



# КОМАНДНАЯ РАБОТА



ИЛЬНАЗ ГИЛЬЯЗОВ



# ИЛЬНАЗ ГИЛЬЯЗОВ

Технический директор в aims



[coursar@gmail.com](mailto:coursar@gmail.com)

---

# ПЛАН ЗАНЯТИЯ

1. Удаленные ветки
2. Issues
3. Pull Requests
4. GitHub Pages

---

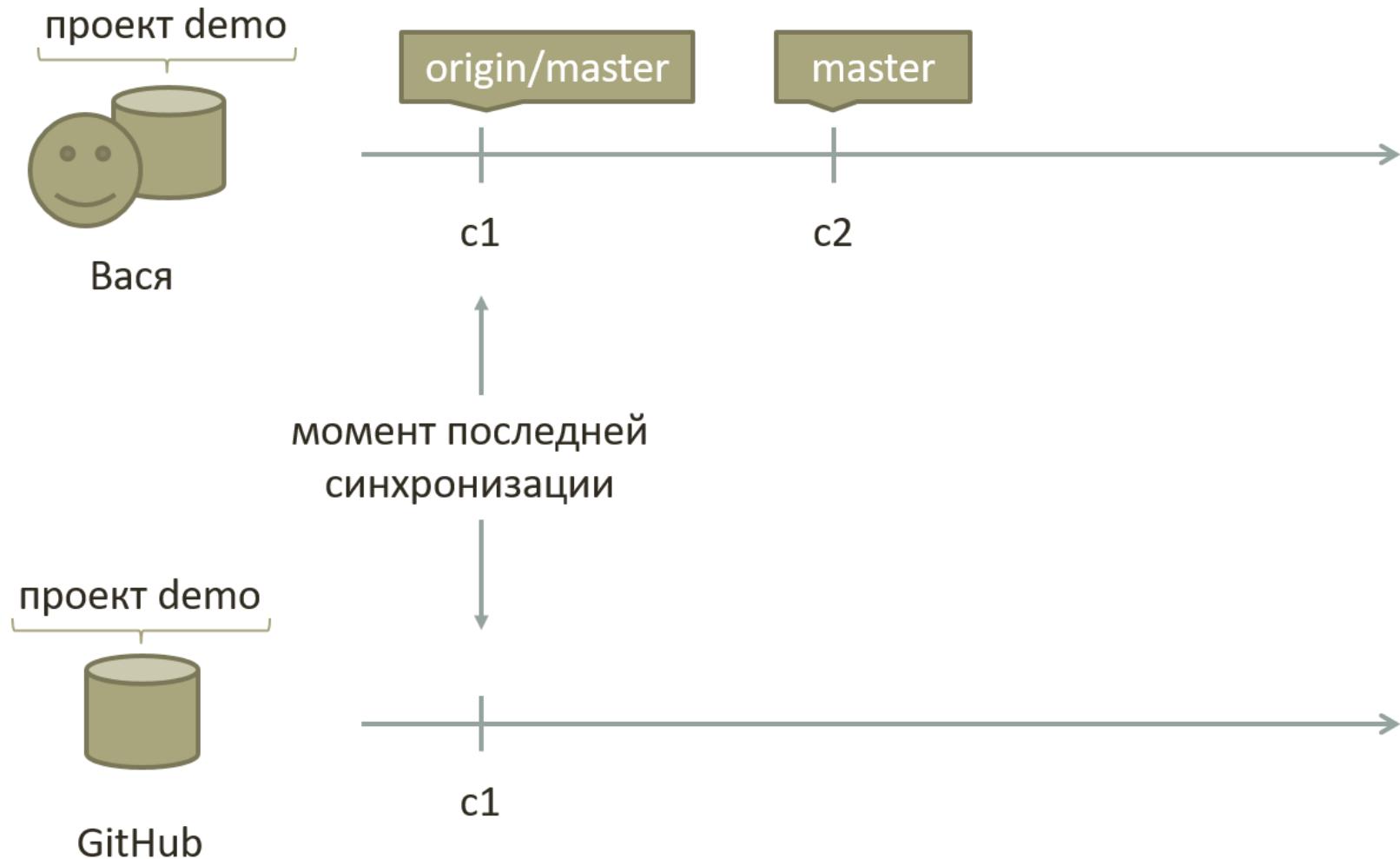
# УДАЛЕННЫЕ ВЕТКИ

---

# УДАЛЁННЫЕ ВЕТКИ

Удалённые ветки - это «специальные локальные ветки», задача которых указывать на состояние ветки в удалённом репозитории на момент последней синхронизации между вашим локальным репозиторием и удалённым.

# УДАЛЁННЫЕ ВЕТКИ



# КЛОНИРОВАНИЕ

При клонировании репозитория с удалённого сервера Git автоматически создаёт удалённую ветку `origin/master` и локальную `master`.

Для всех остальных удалённых веток локальные ветки не создаются их нужно создавать с помощью команды: `git branch <local-branch> <origin/remote-branch>`

# СОЗДАНИЕ ЛОКАЛЬНОЙ ВЕТКИ

Также можно создать локальную ветку с именем, соответствующем удалённой с помощью специальной сокращённой команды:

```
git checkout --track <remote-branch>
```

# UPSTREAM BRANCH

Когда мы с вами рассматривали команду `git push`, мы явно указывали в какую ветку на удалённом репозитории заливать изменения, кроме того указывали флаг `-u`:



Для этого используется команда `git push`, для первой отправки в формате `git push -u origin master`, в дальнейшем достаточно использовать `git push`.

# UPSTREAM BRANCH

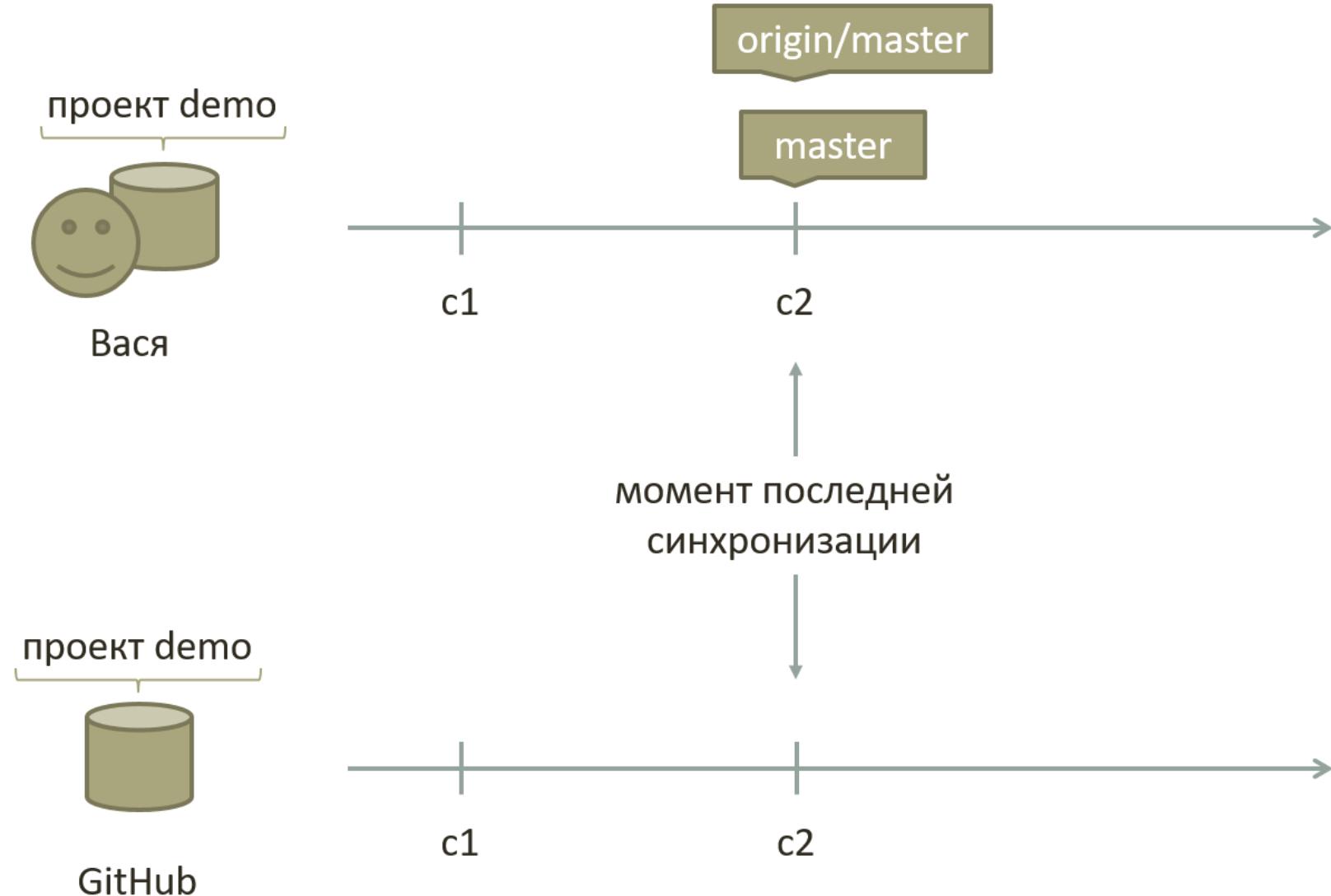
Git нам позволяет создавать так называемые «отслеживаемые ветки» – это локальные ветки, которые занимаются отслеживанием удалённых.

Флаг `-u` с командой `git push` создаёт такую связь между локальной веткой (которую вы push'ите на сервер) и удалённой.

# UPSTREAM BRANCH

Когда связь «отслеживания» создана, для большинства команд (`push`,  
`fetch`, `pull`) уже не надо указывать имя удалённой ветки

# СИНХРОНИЗАЦИЯ



# СИНХРОНИЗАЦИЯ

Для синхронизации удалённых веток Git предлагает три команды:

- `push` ;
- `fetch` ;
- `pull` .

# push

Команда `git push` используется для отправки ваших локальных коммитов (из текущей ветки) в удалённый репозиторий.

Если установлена связь «отслеживания», то достаточно в текущей ветке выполнить `git push`

# fetch

Команда `git fetch <origin>` забирает изменения с удалённого сервера (если они там есть) и сдвигает указатели удалённой ветки на нужную позицию.

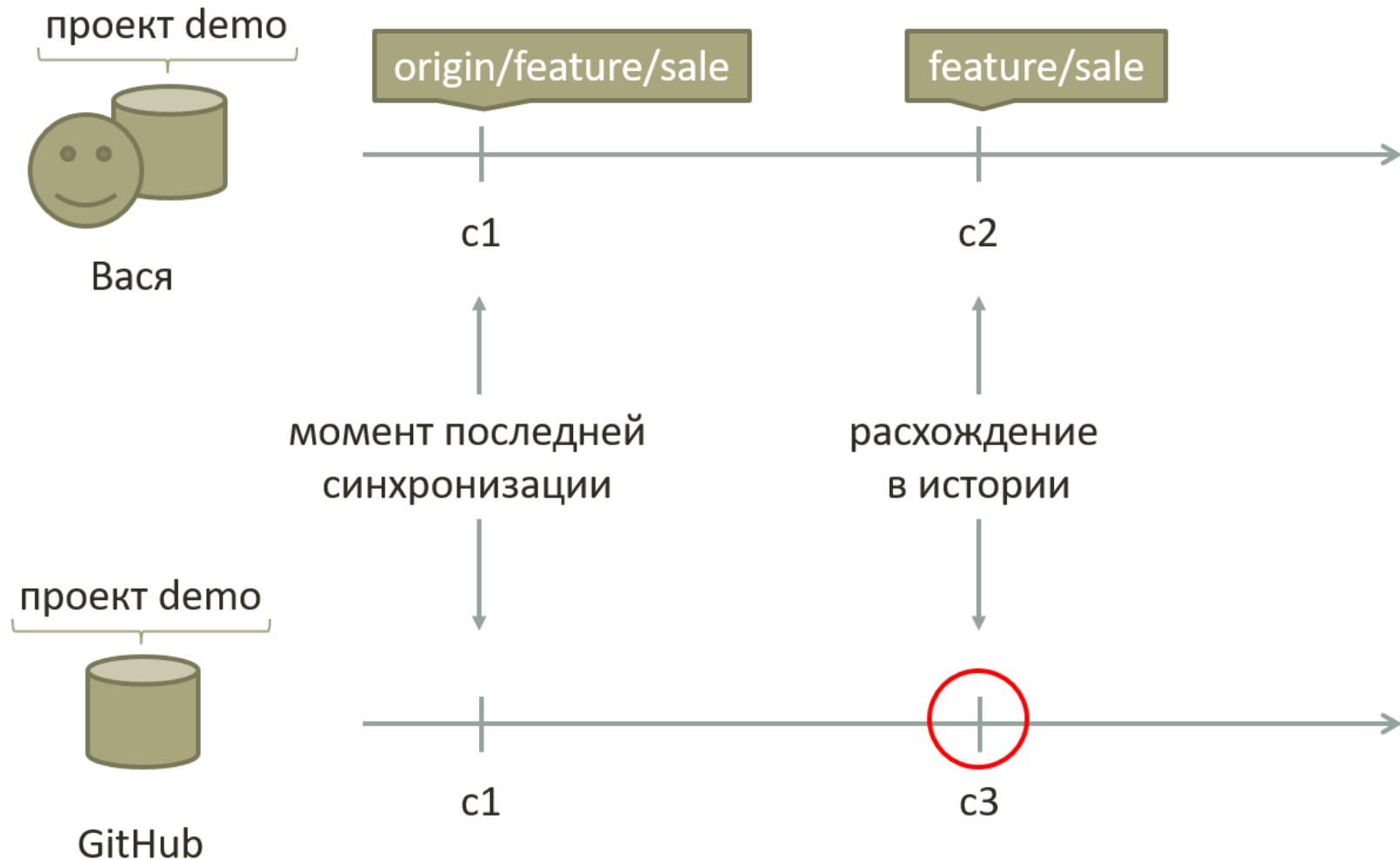
`fetch` работает с тем `branch`'ем, в котором вы сейчас работаете.

# КОНФЛИКТЫ

Вы работаете над проектом в команде в ветке `feature/sale` – ветка, которая содержит код для специальной акции по распродаже.

Случилось так, что пока вы работали локально, ваш коллега залил новые изменения в эту ветку на удалённом сервере, и теперь ситуация соответствует картинке на следующем слайде.

# СИНХРОНИЗАЦИЯ



# КОНФЛИКТЫ

Если вы попытаетесь сделать `git push` (отправить изменения на сервер), вы получите предупреждение об ошибке:

```
feature/sale$ git push
To https://github.com/netology-git/demo.git
! [rejected]      feature/sale -> feature/sale (fetch first)
error: failed to push some refs to 'https://github.com/netology-git/demo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

# git fetch

`git fetch` позволяет вам получить удалённые изменения не модифицируя файлы в working directory.

Это позволит вам в комфортном режиме отсмотреть все изменения и сделать `merge`.

# git fetch

```
feature/sale$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 2), reused 3 (delta 2), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/netology-git/demo
    fbf634f..e294b81  feature/sale    -> origin/feature/sale
```

# git fetch

```
feature/sale$ git status
```

On branch master

Your branch and 'origin/feature/sale' have diverged,  
and have 1 and 1 different commits each, respectively.  
(use "git pull" to merge the remote branch into yours)

# КАК ПОСМОТРЕТЬ ИЗМЕНЕНИЯ?

Для этого нам нужна команда `git checkout`:

```
feature/sale$ git checkout origin/feature/sale
Note: checking out 'origin/feature/sale'.
```

You are `in 'detached HEAD'` state. You can `look` around, `make` experimental changes and commit them, and you can discard any commits you `make in` this state without impacting any branches by performing another checkout.

Git предупредит нас, что мы находимся в режиме Detached HEAD.

# КАК СЛИТЬ ИЗМЕНЕНИЯ?

Каждый раз, когда вы будете переключаться на ветку `feature/sale` (локальную) git вас будет предупреждать о том, что имеются расхождения, которые нужно слить:

```
master$ git checkout feature/sale
Your branch and 'origin/feature/sale' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)
```

# КАК СЛИТЬ ИЗМЕНЕНИЯ?

Для слияния достаточно выполнить в ветке `feature/sale` команду `git merge`: процесс будет аналогичен тому, который мы рассматривали в теме про разрешение конфликтов:

```
feature/sale$ git merge
Merge made by the 'recursive' strategy.
index.html | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)
```

# КАК ЗАЛИТЬ ИЗМЕНЕНИЯ?

После merge уже можно делать `push`:

```
feature/sale$ git push
Enumerating objects: 9, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 545 bytes | 272.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/netology-git/demo.git
  e294b81..9ddbb93  feature/sale -> feature/sale
```

# pull

Команда `git pull` позволяет выполнить описанный выше двухшаговый процесс (`fetch` + `merge`) за один шаг: т.е. вы выкачиваете изменения нужной ветки и сразу переходите в режим слияния.

# pull

```
feature/sale$ git pull
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 5 (delta 3), reused 5 (delta 3), pack-reused 0
Unpacking objects: 100% (5/5), done.
From https://github.com/netology-git/demo
  e294b81..9ddb93  feature/sale    -> origin/feature/sale
Updating e294b81..9ddb93
Fast-forward
 index.html | 4 +-+-
 1 file changed, 2 insertions(+), 2 deletions(-)
```

---

# ИТОГИ

Мы рассмотрели с вами и разобрали:

- Операцию `push` и возможные конфликты;
- `fetch/pull`;
- upstream branches.

---

# ISSUES

---

# ПРОЦЕСС

GitHub – это в первую очередь платформа, обеспечивающая поддержку процесса разработки программного обеспечения.

И этот процесс включает в себя не только создание кода и систему контроля версий, но и другие механизмы, например:

- работу с сообщениями об ошибках;
- разрешение спорных ситуаций;
- предложение улучшений и т.д.

---

# ISSUES

GitHub предлагает нам функцию, называемую Issues, которая и позволяет:

- сообщать об ошибках;
- общаться разработчикам между собой (вне зависимости от часового пояса и местоположения);
- ставить задачи;
- предлагать улучшения;
- задавать вопросы.

# ISSUES

Issues уже интегрированы в каждый репозиторий и находятся на вкладке Issues:

The screenshot shows the 'Issues' tab selected in a GitHub repository named 'netology-git / demo'. The top navigation bar includes 'Watch' (1), 'Star' (0), and 'Fork' (0) buttons. Below the navigation are tabs for 'Code', 'Issues 0' (which is active and highlighted in orange), 'Pull requests 0', 'Projects 0', and 'Insights'. A search bar contains the query 'is:issue is:open'. On the right, there is a green 'New issue' button. The main content area displays a large '!' icon and the message 'There aren't any open issues.' It also suggests searching 'all of GitHub' or using 'advanced search'.

netology-git / demo

Watch 1   Star 0   Fork 0

Code   Issues 0   Pull requests 0   Projects 0   Insights

is:issue is:open   Labels   Milestones   New issue

0 Open   0 Closed   Author ▾   Labels ▾   Projects ▾   Milestones ▾   Assignee ▾   Sort ▾

!

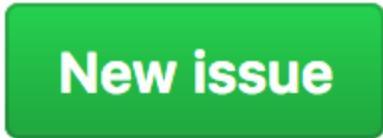
There aren't any open issues.

You could search [all of GitHub](#) or try an [advanced search](#).

---

# ISSUES

Для того, чтобы создать новое issue нужно нажать на кнопку 'New Issue':



New issue

# ISSUES

The screenshot shows the Jira 'Create Issue' page. A red box labeled '1' highlights the 'Title' input field at the top left. A red box labeled '2' highlights the large text area for 'Leave a comment'. A red box labeled '3' highlights the 'Preview' button in the toolbar, which is preceded by a left arrow icon.

Title

Write Preview ← 3

Leave a comment

AA B i “ ‘ ↵

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

---

# ISSUES

Важные поля:

1. Название;

2. Описание:

- для оформления используется Markdown;
- можно прикреплять файлы;
- обязательно пользуйтесь режимом Preview (3) перед отправкой.

---

# ISSUES

Описание позволяет вам осуществлять следующие возможности:

1. Ссылаться на Pull Request'ы (рассмотрим далее);
2. Ссылаться на конкретных людей;
3. Ссылаться на коммиты;
4. Ссылаться на другие Issue.

Детально синтаксис описан на странице:

<https://help.github.com/articles/autolinked-references-and-urls/>

# ISSUES

Также вы можете:

1. Назначить на кого-то issue (ему придёт уведомление);
2. Проставить метку (для фильтрации и поиска);
3. Выбрать Milestone.

Assignees



1

No one—assign yourself

Labels



2

None yet

Projects



None yet

Milestone



3

No milestone

# ISSUES

# ISSUES

Раздел Issues позволяет видеть текущую статистику и осуществлять фильтрацию и сортировку по всем полям.

The screenshot shows a user interface for managing issues. At the top, there are navigation links for 'Filters' (with a dropdown arrow), a search bar containing 'is:issue is:open', 'Labels', 'Milestones', and a green 'New issue' button. Below this is a summary row with filters: '1 Open' (unchecked) and '0 Closed' (checked). To the right are dropdown menus for 'Author', 'Labels', 'Projects', 'Milestones', 'Assignee', and 'Sort'. The main list area displays a single issue: '#2 opened 7 minutes ago by netology-git'. The issue title is 'Баг в отображении формы "Подписаться"' and it is labeled 'bug'. A small pink icon is located to the right of the issue details.

Open	Closed
1	0

Author ▾ Labels ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾

① Баг в отображении формы "Подписаться" bug  
#2 opened 7 minutes ago by netology-git

---

# ISSUES

У Issue может быть несколько статусов:

1.  Означает, что вопрос ещё не решён, либо задача ещё не выполнена;
2.  Означает, что вопрос уже решён, либо задача выполнена.

# ISSUES

В рамках Issue можно вести обсуждения, закрывать и переоткрывать их при необходимости.

Это позволяет в одном месте хранить и исходные коды и историю работы со всеми задачами.

# LABELS - МЕТКИ

Метки позволяют организовать Issues, визуально выделяя их и давая возможность осуществлять фильтрацию.

Вы можете создать свою метку перейдя раздел Labels:

The screenshot shows a software interface for managing issues. At the top, there is a navigation bar with 'Filters' (set to 'is:issue is:open'), a search bar, and buttons for 'Labels', 'Milestones', and 'New issue'. A red circle with the number '1' is drawn around the 'Labels' button. Below the navigation bar, there are filters for '1 Open' and '0 Closed' issues. The main list displays one issue: '#2 opened 12 minutes ago by netology-git' with a 'bug' label and a 'feature' label. The 'feature' label is highlighted with a green box. At the bottom, a modal window is open for creating a new label named 'feature'. It includes fields for 'Label name' (containing 'feature'), 'Description' (containing 'Optional description'), 'Color' (set to '#00ff00'), and buttons for 'Cancel' and 'Create label'.

GitHub позволяет вам закрывать Issue с помощью специального синтаксиса в сообщениях коммитов, например:

```
fixes #2 Добавили форму для отображения
```

```
fixes #2 говорит о том, что закрывает issue с id = 2.
```

Нужно отметить, что issue закрывается только тогда, когда коммит с этим комментарием попадет в основную ветку вашего репозитория (по умолчанию - master).

Ссылка на описание: <https://help.github.com/articles/closing-issues-using-keywords/>

# MILESTONES - ВЕХИ

Значимые вехи проекта. Вы можете создавать свои, перейдя на вкладку Milestones в Issues:

The screenshot shows a project management interface with the following elements:

- Filters:** A dropdown labeled "Filters" and a search bar containing "is:issue is:open".
- Search Bar:** A search bar with a magnifying glass icon.
- Labels:** A button labeled "Labels".
- Milestones:** A button labeled "Milestones" with a red box drawn around it. A red arrow points from the number "1" in a circle to this button.
- New issue:** A green button labeled "New issue".
- Filter Options:** Buttons for "1 Open" (unchecked) and "0 Closed" (checked).
- Sort Options:** Buttons for "Author", "Labels", "Projects", "Milestones", "Assignee", and "Sort".
- Issue List:** A list of issues:
  - Issue 1:** Status: Open, Type: bug, Title: Баг в отображении формы "Подписаться", #2 opened 12 minutes ago by netology-git. It has a red "bug" label.
- Icons:** A small "xx" icon next to the issue list and a larger "xx" icon at the bottom right of the interface.

# MILESTONES - БЕХИ

## New milestone

Create a new milestone to help organize your issues and pull requests. Learn more about [milestones and issues](#).

Title

Due date (optional)

Description

Create milestone

---

# PULL REQUESTS

# GITHUB FLOW

GitHub предлагает легковесную модель разработки (GitHub Flow):

<https://guides.github.com/introduction/flow/>

В рамках этой модели, для любых экспериментов (новые функции, багфиксинг и т.д.), вы следуете простому шаблону:

1. Создаёте `branch`;
2. В рамках `branch`'а коммите изменения;
3. Открываете Pull Request для обсуждения ваших изменений с коллегами (code review, решение вопросов);
4. Делаете `merge`.

---

# PULL REQUESTS

GitHub предлагает модели, подходящие под разные ситуации:

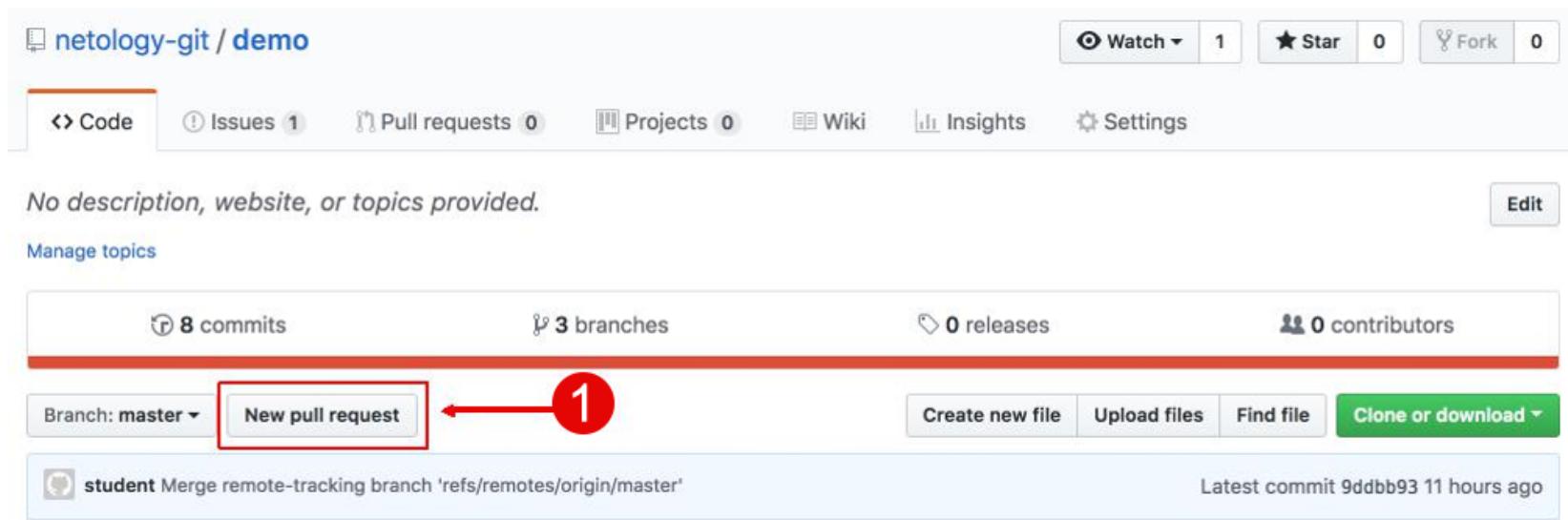
1. Shared Repository – вариант, когда вы являетесь полноценным участником репозитория и можете делать там коммиты;
2. Fork and Pull – вариант, когда вы хотите предложить изменения в чужой репозиторий, либо не хотите хранить исходные коды своего эксперимента в общем репозитории.

# ВАЖНЫЕ ПОНЯТИЯ

- **base** – то, куда мы собираемся делать Pull Request;
- **head** – то, что мы собираемся использовать в качестве Pull Request'a.

# SHARED REPOSITORY

Как только у вас в проекте появляется больше одной ветки (удалённых) вы можете сделать Pull Request:



# PULL REQUEST

Форма для открытия Pull Request'a очень сильно похожа на Issue, но тут вы можете назначить Reviewer'a - того, кто будет проверять ваши изменения.

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

The screenshot shows a pull request creation interface. At the top, it displays the base branch as 'master' and the compare branch as 'feature/experiment'. A green checkmark indicates that the branches are 'Able to merge'. The main area contains a text input field with placeholder text 'Marketing campaing page', a 'Write' button, a 'Preview' button, and a rich text editor toolbar. Below this is a comment input field with placeholder 'Leave a comment'. Further down is a file attachment section with the instruction 'Attach files by dragging & dropping, selecting them, or pasting from the clipboard.' At the bottom, there is a note about Markdown support and a large green 'Create pull request' button. To the right of the main form, there are several configuration sections: 'Reviewers' (with 'netology-reviewer' selected), 'Assignees' (with 'netology-git'), 'Labels' (set to 'None yet'), 'Projects' (set to 'None yet'), and 'Milestone' (set to 'No milestone'). The 'Reviewers' section is highlighted with a red box.

# PULL REQUEST

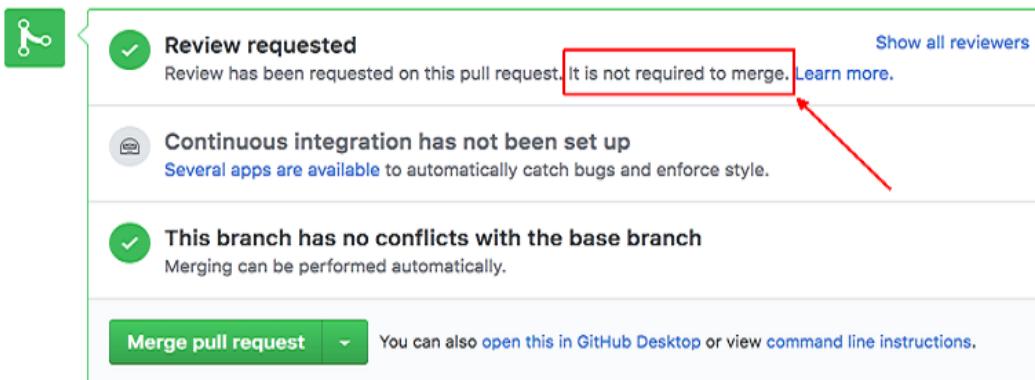
Ниже формы представлен `diff` между бранчами:

The screenshot shows a GitHub pull request interface. At the top, it displays summary statistics: 1 commit, 2 files changed, 0 commit comments, and 1 contributor. Below this, the commit details show a commit made by 'student' on Nov 23, 2018, titled 'Marketing campaing page' with hash e239e04. A note indicates 'Showing 2 changed files with 14 additions and 0 deletions.' The main content is a diff view for the file 'campaign.html'. The diff shows 14 additions (marked with '+') and 0 deletions (marked with '-'). The code changes are as follows:

```
1 + <!DOCTYPE html>
2 + <html lang="en">
3 + <head>
4 +   <meta charset="UTF-8">
5 +   <meta name="viewport" content="width=device-width, initial-
scale=1.0">
6 +   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 +   <title>Document</title>
8 +   <link rel="stylesheet" href="css/style.css">
9 + </head>
10 + <body>
11 +   <h1>Sale page</h1>
12 + </body>
13 + </html>
```

# PULL REQUEST

Если конфликтов при Merge между бранчами не обнаружено, вы можете не дожидаться Review и сделать Merge (также это зависит от типа вашего репозитория и настроек).



# PULL REQUEST

Reviewer'у придёт уведомление о том, что вы запросили Review:

netology-git requested your review on this pull request.

Add your review

## Marketing campaing page #3

Open netology-git wants to merge 1 commit into master from feature/experiment

Conversation 0 Commits 1 Checks 0 Files changed 2 +14 -0

netology-git commented a minute ago  
No description provided.

Marketing campaing page e239e04  
netology-git self-assigned this a minute ago  
netology-git requested a review from coursar a minute ago

Reviewers netology-reviewer  
Assignees netology-git  
Labels None yet  
Projects None yet

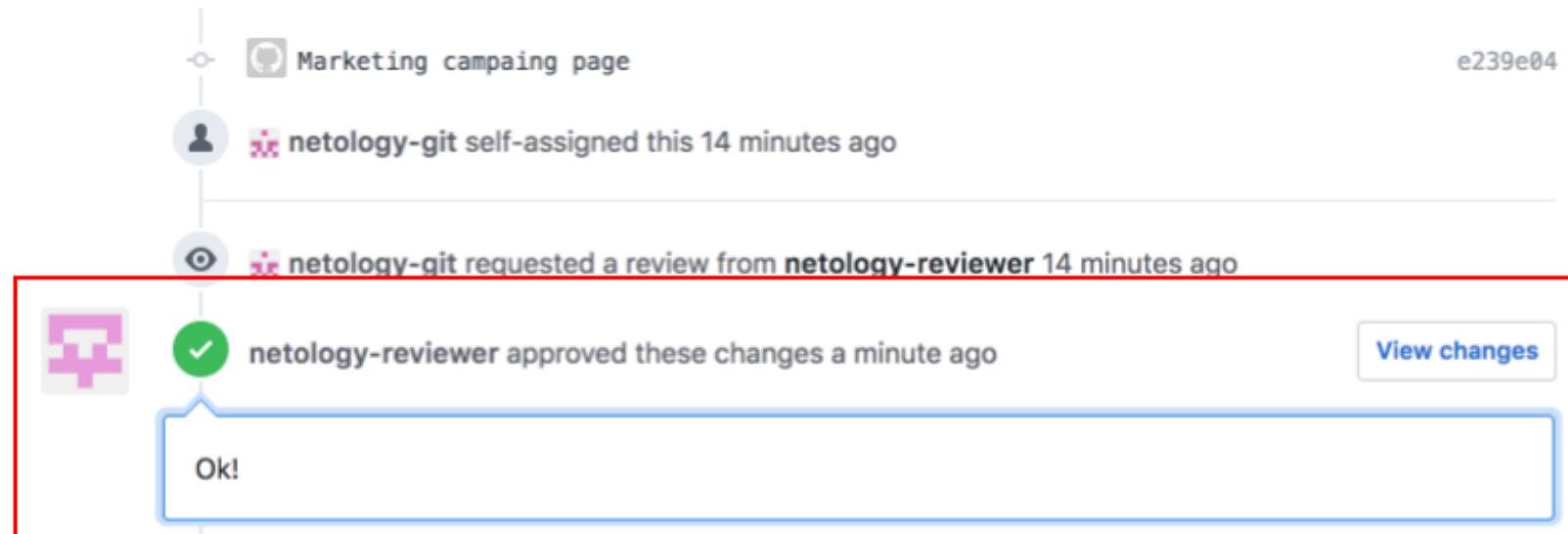
Add more commits by pushing to the feature/experiment branch on netology-git/demo.

# PULL REQUEST

Он может выбрать один из трёх вариантов:

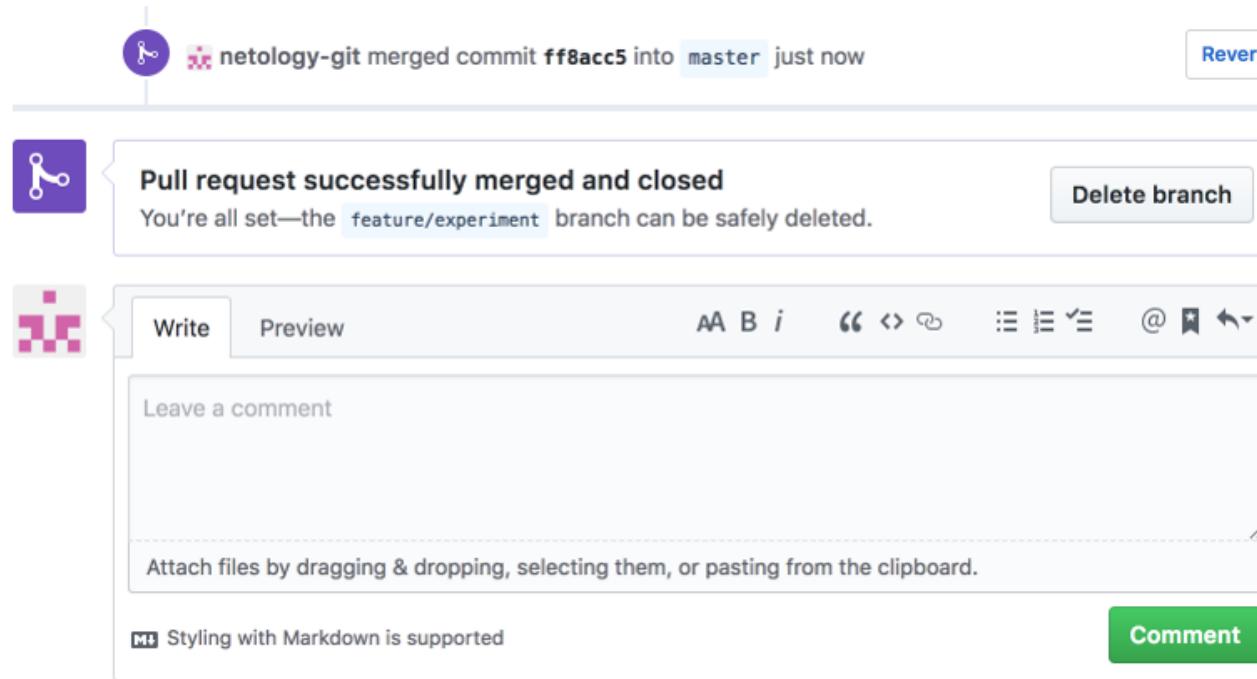
# PULL REQUEST

После чего в истории Pull Request'a появится информация о Review:



# PULL REQUEST

После операции Merge создаётся Merge Commit и Pull Request будет закрыт:



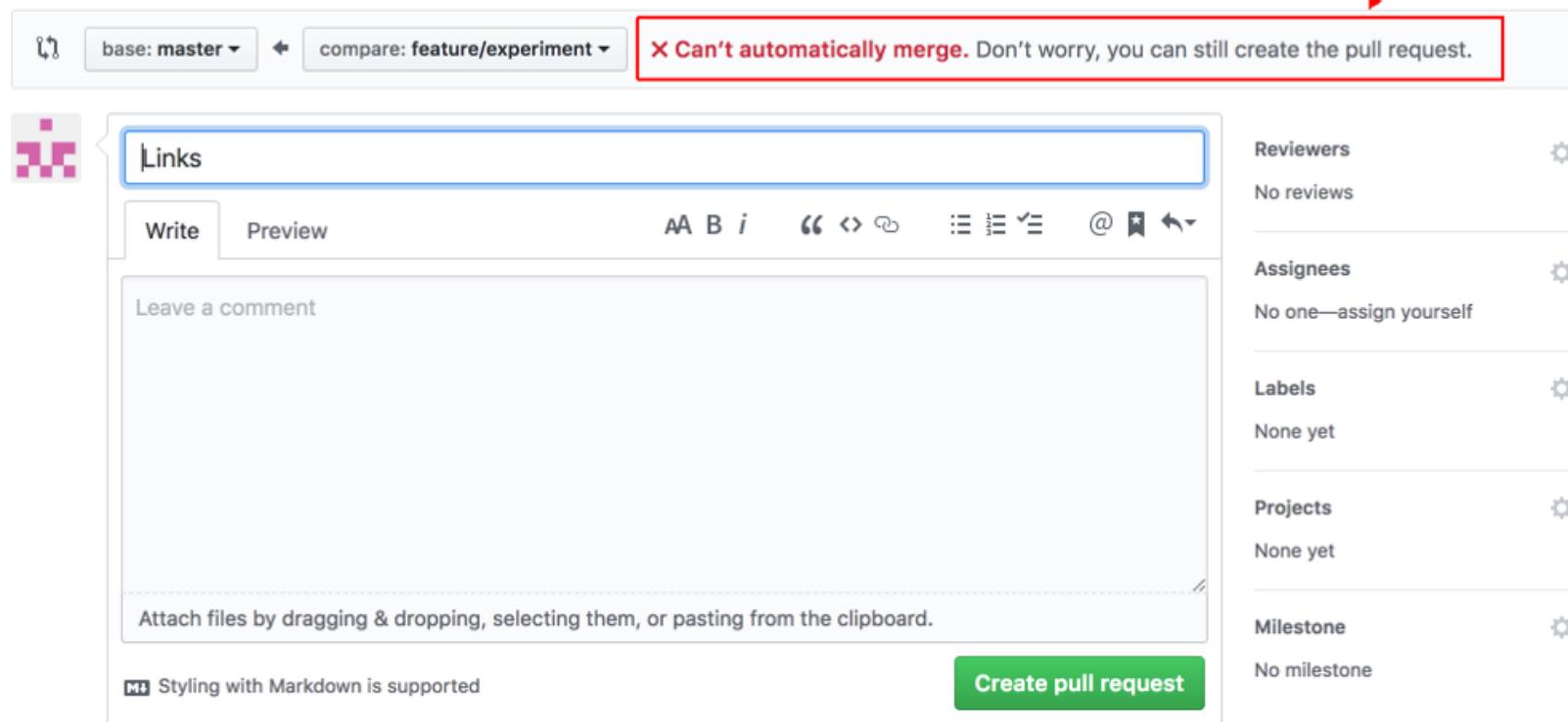
Вам необходимо будет выполнить синхронизацию веток.

# PULL REQUEST

В случае, если конфликт при Merge присутствует, GitHub сообщит об этом:

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



The screenshot shows the GitHub interface for creating a pull request. At the top, there are dropdown menus for 'base: master' and 'compare: feature/experiment'. A red box highlights a red X icon followed by the text 'Can't automatically merge. Don't worry, you can still create the pull request.' To the right of the main form, there are sections for 'Reviewers' (No reviews), 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), and 'Milestone' (No milestone). At the bottom left, it says 'Styling with Markdown is supported', and at the bottom right is a green 'Create pull request' button.

# PULL REQUEST

Необходимо будет разрешить конфликт:

The screenshot shows a GitHub pull request interface. At the top, there is a warning message: "This branch has conflicts that must be resolved. Use the [web editor](#) or the [command line](#) to resolve conflicts." A red box highlights the "Resolve conflicts" button, which is located in a callout box above the "Conflicting files" section. Below this, the "Conflicting files" section lists "index.html". Further down, there is a "Merge pull request" button with a dropdown menu, and a note stating "You can also [open this in GitHub Desktop](#) or view [command line instructions](#)." At the bottom, there is a rich text editor toolbar with "Write" and "Preview" tabs, a comment input field ("Leave a comment"), a file attachment area ("Attach files by dragging & dropping, selecting them, or pasting from the clipboard."), and two buttons: "Close pull request" and "Comment".

# PULL REQUEST

Сделать это можно как с помощью интерфейса GitHub, так и локально:

This screenshot shows a GitHub pull request interface with a red box highlighting the 'Checkout via command line' section. A red arrow points from the 'command line instructions' link in the 'Merge pull request' dropdown to this section.

**This branch has conflicts that must be resolved**  
Use the [web editor](#) or the [command line](#) to resolve conflicts.

**Resolve conflicts**

**Conflicting files**  
index.html

Merge pull request ▾ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

**Checkout via command line**  
If you cannot merge a pull request automatically here, you have the option of checking it out via command line to resolve conflicts and perform a manual merge.

HTTPS Git Patch <https://github.com/netology-git/demo.git>

Step 1: From your project repository, bring in the changes and test.

```
git fetch origin  
git checkout -b feature/experiment origin/feature/experiment  
git merge master
```

Step 2: Merge the changes and update on GitHub.

```
git checkout master  
git merge --no-ff feature/experiment  
git push origin master
```

# PULL REQUEST

Интерфейс GitHub для разрешения конфликта:

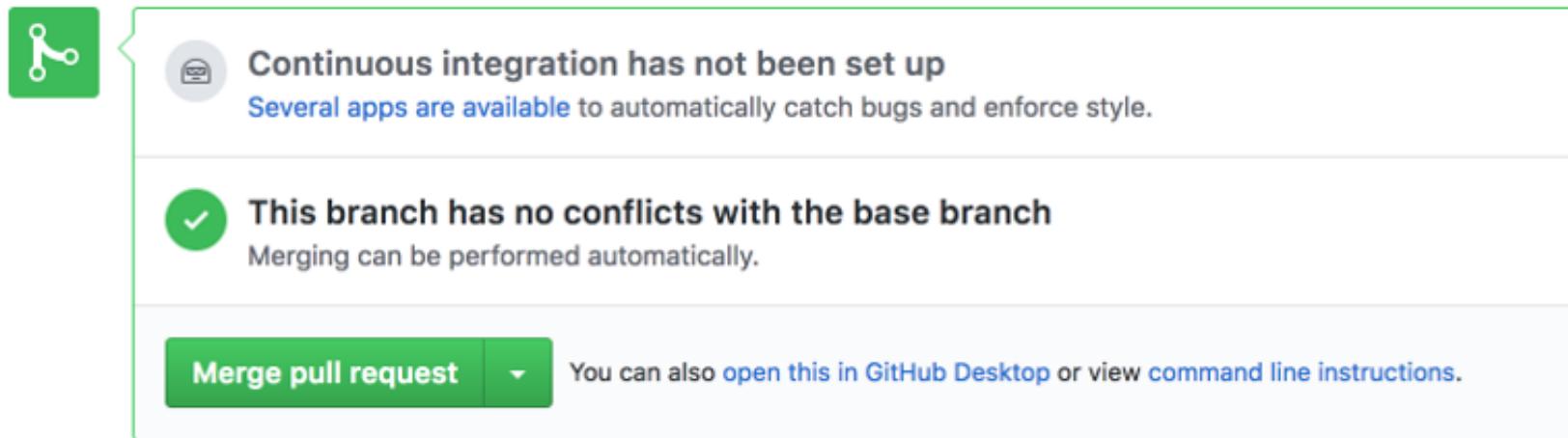
## Links #4

Resolving conflicts between `feature/experiment` and `master` and committing changes ➔ `feature/experiment`

1 conflicting file	index.html	1 conflict	Prev ^	Next ▼
 <code>index.html</code> <code>index.html</code>	<pre>1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3  &lt;head&gt; 4      &lt;meta charset="UTF-8"&gt; 5      &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt; 6      &lt;meta http-equiv="X-UA-Compatible" content="ie=edge"&gt; 7      &lt;title&gt;Document&lt;/title&gt; 8      &lt;link rel="stylesheet" href="css/style.css"&gt; 9  &lt;/head&gt; 10 &lt;body&gt; 11     &lt;h1&gt;Hello world (Updated)&lt;/h1&gt; 12     &lt;a href="campaign.html"&gt;Marketing Campaign&lt;/a&gt; 13     &lt;===== feature/experiment 14     &lt;a href="sale.html"&gt;Sale&lt;/a&gt; 15     &lt;a href="contactS.html"&gt;Sale&lt;/a&gt; 16     ===== 17     &lt;a href="partners.html"&gt;Partners&lt;/a&gt; 18     &gt;&gt;&gt;&gt;&gt; master 19     &lt;script src="js/app.js"&gt;&lt;/script&gt; 20 21 22</pre>	1 conflict	Prev ^	Next ▼

# PULL REQUEST

После разрешения конфликта станет доступной возможность Merge Pull Request'a:



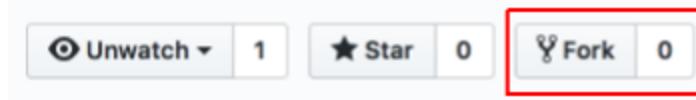
The screenshot shows a GitHub pull request merge status interface. On the left, there's a green icon of two interlocking gears. To its right, a grey circular icon contains a white envelope symbol. Below these are two main status sections:

- Continuous integration has not been set up**  
Several apps are available to automatically catch bugs and enforce style.
- This branch has no conflicts with the base branch**  
Merging can be performed automatically.

At the bottom left is a large green button labeled "Merge pull request". To its right is a dropdown arrow. At the bottom right, there's a note: You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

# FORK AND PULL

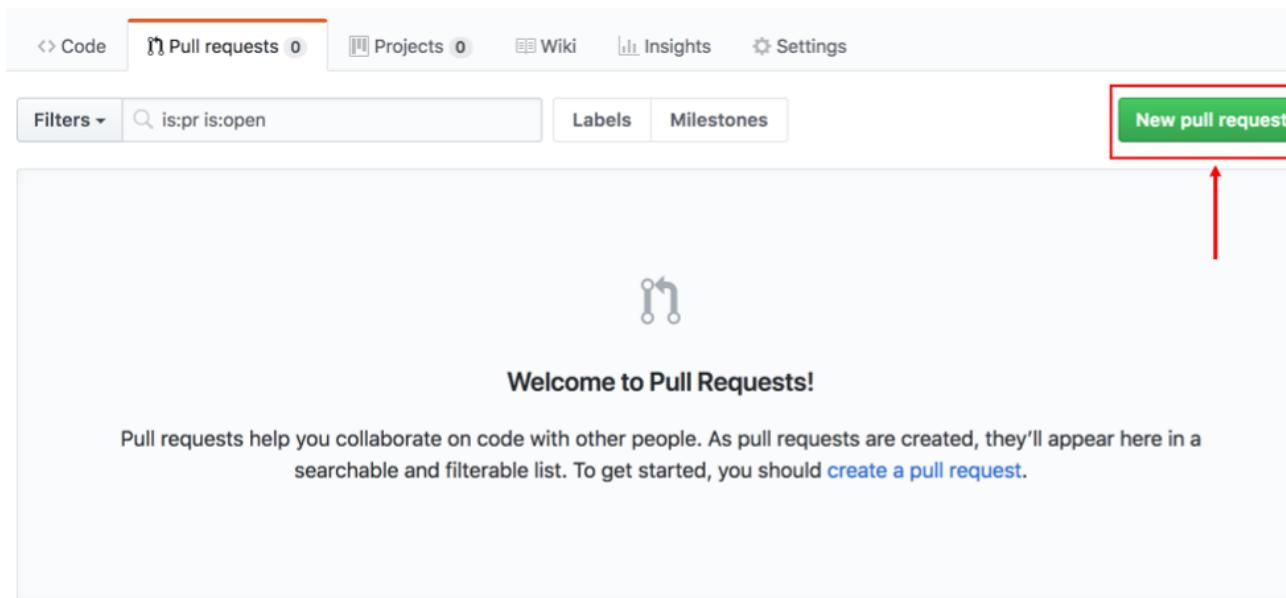
Если у вас нет прав на запись в репозиторий, но вы хотите предложить в этот репозиторий улучшения, то воспользуйтесь кнопкой Fork.



# FORK AND PULL

В результате операции Fork вы получите копию репозитория под собственным управлением.

После внесения всех необходимых улучшений вы можете создать Pull Request в исходный репозиторий:



# FORK AND PULL

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base fork: netology-git/demo ▾ base: master ▾ ← head fork: netology-forker/demo ▾ compare: master ▾ 1

✓ Able to merge. These branches can be automatically merged.

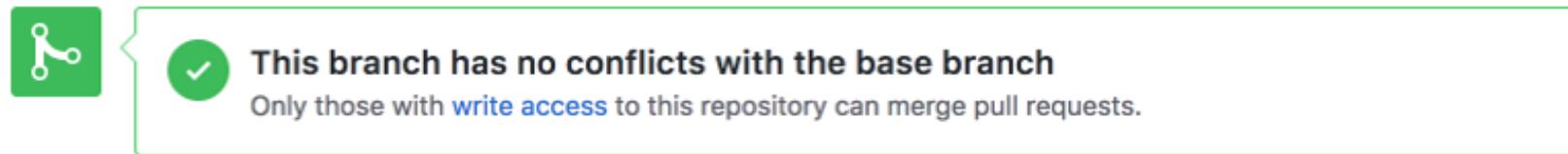
Create pull request 2 Discuss and review the changes in this comparison with others. ?

-o 1 commit 1 file changed 0 commit comments 1 contributor

Commits on Nov 23, 2018  
netology-forker Extension 39e8057

# FORK AND PULL

Если у вас нет доступа на запись в исходный репозиторий, вы не сможете сделать Merge.



Его должен будет сделать кто-то из участников, обладающих подобным правом (запрос на Pull Request появится во вкладке Pull Requests проекта).

# FORK AND PULL

В случае, если нет возможности сделать Merge с автоматическим разрешением конфликта, GitHub сообщит об этом так же, как в случае модели Shared Repository.

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

The screenshot shows the GitHub interface for comparing changes between two branches. At the top, there are four dropdown menus: 'base fork: netology-git/demo', 'base: master', 'head fork: netology-forker/demo', and 'compare: master'. Below these, a red-bordered box contains the message: 'X Can't automatically merge. Don't worry, you can still create the pull request.' A red arrow points from the right side of the slide towards this message. At the bottom, there is a green button labeled 'Create pull request' with a icon, and the text 'Discuss and review the changes in this comparison with others.' To the right of the discussion text is a help icon (a question mark inside a circle).

# FORK AND PULL

Так же GitHub уведомит об отсутствии доступа на запись для Merge:



# FORK AND PULL

При этом разрешение конфликта (при отсутствии прав на запись в исходный репозиторий) будет записано в fork:

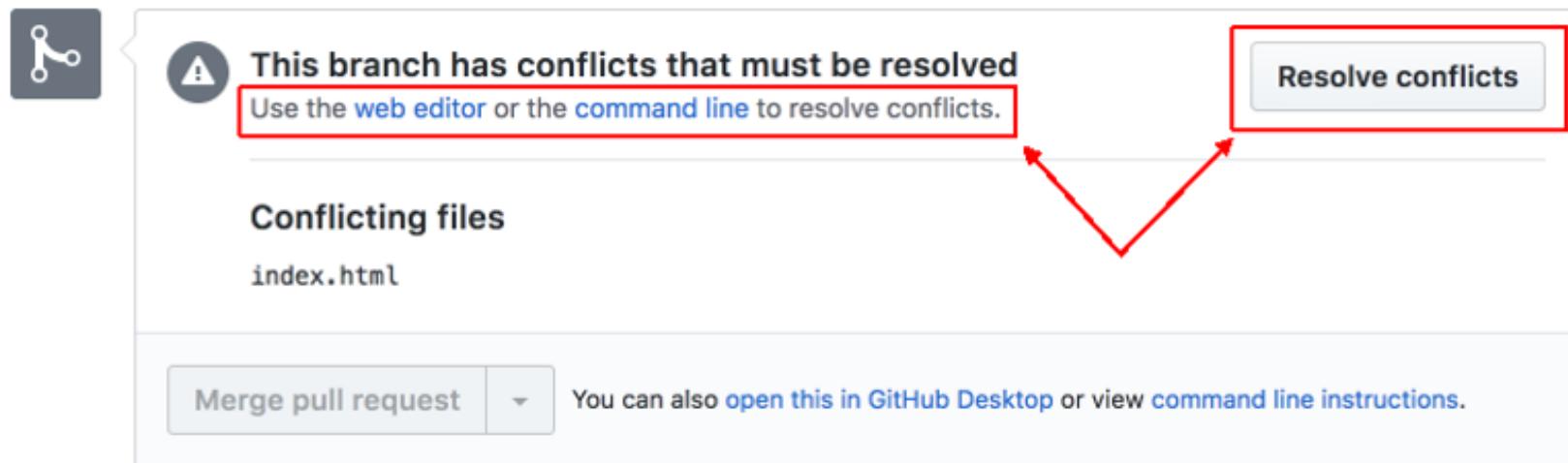
Conflict Pull request #6

Resolving conflicts between `netology-forker:master` and `netology-git:master` and committing changes ➔ `netology-forker:master`

1 conflicting file	index.html	1 conflict	Prev ⌂	Next ⌂	⚙️
<code>index.html</code>	<pre>1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3  &lt;head&gt; 4      &lt;meta charset="UTF-8"&gt; 5      &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt; 6      &lt;meta http-equiv="X-UA-Compatible" content="ie=edge"&gt; 7      &lt;title&gt;Document&lt;/title&gt; 8      &lt;link rel="stylesheet" href="css/style.css"&gt; 9  &lt;/head&gt; 10 &lt;body&gt; 11     &lt;h1&gt;Hello world (Updated)&lt;/h1&gt; 12     &lt;&lt;&lt;&lt;&lt; master 13         &lt;div class="hero"&gt;Our global mission&lt;/div&gt; 14     ===== 15         &lt;a href="campaign.html"&gt;Marketing Campaign&lt;/a&gt; 16         &lt;a href="partners.html"&gt;Partners&lt;/a&gt; 17         &lt;a href="store.html"&gt;Store&lt;/a&gt; 18         &lt;a href="about.html"&gt;About&lt;/a&gt; 19     &gt;&gt;&gt;&gt;&gt; master 20         &lt;script src="js/app.js"&gt;&lt;/script&gt; 21     &lt;/body&gt; 22 &lt;/html&gt;</pre>	1 conflict	Prev ⌂	Next ⌂	⚙️

# FORK AND PULL

При этом разработчик, обладающий правами на запись в исходный репозиторий может сделать Merge с разрешением конфликта:



# FORK AND PULL

После того, как конфликт будет разрешён какой-либо из сторон, появится возможность сделать Merge (у разработчика, обладающего правами на запись):

netology-forker and others added some commits 2 hours ago

- Menu enhancement 44f7daa
- Global mission d624015
- Merge branch 'master' into master Verified 2c4e0c1

Add more commits by pushing to the **master** branch on **netology-forker/demo**.

Continuous integration has not been set up  
Several apps are available to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch  
Merging can be performed automatically.

**Merge pull request** You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

A red arrow points to the "Merge pull request" button at the bottom left of the screenshot.

---

# GITHUB PAGES

---

# GITHUB PAGES

GitHub Pages - сервис, позволяющий публиковать веб-сайты, используя мощности GitHub.

---

# ПРЕДВАРИТЕЛЬНЫЕ ТРЕБОВАНИЯ

Необходимо создать репозиторий на GitHub и склонировать его на локальную машину.

# BRANCH

GitHub использует бранч со специальным именем **gh-pages** в качестве хранилища файлов будущего веб-сайта:

```
master$ git checkout -b gh-pages
Switched to a new branch 'gh-pages'

gh-pages$ git add index.html
gh-pages$ git commit -m "Index page"
[gh-pages (root-commit) 1f8d31a] index page
 1 file changed, 12 insertions(+)
  create mode 100644 index.html
gh-pages$ git push -u origin gh-pages
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 391 bytes | 391.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/netology-git/github-pages.git
 * [new branch]      gh-pages -> gh-pages
Branch 'gh-pages' set up to track remote branch 'gh-pages' from 'origin'.
```

# ENVIRONMENT

GitHub Pages создаст веб-сайт по адресу:

`https://< ваш логин >.github.io/<название репозитория>/`

В интерфейсе можно увидеть на вкладке "Environment" репозитория:

The screenshot shows a GitHub repository interface. At the top, there are navigation links: Code (highlighted in orange), Issues 0, Pull requests 0, Projects 0, Wiki, Insights, and Settings. Below this, a message says "No description, website, or topics provided." with an "Edit" button. A "Manage topics" link is also present. In the center, there are summary statistics: 1 commit, 1 branch, 0 releases, 1 environment (which is highlighted with a red box), and 1 contributor. At the bottom, there are buttons for Branch: gh-pages ▾, New pull request, Create new file, Upload files, Find file, and Clone or download ▾.

# DEPLOYMENT

На странице будет указана ссылка на сам веб-сайт и история развертываний:

Deployed to **github-pages**

[1f8d31a](#) was deployed by **coursar** 11 minutes ago Active

[View deployment](#)

Activity log

Show: All environments ▾

 **github-pages** at 1f8d31a

Deployed by **coursar** 11 minutes ago Active

[View deployment](#)

---

# ИТОГИ

Сегодня мы с вами разобрали достаточно много вопросов:

1. Работа с удалёнными ветками;
2. Работа с Issues на GitHub;
3. Pull Request'ы (модели Shared Repository, Fork & Pull);
4. GitHub Pages.



**Задавайте вопросы и напишите отзыв о лекции!**

**ИЛЬНАЗ ГИЛЬЯЗОВ**

 [coursar@gmail.com](mailto:coursar@gmail.com)