Зависимости задач в /task: как блокировки защищают сроки

23.11.2026 · PDF-версия статьи с сайта cms.dolinais.ru

В управлении проектами часто возникает ситуация, когда одна задача не может быть выполнена до завершения другой. Например, тестирование новой функции нельзя начать, пока разработчик не закончил код, а релиз не состоится без прохождения тестов. Если такие связи не учитывать, проект рискует сорвать сроки, а команда — потратить время впустую. В системе **/task** для решения этой проблемы предусмотрены **зависимости задач** — механизм, который автоматически отслеживает блокирующие связи и не даёт закрыть задачу раньше времени. В этой статье мы подробно разберём, что такое блокирующие задачи, как работают списки «Ждёт завершения» и «Блокирует», зачем нужен замок на карточке и какие преимущества даёт такой подход.

Живое демо /task

Ниже — интерактивная демонстрация возможностей /task. Вы можете переключать виды, открывать задачи и пробовать действия с зависимостями.

ДЕМО /TASK — КАНБАН: ПРИМЕР ДАННЫХ

Задачи в блоке не заданы.

Что такое блокирующие задачи и зависимости

Блокирующая задача (или блокиратор) — это задача, которая должна быть завершена до того, как можно будет приступить к другой задаче или закрыть её. В /task такие связи называются **зависимостями**. Они бывают двух типов:

- **Ждёт завершения** — текущая задача зависит от другой, то есть не может быть завершена, пока блокиратор не закрыт.
- **Блокирует** — текущая задача является блокиратором для другой, то есть мешает её завершению.

Эти два списка отображаются на карточке задачи и позволяют быстро оценить, какие связи установлены. Например, если задача «Тестирование» ждёт завершения задачи «Разработка», то на карточке «Тестирование» в списке «Ждёт завершения» будет указана «Разработка», а на карточке «Разработка» в списке «Блокирует» — «Тестирование».

Замок на карточке: запрет завершения до закрытия блокираторов

Одна из ключевых особенностей /task — **замок на карточке**. Он не позволяет завершить задачу, если хотя бы один из её блокираторов ещё не закрыт. Визуально замок отображается на карточке и служит напоминанием о том, что путь к завершению заблокирован. При попытке закрыть такую задачу система выдаст предупреждение и предложит сначала завершить зависимые задачи.

Замок работает автоматически: как только все блокираторы закрыты, замок снимается, и задача становится

доступной для завершения. Это исключает человеческий фактор — например, случайное закрытие задачи, которая ещё не готова, или игнорирование зависимостей.

Зачем нужны зависимости и замок: практические сценарии

Рассмотрим несколько типичных ситуаций, где блокировки критически важны.

Разработка и тестирование

Разработчик создаёт задачу «Написать модуль авторизации». Тестировщик создаёт задачу «Протестировать авторизацию» и указывает, что она ждёт завершения первой. Пока модуль не готов, тестировщик не может закрыть свою задачу — замок не позволит. Это гарантирует, что тестирование не будет пропущено и не начнётся раньше времени.

Согласование документации

Задача «Обновить руководство пользователя» зависит от задачи «Утвердить новые скриншоты». Пока скриншоты не утверждены, технический писатель не может завершить обновление. Замок предотвращает преждевременную публикацию устаревших материалов.

Релиз продукта

Задача «Выпустить релиз» блокируется задачами «Завершить все тесты» и «Обновить конфигурацию сервера». Без их закрытия релиз невозможен. Замок гарантирует, что ни один этап не будет пропущен.

Во всех этих случаях зависимости помогают избежать ошибок, переделок и срыва сроков.

Как работать с зависимостями в /task: пошаговая инструкция

Чтобы настроить блокировки, выполните следующие шаги:

1. **Откройте карточку задачи**, которую нужно заблокировать.
2. В разделе «Зависимости» нажмите «Добавить блокиратор».
3. Выберите из списка задач ту, которая должна быть выполнена первой (блокиратор). Она появится в списке «Ждёт завершения».
4. На карточке блокиратора автоматически появится запись в списке «Блокирует» с ссылкой на вашу задачу.
5. При попытке завершить заблокированную задачу система покажет замок и предупреждение.
6. После завершения всех блокираторов замок исчезнет, и задача сможет быть закрыта.

Обратите внимание: зависимости можно устанавливать только между задачами в рамках одного проекта или между задачами, доступными пользователю. Если задача находится в другом проекте, убедитесь, что у вас есть права на её просмотр.

Преимущества использования зависимостей и замка

- **Прозрачность**: все участники видят, какие задачи блокируют другие, и могут планировать работу.
- **Предотвращение ошибок**: замок исключает случайное или намеренное закрытие задачи до выполнения предварительных условий.
- **Соблюдение сроков**: зависимости помогают выявить узкие места и не допустить задержек на критическом пути.
- **Автоматизация контроля**: не нужно вручную проверять, готова ли задача — система сделает это за вас.
- **Улучшение коммуникации**: списки «Ждёт завершения» и «Блокирует» служат напоминанием о взаимных обязательствах команды.

Заключение

Зависимости задач — мощный инструмент, который помогает поддерживать порядок в проекте и избегать срывов сроков. Блокирующие задачи, списки «Ждёт завершения» и «Блокирует», а также замок на карточке создают надёжный барьер для преждевременного завершения работ. В /task этот механизм реализован интуитивно и автоматически, что делает его доступным для команд любого размера. Используйте зависимости, чтобы ваш проект двигался по плану, а задачи закрывались только тогда, когда для этого действительно есть все основания.

Источник: https://cms.dolinais.ru/zavisimosti-zadach-v-task-kak-blokirovki-zaschischayut-sroki-4