Зависимости задач в /task: блокирующие задачи и запрет завершения

24.11.2026 · PDF-версия статьи с сайта cms.dolinais.ru

Представьте: разработчик закрывает задачу «Верстка страницы оплаты», хотя API платежного шлюза ещё не готов. Верстка уходит в тестирование, тестировщик тратит время, а через день выясняется, что интеграция не работает. Приходится переделывать. Такие ситуации возникают из-за отсутствия явных зависимостей между задачами. В /task эта проблема решается с помощью блокирующих задач, списков «Ждёт завершения» и «Блокирует», а также замка на карточке, который физически не даёт завершить задачу раньше блокираторов.

В этой статье разберём, что это за механизмы, зачем они нужны, как с ними работать и какие плюсы вы получаете. Опираемся на актуальную версию /task (2026 год).

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 блокирующие задачи и списки «Ждёт завершения» / «Блокирует»

Блокирующая задача — это задача, которая должна быть завершена до того, как вы сможете закрыть текущую. В /task такие связи отображаются в двух списках на карточке задачи:

- **«Ждёт завершения» —** задачи, которые блокируют текущую. Пока они не закрыты, текущую завершить нельзя.
- **«Блокирует» —** задачи, которые зависят от текущей. Пока вы не закроете текущую, эти задачи не могут быть завершены.

Эти списки двусторонние: если задача А блокирует задачу В, то в карточке А в списке «Блокирует» появится В, а в карточке В в списке «Ждёт завершения» — А. Связи устанавливаются вручную или автоматически при использовании шаблонов.

Замок на карточке — это визуальный индикатор и техническое ограничение, которое не позволяет завершить задачу, пока все блокираторы из списка «Ждёт завершения» не будут закрыты. Замок отображается на карточке в виде иконки и подсказки, какие именно задачи мешают завершению.

Зачем это нужно: три типичные проблемы

Без управления зависимостями команды сталкиваются с:

1. **Нарушением порядка работ.** Задачи выполняются не в той последовательности, что приводит к переделкам и простоям.
2. **Ложным ощущением готовности.** Задача помечена как выполненная, хотя её результаты нельзя использовать.
3. **Потерей контекста.** Участники не видят, что именно блокирует их работу, и тратят время на выяснение.

Блокировки решают эти проблемы: они делают зависимости явными и не позволяют обойти установленный порядок.

Как работать с зависимостями в /task

Рассмотрим пошаговый сценарий.

Шаг 1. Создайте задачи и установите связи

Откройте карточку задачи, которая должна ждать другую. В блоке «Зависимости» нажмите «Добавить блокиратор» и выберите нужную задачу из списка. Она появится в списке «Ждёт завершения». В карточке выбранной задачи автоматически появится обратная связь в списке «Блокирует».

Шаг 2. Проверьте замок

Как только у задачи появляется хотя бы один незакрытый блокиратор, на карточке загорается замок. При попытке завершить задачу система выдаст предупреждение и не даст это сделать, пока блокираторы не будут закрыты.

Шаг 3. Закрывайте блокираторы

Перейдите в задачу из списка «Ждёт завершения», выполните её и завершите. После этого замок с текущей задачи снимается автоматически, если больше нет других незакрытых блокираторов.

Шаг 4. Используйте списки для планирования

Список «Блокирует» помогает понять, какие задачи пострадают, если вы задержите текущую. Это полезно для приоритизации: если задача блокирует много других, её стоит делать в первую очередь.

> Совет: устанавливайте зависимости на этапе планирования спринта, чтобы сразу видеть критический путь проекта.

Запрет завершения до закрытия блокираторов: как это работает технически

В /task запрет реализован на уровне бизнес-логики: при вызове метода завершения задачи сервер проверяет все связи в списке «Ждёт завершения». Если хотя бы одна задача не имеет статуса «Завершена», возвращается ошибка с перечнем блокираторов. Обойти это ограничение через API или интерфейс нельзя — только закрыть блокираторы или удалить связь (если у вас есть права).

Важно: запрет действует и для массовых операций. Если вы попытаетесь закрыть несколько задач сразу, а у одной из них есть активный блокиратор, операция не выполнится для этой задачи, остальные закроются.

Плюсы использования зависимостей

- **Порядок в работе.** Задачи выполняются в логической последовательности, меньше переделок.
- **Прозрачность.** Все видят, что и кого блокирует, не нужно держать это в голове.
- **Защита сроков.** Замок не даёт закрыть задачу раньше времени и создать иллюзию готовности.
- **Автоматизация контроля.** Не нужно вручную проверять, все ли предпосылки выполнены, — система сделает это за вас.
- **Улучшение коммуникации.** Участники быстрее понимают, почему задача стоит на месте, и могут помочь с блокиратором.

Внедрение зависимостей требует дисциплины: связи нужно поддерживать в актуальном состоянии. Но затраты окупаются снижением количества ошибок и простоев. Начните с малого — добавьте блокировки для нескольких критичных задач, и вы увидите эффект.

Источник: https://cms.dolinais.ru/zavisimosti-zadach-v-task-blokiruyuschie-zadachi-i-zapret-zaversheniya