тормозит процесс.

Живое демо /task

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

ДЕМО /TASK — КАНБАН: ПРИМЕР ДАННЫХ

Задачи в блоке не заданы.

Как устроены блокирующие связи

В /task у каждой задачи есть два встречных списка, которые показывают зависимости с разных сторон:

- **«Ждёт завершения» —** задачи, которые блокируют текущую. Пока хотя бы одна из них не закрыта, текущую нельзя завершить. Это список «я жду».
- **«Блокирует» —** задачи, которые зависят от текущей. Это список «меня ждут». Если закрыть текущую задачу, у зависимых снимется ограничение.

Оба списка видны на карточке задачи. Они не дублируют друг друга: если задача А блокирует В, то в карточке А в списке «Блокирует» появится В, а в карточке В в списке «Ждёт завершения» — А. Это зеркальные представления одной и той же связи.

Важно: блокировка — это не подзадача и не чек-лист. Подзадача входит в объём работ, а блокировка лишь фиксирует порядок. Одна и та же задача может одновременно кого-то ждать и кого-то блокировать, образуя

цепочку. /task не запрещает длинные цепочки, но циклические зависимости не создаст — система не даст замкнуть круг, потому что такой план невозможно выполнить.

Замок на карточке: что он означает

Если задача ждёт завершения хотя бы одного блокиратора, на её карточке появляется **замок**. Это визуальный признак того, что задача заблокирована. Замок не мешает открыть карточку, читать описание, писать комментарии или прикреплять файлы — он ограничивает именно завершение.

Замок полезен в ежедневной работе: на канбан-доске и в списках сразу видно, какие задачи нельзя двигать дальше, а какие готовы к работе. Не нужно открывать каждую карточку и вручную сверять статусы блокираторов — достаточно взглянуть на значок. Когда последний блокиратор закрывается, замок исчезает автоматически, и задача снова доступна для завершения.

> Замок — это не наказание, а навигация. Он отвечает на вопрос «почему эта задача стоит» быстрее, чем чтение комментариев.

Запрет завершения до закрытия блокираторов

Главное правило: задачу нельзя перевести в статус завершения, пока открыт хотя бы один блокиратор из списка «Ждёт завершения». При попытке закрыть такую задачу /task покажет, какие именно задачи мешают, и предложит перейти к ним. Это защищает от случайного закрытия и от ситуации «закрыли всё, а релиз не собрался».

Запрет действует и при массовых операциях: если выбрать несколько задач и попытаться завершить их разом, система не пропустит те, у которых есть активные блокираторы. Такой же принцип работает при автоматическом закрытии по правилам — зависимость проверяется до смены статуса.

Что делать, если блокиратор больше не актуален? Удалите связь вручную в карточке задачи. Это осознанное действие: снять блокировку можно только явно, случайно она не исчезнет. Если блокиратор был создан по ошибке, удаление связи сразу разблокирует задачу.

Как работать с зависимостями: пошаговый сценарий

1. Откройте задачу, которую нужно поставить в ожидание. Например, «Собрать релиз».
2. В блоке зависимостей добавьте блокиратор: выберите задачу «Закрыть макет» и укажите тип связи «Ждёт завершения».
3. Проверьте карточку блокиратора: в списке «Блокирует» должна появиться «Собрать релиз». Связь создана в обе стороны.
4. Убедитесь, что на карточке «Собрать релиз» появился замок. Попробуйте завершить задачу — /task не даст это сделать и покажет причину.
5. Закройте блокиратор «Закрыть макет». Замок с «Собрать релиз» снимется автоматически, и задача станет доступной для завершения.

Если вы работаете на канбан-доске, включите отображение зависимостей в настройках вида — тогда замки будут видны прямо на карточках в колонках. В списках задач блокировки отображаются отдельным значком, а

фильтры позволяют быстро найти все заблокированные задачи или все задачи, которые кого-то блокируют.

Типичные ошибки и как их избежать

- **Дублирование зависимостей в описании.** Если связь уже оформлена как блокировка, не нужно писать «ждём макет» в тексте — карточка покажет это сама.
- **Слишком длинные цепочки без необходимости.** Цепочка из десяти задач выглядит солидно, но усложняет планирование. Оставляйте только реальные зависимости, без которых задача действительно не может быть выполнена.
- **Попытка обойти замок сменой статуса.** Запрет завершения действует на уровне системы, поэтому обходных путей нет — и это хорошо: правила одинаковы для всех.
- **Забытые блокировки.** Если блокиратор закрыт, но связь осталась, задача может остаться заблокированной. Проверяйте список «Ждёт завершения» при планировании спринта.

В чём плюсы для команды

Зависимости задач в /task дают прозрачность: каждый видит, кто кого ждёт, и не тратит время на выяснение причин простоя. Замок на карточке и список «Ждёт завершения» работают как раннее предупреждение — проблема видна до того, как задача попадёт в релиз. Запрет завершения снижает риск ложных «Готово» и делает статусы достоверными.

Для руководителя это ещё и инструмент планирования: по списку «Блокирует» можно оценить, какие задачи находятся на критическом пути, а по количеству замков — где скапливается работа. В итоге зависимости не усложняют процесс, а убирают ручные договорённости и делают порядок выполнения задач явным.

Источник: https://cms.dolinais.ru/zavisimosti-zadach-v-task-blokiruyuschie-svyazi-spiski-zhdet-zaversheniya-i-blokiruet