

dfhgakshdqk dgaehgkla

© 2015 by [unreadable] Inc.

18+

dfhgakshdgk dgaehgkla
es_owjfkdsllwejfndnv

«Автор»

2026

dgaehgkla d.

es_owjfkdslllwejfndnv / d. dgaehgkla — «Автор», 2026

ksdnckjdnksbdkjbdksknckdsneckdn sckjdnskjbdsckjc kjdsbckjhdsbcjkhdsbckjdsncks

© dgaehgkla d., 2026

© Автор, 2026

Содержание

Глава 1	5
Конец ознакомительного фрагмента.	17

dfhgakshdgk dgaehgkla es_owjfkdsllwejfjndnv

Глава 1

Глава 8. Чек-лист

Утро началось с сообщения от тимлида.

Без «доброе утро».

Без смайликов.

А значит — действительно что-то серьёзное.

Без права на сюрпризы.»«Катя, нужен полный регресс по новой логике скачивания.

Сегодня. Перед релизом.

Катя посмотрела на экран.

Потом на кофе.

Потом снова на экран.

— Без права на сюрпризы, — повторила она вслух. — Какая наивная вера в жизнь.

Она открыла ноутбук.

Новая логика скачивания книг.

Та самая задача, которая уже две недели методично разрушала нервную систему всей команды.

Любимый набор.Foundation API. Новые ссылки. Grouped files. File link. TTL. Проверка лицензий. Офлайн-доступ. Повторная загрузка. 403. 404. 422.

Это был не просто тест.

Это был экзамен на выживание.

Катя открыла свой чек-лист.

Не официальный.

Настоящий.

Тот, который не пишут в Confluence, потому что он выглядит как личная переписка с хаосом.

Основное

— проверить stream without full download— скачать текстовую книгу — скачать аудиокнигу — скачать fb2 — скачать epub — скачать mp3 — проверить webtoon — проверить trial — проверить full version — проверить по подписке — проверить купленную книгу — проверить библиотечную выдачу — проверить офлайн после скачивания — проверить повторное открытие — проверить после перелога — проверить после переустановки — проверить после истечения ссылки — проверить после смены сети — проверить медленный интернет — проверить плохой интернет — проверить отсутствие интернета — проверить повторную загрузку — проверить удаление файла — проверить повторное скачивание — проверить старую ссылку — проверить новую ссылку — проверить range request

Она смотрела на список как на личного врага.

что-то обязательно сломается.И знала:

Вопрос только — где.

Первый час прошёл слишком спокойно.

Это раздражало.

Текстовая книга скачивалась.

Аудио открывалось.

Файл не исчезал.

Офлайн работал.
Повторный вход не ломал доступ.
Даже слишком хорошо.
Катя не верила в такое счастье.
Она открыла Proxuman.
Если баг не приходит сам — надо идти за ним лично.
Она подменила ответ на /foundation/api/arts/files/{file_id}/link
Сделала принудительный 403.
Проверила реакцию клиента.
И вот тут началось интересное.
Приложение показало пользователю:
«Что-то пошло не так»
Катя уставилась в экран.
Моя жизнь?— Что именно?
Нет.
Так нельзя.
Если пользователь теряет доступ к книге, он должен понимать, что произошло.
403 — это не «что-то».
Это конкретная проблема доступа.
Она создала баг.
Title: Некорректный текст ошибки при 403 на получение ссылки файла
Понятное сообщение о проблеме доступа / отсутствии правExpected:
Абстрактное «Что-то пошло не так»Actual:
Severity: Medium
Потому что UX тоже часть качества.
Иногда даже важнее backend-идеальности.
Следующий сценарий.
Истечение ссылки.
Ссылка на скачивание имела TTL.
Менее красиво в жизни.Красиво на схеме.
Повторила запрос.Катя получила ссылку. Подождала.

Отлично.
Так и должно быть.
Но приложение...
...начинало бесконечную загрузку.
Без смысла.Без ошибки. Без выхода.
Просто вечный spinner.
Катя смотрела на экран почти философски.
Это экзистенциальный кризис.— Это не loading.
Ещё один баг.
Critical
Потому что бесконечная загрузка — это почти личное оскорбление.
или приложение просто решило уйти в монастырь.Пользователь не должен гадать: сло-
малось, думает,
После обеда прилетел Дима.
Физически.
Прямо к её столу.
С лицом человека, который пришёл либо за помощью, либо за прощением.

Обычно это одно и то же.

— У меня вопрос.

— Уже нет.

— Катя.

— Дима.

— Это быстро.

— Ложь.

Он сел рядом.

Если пользователь удалил скачанную книгу вручную из файловой системы, а приложение думает, что книга всё ещё скачана — это баг клиента или edge case?— Смотри.

Катя даже не задумалась.

— Баг.

— Почему?

— Потому что пользователь не обязан понимать внутреннюю архитектуру нашего страдания.

— Сильно.

— Правильно.

Она повернула к нему ноутбук.

Пользователь не виноват.— Если UI показывает “книга скачана”, а книги нет — это ложь интерфейса.

Дима кивнул.

— Я иногда думаю, что ты споришь с системой как с живым человеком.

— Потому что она ведёт себя как токсичный бывший.

— Справедливо.

Он ушёл.

Вероятно, переваривать.

К вечеру случилось то, чего Катя ждала весь день.

Regression.

Куда же без него.

После фикса бесконечного spinner’a перестал работать streaming без полной загрузки.

и приложение требовало скачать весь файл целиком.Теперь пользователь нажимал «слушать» —

Для длинных аудиокниг это было почти преступлением.

Streaming должен стартовать сразу.

Без ожидания.

Без полного скачивания.

Иначе смысл всей новой логики просто умирал.

Катя медленно откинулась на спинку стула.

Вот он.

Главный баг дня.

Release stopper.Красивый. Жирный.

Она открыла задачу.

Очень спокойно.

и осталась только профессиональная ясность.С этим особым состоянием QA, когда злость уже прошла,

Title: Streaming аудиокниги не стартует без полной загрузки после фикса expired link

Приложение требует полный download перед воспроизведениемActual:

Playback должен стартовать через stream без необходимости полного скачиванияExpected:

Severity: Blocker

Priority: Highest

Комментарий:

«Текущее поведение ломает основной пользовательский сценарий потребления аудио-контента»

мы сейчас всё отменяемПеревод с QA-языка:

Она отправила баг.

Через минуту сообщение от тимлида:

«Это точно blocker?»

Катя закрыла глаза.

Очень медленно.

Потом открыла.

И написала:

Это core functionality.»«Если пользователь не может начать слушать книгу сразу — да.

Это не nice-to-have.

Пауза.

Потом:

Релиз стопаем.»«Принято.

Она выдохнула.

Ещё один релиз, который не ушёл в прод с ножом в руках.Ещё один день.

Иногда работа тестировщика — это не искать баги.

Иногда это просто не дать катастрофе выглядеть как “почти нормально”.

Она закрыла ноутбук.

На сегодня хватит.

Кофе больше не помогал.Голова гудела. Глаза болели.

ради которого всё это вообще имеет смысл.Но внутри было то редкое чувство,

Ты увидела.

Ты не пропустила.

Ты защитила пользователя.

Не попадает в презентации.Это не громко. Не красиво.

Но именно на этом всё и держится.

Катя взяла телефон.

Новое сообщение от Макса:

«Как прошёл день?»

Она посмотрела на экран и улыбнулась.

Ответила честно:

«Сегодня я снова спасла людей от очень плохих решений.»

Пауза.

Потом пришёл ответ:

«То есть обычный вторник?»

Она усмехнулась.

Это был регресс.»«Хуже.

И почему-то именно это объясняло всё.

Глава 8. Чек-лист

Утро началось с сообщения от тимлида.

Без «доброе утро».

Без смайликов.

А значит — действительно что-то серьёзное.

Без права на сюрпризы.»«Катя, нужен полный регресс по новой логике скачивания. Сегодня. Перед релизом.

Катя посмотрела на экран.

Потом на кофе.

Потом снова на экран.

— Без права на сюрпризы, — повторила она вслух. — Какая наивная вера в жизнь.

Она открыла ноутбук.

Новая логика скачивания книг.

Та самая задача, которая уже две недели методично разрушала нервную систему всей команды.

Любимый набор.Foundation API. Новые ссылки. Grouped files. File link. TTL. Проверка лицензий. Офлайн-доступ. Повторная загрузка. 403. 404. 422.

Это был не просто тест.

Это был экзамен на выживание.

Катя открыла свой чек-лист.

Не официальный.

Настоящий.

Тот, который не пишут в Confluence, потому что он выглядит как личная переписка с хаосом.

Основное

— проверить stream without full download— скачать текстовую книгу — скачать аудиокнигу — скачать fb2 — скачать epub — скачать mp3 — проверить webtoon — проверить trial — проверить full version — проверить по подписке — проверить купленную книгу — проверить библиотечную выдачу — проверить офлайн после скачивания — проверить повторное открытие — проверить после перелога — проверить после переустановки — проверить после истечения ссылки — проверить после смены сети — проверить медленный интернет — проверить плохой интернет — проверить отсутствие интернета — проверить повторную загрузку — проверить удаление файла — проверить повторное скачивание — проверить старую ссылку — проверить новую ссылку — проверить range request

Она смотрела на список как на личного врага.

что-то обязательно сломается.И знала:

Вопрос только — где.

Первый час прошёл слишком спокойно.

Это раздражало.

Текстовая книга скачивалась.

Аудио открывалось.

Файл не исчезал.

Офлайн работал.

Повторный вход не ломал доступ.

Даже слишком хорошо.

Катя не верила в такое счастье.

Она открыла Proxuman.

Если баг не приходит сам — надо идти за ним лично.

Она подменила ответ на /foundation/api/arts/files/{file_id}/link

Сделала принудительный 403.

Проверила реакцию клиента.

И вот тут началось интересное.

Приложение показало пользователю:

«Что-то пошло не так»

Катя уставилась в экран.

Моя жизнь?— Что именно?

Нет.

Так нельзя.

Если пользователь теряет доступ к книге, он должен понимать, что произошло.

403 — это не «что-то».

Это конкретная проблема доступа.

Она создала баг.

Title: Некорректный текст ошибки при 403 на получение ссылки файла

Понятное сообщение о проблеме доступа / отсутствии правExpected:

Абстрактное «Что-то пошло не так»Actual:

Severity: Medium

Потому что UX тоже часть качества.

Иногда даже важнее backend-идеальности.

Следующий сценарий.

Истечение ссылки.

Ссылка на скачивание имела TTL.

Менее красиво в жизни.Красиво на схеме.

Повторила запрос.Катя получила ссылку. Подождала.

Отлично.

Так и должно быть.

Но приложение...

...начинало бесконечную загрузку.

Без смысла.Без ошибки. Без выхода.

Просто вечный spinner.

Катя смотрела на экран почти философски.

Это экзистенциальный кризис.— Это не loading.

Ещё один баг.

Critical

Потому что бесконечная загрузка — это почти личное оскорбление.

или приложение просто решило уйти в монастырь.Пользователь не должен гадать: сломалось, думает,

После обеда прилетел Дима.

Физически.

Прямо к её столу.

С лицом человека, который пришёл либо за помощью, либо за прощением.

Обычно это одно и то же.

— У меня вопрос.

— Уже нет.

— Катя.

— Дима.

— Это быстро.

— Ложь.

Он сел рядом.

Если пользователь удалил скачанную книгу вручную из файловой системы, а приложение думает, что книга всё ещё скачана — это баг клиента или edge case?— Смотри.

Катя даже не задумалась.

— Баг.

— Почему?

— Потому что пользователь не обязан понимать внутреннюю архитектуру нашего страдания.

— Сильно.

— Правильно.

Она повернула к нему ноутбук.

Пользователь не виноват.— Если UI показывает “книга скачана”, а книги нет — это ложь интерфейса.

Дима кивнул.

— Я иногда думаю, что ты споришь с системой как с живым человеком.

— Потому что она ведёт себя как токсичный бывший.

— Справедливо.

Он ушёл.

Вероятно, переваривать.

К вечеру случилось то, чего Катя ждала весь день.

Regression.

Куда же без него.

После фикса бесконечного spinner’а перестал работать streaming без полной загрузки. и приложение требовало скачать весь файл целиком. Теперь пользователь нажимал «слушать» —

Для длинных аудиокниг это было почти преступлением.

Streaming должен стартовать сразу.

Без ожидания.

Без полного скачивания.

Иначе смысл всей новой логики просто умирал.

Катя медленно откинулась на спинку стула.

Вот он.

Главный баг дня.

Release stopper. Красивый. Жирный.

Она открыла задачу.

Очень спокойно.

и осталась только профессиональная ясность. С этим особым состоянием QA, когда злость уже прошла,

Title: Streaming аудиокниги не стартует без полной загрузки после фикса expired link

Приложение требует полный download перед воспроизведением Actual:

Playback должен стартовать через stream без необходимости полного скачивания Expected:

Severity: Blocker

Priority: Highest

Комментарий:

«Текущее поведение ломает основной пользовательский сценарий потребления аудио-контента»

мы сейчас всё отменяем Перевод с QA-языка:

Она отправила баг.

Через минуту сообщение от тимлида:

«Это точно blocker?»

Катя закрыла глаза.

Очень медленно.

Потом открыла.

И написала:

Это core functionality.»«Если пользователь не может начать слушать книгу сразу — да. Это не nice-to-have.

Пауза.

Потом:

Релиз стопаем.»«Принято.

Она выдохнула.

Ещё один релиз, который не ушёл в прод с ножом в руках.Ещё один день.

Иногда работа тестировщика — это не искать баги.

Иногда это просто не дать катастрофе выглядеть как “почти нормально”.

Она закрыла ноутбук.

На сегодня хватит.

Кофе больше не помогал.Голова гудела. Глаза болели.

ради которого всё это вообще имеет смысл.Но внутри было то редкое чувство,

Ты увидела.

Ты не пропустила.

Ты защитила пользователя.

Не попадает в презентации.Это не громко. Не красиво.

Но именно на этом всё и держится.

Катя взяла телефон.

Новое сообщение от Макса:

«Как прошёл день?»

Она посмотрела на экран и улыбнулась.

Ответила честно:

«Сегодня я снова спасла людей от очень плохих решений.»

Пауза.

Потом пришёл ответ:

«То есть обычный вторник?»

Она усмехнулась.

Это был регресс.»«Хуже.

И почему-то именно это объясняло всё.

Глава 8. Чек-лист

Утро началось с сообщения от тимлида.

Без «доброе утро».

Без смайликов.

А значит — действительно что-то серьёзное.

Без права на сюрпризы.»«Катя, нужен полный регресс по новой логике скачивания.

Сегодня. Перед релизом.

Катя посмотрела на экран.

Потом на кофе.

Потом снова на экран.

— Без права на сюрпризы, — повторила она вслух. — Какая наивная вера в жизнь.

Она открыла ноутбук.

Новая логика скачивания книг.

Та самая задача, которая уже две недели методично разрушала нервную систему всей команды.

Любимый набор.Foundation API. Новые ссылки. Grouped files. File link. TTL. Проверка лицензий. Офлайн-доступ. Повторная загрузка. 403. 404. 422.

Это был не просто тест.

Это был экзамен на выживание.

Катя открыла свой чек-лист.

Не официальный.

Настоящий.

Тот, который не пишут в Confluence, потому что он выглядит как личная переписка с хаосом.

Основное

— проверить stream without full download — скачать текстовую книгу — скачать аудиокнигу — скачать fb2 — скачать epub — скачать mp3 — проверить webtoon — проверить trial — проверить full version — проверить по подписке — проверить купленную книгу — проверить библиотечную выдачу — проверить офлайн после скачивания — проверить повторное открытие — проверить после перелога — проверить после переустановки — проверить после истечения ссылки — проверить после смены сети — проверить медленный интернет — проверить плохой интернет — проверить отсутствие интернета — проверить повторную загрузку — проверить удаление файла — проверить повторное скачивание — проверить старую ссылку — проверить новую ссылку — проверить range request

Она смотрела на список как на личного врага.

что-то обязательно сломается. И знала:

Вопрос только — где.

Первый час прошёл слишком спокойно.

Это раздражало.

Текстовая книга скачивалась.

Аудио открывалось.

Файл не исчезал.

Офлайн работал.

Повторный вход не ломал доступ.

Даже слишком хорошо.

Катя не верила в такое счастье.

Она открыла Proxuman.

Если баг не приходит сам — надо идти за ним лично.

Она подменила ответ на /foundation/api/arts/files/{file_id}/link

Сделала принудительный 403.

Проверила реакцию клиента.

И вот тут началось интересное.

Приложение показало пользователю:

«Что-то пошло не так»

Катя уставилась в экран.

Моя жизнь? — Что именно?

Нет.

Так нельзя.

Если пользователь теряет доступ к книге, он должен понимать, что произошло.

403 — это не «что-то».

Это конкретная проблема доступа.

Она создала баг.

Title: Некорректный текст ошибки при 403 на получение ссылки файла

Понятное сообщение о проблеме доступа / отсутствии правExpected:

Абстрактное «Что-то пошло не так»Actual:

Severity: Medium

Потому что UX тоже часть качества.

Иногда даже важнее backend-идеальности.
Следующий сценарий.
Истечение ссылки.
Ссылка на скачивание имела TTL.
Менее красиво в жизни. Красиво на схеме.
Повторила запрос. Катя получила ссылку. Подождала.

Отлично.
Так и должно быть.
Но приложение...
...начинало бесконечную загрузку.
Без смысла. Без ошибки. Без выхода.
Просто вечный spinner.
Катя смотрела на экран почти философски.
Это экзистенциальный кризис. — Это не loading.
Ещё один баг.

Critical

Потому что бесконечная загрузка — это почти личное оскорбление.
или приложение просто решило уйти в монастырь. Пользователь не должен гадать: сломалось, думает,

После обеда прилетел Дима.

Физически.

Прямо к её столу.

С лицом человека, который пришёл либо за помощью, либо за прощением.

Обычно это одно и то же.

— У меня вопрос.

— Уже нет.

— Катя.

— Дима.

— Это быстро.

— Ложь.

Он сел рядом.

Если пользователь удалил скачанную книгу вручную из файловой системы, а приложение думает, что книга всё ещё скачана — это баг клиента или edge case? — Смотри.

Катя даже не задумалась.

— Баг.

— Почему?

— Потому что пользователь не обязан понимать внутреннюю архитектуру нашего страдания.

— Сильно.

— Правильно.

Она повернула к нему ноутбук.

Пользователь не виноват. — Если UI показывает “книга скачана”, а книги нет — это ложь интерфейса.

Дима кивнул.

— Я иногда думаю, что ты споришь с системой как с живым человеком.

— Потому что она ведёт себя как токсичный бывший.

— Справедливо.

Он ушёл.

Вероятно, переваривать.

К вечеру случилось то, чего Катя ждала весь день.

Regression.

Куда же без него.

После фикса бесконечного spinner'a перестал работать streaming без полной загрузки.

и приложение требовало скачать весь файл целиком. Теперь пользователь нажимал «слушать» —

Для длинных аудиокниг это было почти преступлением.

Streaming должен стартовать сразу.

Без ожидания.

Без полного скачивания.

Иначе смысл всей новой логики просто умирал.

Катя медленно откинулась на спинку стула.

Вот он.

Главный баг дня.

Release stopper. Красивый. Жирный.

Она открыла задачу.

Очень спокойно.

и осталась только профессиональная ясность. С этим особым состоянием QA, когда злость уже прошла,

Title: Streaming аудиокниги не стартует без полной загрузки после фикса expired link

Приложение требует полный download перед воспроизведением Actual:

Playback должен стартовать через stream без необходимости полного скачивания Expected:

Severity: Blocker

Priority: Highest

Комментарий:

«Текущее поведение ломает основной пользовательский сценарий потребления аудио-контента»

мы сейчас всё отменяем Перевод с QA-языка:

Она отправила баг.

Через минуту сообщение от тимлида:

«Это точно blocker?»

Катя закрыла глаза.

Очень медленно.

Потом открыла.

И написала:

Это core functionality.» «Если пользователь не может начать слушать книгу сразу — да.

Это не nice-to-have.

Пауза.

Потом:

Релиз стопаем.» «Принято.

Она выдохнула.

Ещё один релиз, который не ушёл в прод с ножом в руках. Ещё один день.

Иногда работа тестировщика — это не искать баги.

Иногда это просто не дать катастрофе выглядеть как “почти нормально”.

Она закрыла ноутбук.

На сегодня хватит.

Кофе больше не помогал. Голова гудела. Глаза болели.

ради которого всё это вообще имеет смысл. Но внутри было то редкое чувство,
Ты увидела.
Ты не пропустила.
Ты защитила пользователя.
Не попадает в презентации. Это не громко. Не красиво.
Но именно на этом всё и держится.
Катя взяла телефон.
Новое сообщение от Макса:
«Как прошёл день?»
Она посмотрела на экран и улыбнулась.
Ответила честно:
«Сегодня я снова спасла людей от очень плохих решений.»
Пауза.
Потом пришёл ответ:
«То есть обычный вторник?»
Она усмехнулась.
Это был регресс.» «Хуже.
И почему-то именно это объясняло всё.
Глава 8. Чек-лист
Утро началось с сообщения от тимлида.
Без «доброе утро».
Без смайликов.
А значит — действительно что-то серьёзное.
Без права на сюрпризы.» «Катя, нужен полный регресс по новой логике скачивания.
Сегодня. Перед релизом.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.