

[git](#)

Git — это распределённая система управления версиями (DVCS). Она была создана Линусом Торвальдсом в 2005 году специально для разработки ядра Linux. Git позволяет разработчикам отслеживать изменения в коде, управлять различными версиями проекта, создавать ветки для параллельной разработки и объединять изменения обратно в основную ветку.

Основные концепции Git:

Репозиторий (Repository)

Это каталог, содержащий файлы проекта вместе с историей изменений каждого файла. Каждый репозиторий имеет свою собственную историю версий, состоящую из коммитов.

Клон (Clone)

Создание локальной копии удалённого репозитория. Это позволяет вам иметь полную копию проекта, включая всю историю изменений, прямо на вашем компьютере.

Ветви (Branches)

Отделение отдельных линий развития проекта. Каждая ветвь представляет собой отдельную линию истории, позволяющую экспериментировать с изменениями без риска повредить основной проект.

Коммит (Commit)

Запись текущего состояния файлов в историю изменений. Когда вы делаете коммит, Git сохраняет снимок всех файлов, которые были изменены с момента последнего коммита.

Фиксация (Push/Pull)

Передача ваших локальных изменений на удалённый сервер (push) или получение обновлений с сервера (pull).

Разрешение конфликтов (Conflict Resolution)

Иногда два разработчика меняют одну и ту же строку кода одновременно. Git обнаружит конфликт и попросит вас вручную решить проблему перед слиянием изменений.

Основные команды Git

Вот некоторые базовые команды, которые используются чаще всего:

Создание нового репозитория

git init

Получение существующего репозитория

git clone <url>

Просмотр статуса текущих изменений

git status

Добавление файлов в индекс (для подготовки к коммиту)

git add <file>

Запись изменений в репозиторий

git commit -m "Сообщение коммита"

Отправка изменений на удалённый сервер

git push origin main

Загрузка изменений с удалённого сервера

git pull origin main

Создание новой ветви

git branch <branch_name>

Переключение на существующую ветвь

git checkout <branch_name>

Почему используют Git?

- **История изменений:** Возможность вернуться к предыдущему состоянию вашего проекта.
- **Параллельная разработка:** Несколько разработчиков могут работать над проектом одновременно, используя разные ветви.
- **Безопасность:** Все операции в Git являются атомарными, что предотвращает потерю данных.
- **Эффективность:** Высокая производительность даже при работе с большими проектами благодаря эффективному хранению и обработке данных.

Популярные сервисы хостинга Git

Некоторые известные сервисы для размещения проектов на основе Git включают:

- **GitHub**

- **GitLab**
- **Bitbucket**

Эти сервисы предоставляют веб-интерфейсы для просмотра кода, отслеживания проблем, организации совместной работы и многое другое. Таким образом, Git является мощным инструментом для контроля версий и сотрудничества среди разработчиков, обеспечивающим гибкость и надежность в управлении проектами любого масштаба.

У меня в локальном git все время образовывается большое кол-во вот таких файлов. delete mode 100644 CsvSln/.vs/CsvSln/FileContentIndex/341b8f37-5169-4b45-b3c5-54f0a916243d.vsidx create mode 100644

CsvSln/.vs/CsvSln/FileContentIndex/6bf8026c-0f3d-4946-8b31-ff8ec238b2da.vsidx create mode 100644 CsvSln/.vs/CsvSln/FileContentIndex/750703c0-00b3-48a3-8ece-dd7f6716372a.vsidx delete mode 100644

CsvSln/.vs/CsvSln/FileContentIndex/871f0638-85ca-40da-8c83-4df64b1972d3.vsidx create mode 100644 CsvSln/.vs/CsvSln/FileContentIndex/931003e0-8bc4-4ada-bf04-0f58fb19dfab.vsidx create mode 100644

CsvSln/.vs/CsvSln/FileContentIndex/9cc7028b-4914-45f7-866c-0ad355ea0e10.vsidx delete mode 100644 CsvSln/.vs/CsvSln/FileContentIndex/9f12b2b5-c11f-41f6-b37b-1269f44457ff.vsidx delete mode 100644

CsvSln/.vs/CsvSln/FileContentIndex/c34dd8d4-9869-4ca7-b816-8f3dc142072e.vsidx delete mode 100644 CsvSln/.vs/CsvSln/FileContentIndex/ded2031a-09c2-4e94-a7b1-6f912513d157.vsidx create mode 100644

CsvSln/.vs/CsvSln/FileContentIndex/fd103354-fb10-4df4-9e93-a8e85c5e0578.vsidx
Что это, что с ними делать ?

Ответ

Файлы вида *.vsidx, которые появляются и исчезают в директории .vs, создаются Visual Studio для индексации содержимого файлов проекта. Эти файлы помогают IDE быстрее искать символы, переменные и классы внутри проекта, улучшая производительность автодополнения и навигации по коду. Однако такие временные файлы не нужны для версионного контроля и синхронизации исходников. Поэтому рекомендуется исключить их из системы контроля версий Git, добавив в файл .gitignore. Чтобы избавиться от появления этих файлов в списке изменений, выполните следующие шаги:

Шаг 1: Создайте или откройте файл .gitignore

Откройте терминал или командную строку в корне вашего проекта и создайте файл .gitignore, если его ещё нет:

```
touch .gitignore
```

Или откройте существующий файл .gitignore в любом текстовом редакторе.

Шаг 2: Добавьте правила игнорирования

Добавьте в этот файл следующую строчку, чтобы исключить папку .vs целиком:

```
.vs/
```

Теперь все содержимое каталога .vs будет автоматически игнорироваться Git'ом.

Шаг 3: Удалите лишние файлы из индекса Git

Если ранее файлы уже попали в индекс Git, очистите их:

```
git rm -r --cached .
```

```
git add .
```

```
git commit -m "Ignore VS-generated files in the future."
```

Эта команда уберёт ненужные файлы из следующего коммита, оставив их на диске.

Итог

Теперь ваша рабочая копия станет чище, и подобные файлы больше не будут попадать в список изменений, позволяя сосредоточиться исключительно на значимой части проекта.