

Создание проектов с использованием контроля версий в Visual Studio

1. Зарегистрируйте аккаунт на Github.com
2. Создайте новый репозиторий на Github.com

Нажмите «New» (рис. 1)

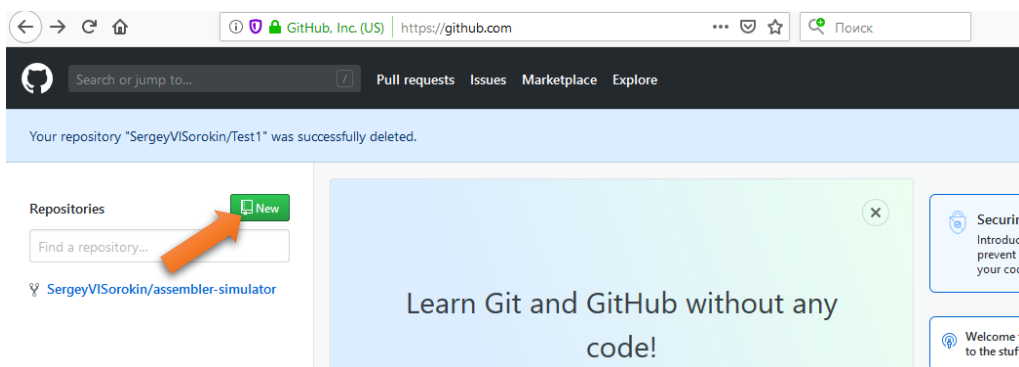


Рис. 1. Создание нового репозитория

Введите имя, например «Compiler». В списке **Add .gitignore** выберите «Visual Studio» и нажмите «Create repository» (рис. 2). Не изменяйте других настроек. Если не добавить правильный .gitignore будет много проблем!

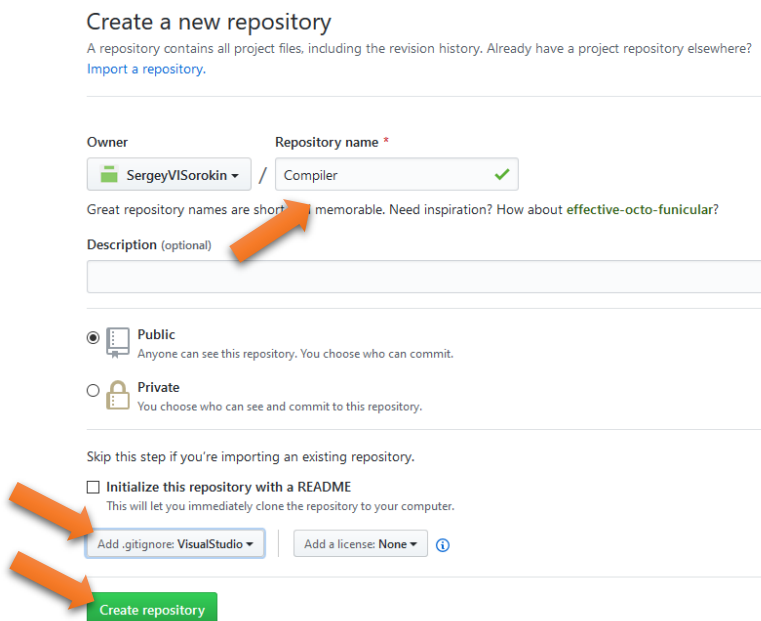


Рис. 2. Настройки создания нового проекта

3. Запомните ссылку на ваш репозиторий

Вы попадёте на страницу нового проекта, нажмите “Clone or download”, там будет написана ссылка (рис. 3). Запомните эту ссылку.

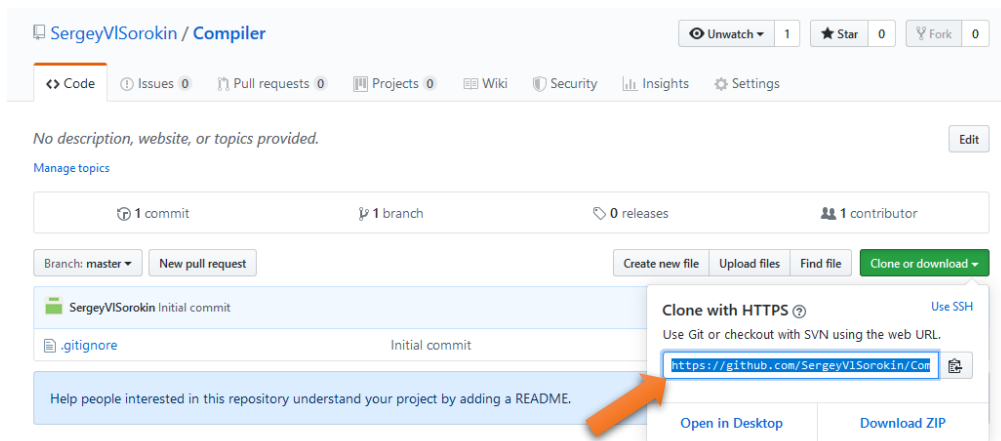


Рис. 3. Ссылка на репозиторий на Github

4. Клонирование репозитория в Visual Studio

Запустите Visual Studio. Выберите пункт «Clone or check out code» (рис. 4.).

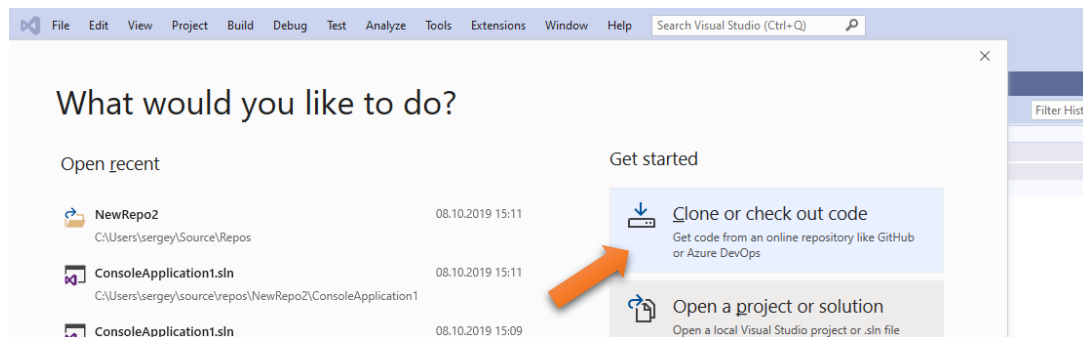


Рис. 4. Запуск клонирования репозитория в Visual Studio

Введите адрес репозитория, полученный на GitHub.com и путь, где проект будет находиться на локальном диске и нажмите Clone (рис. 5).

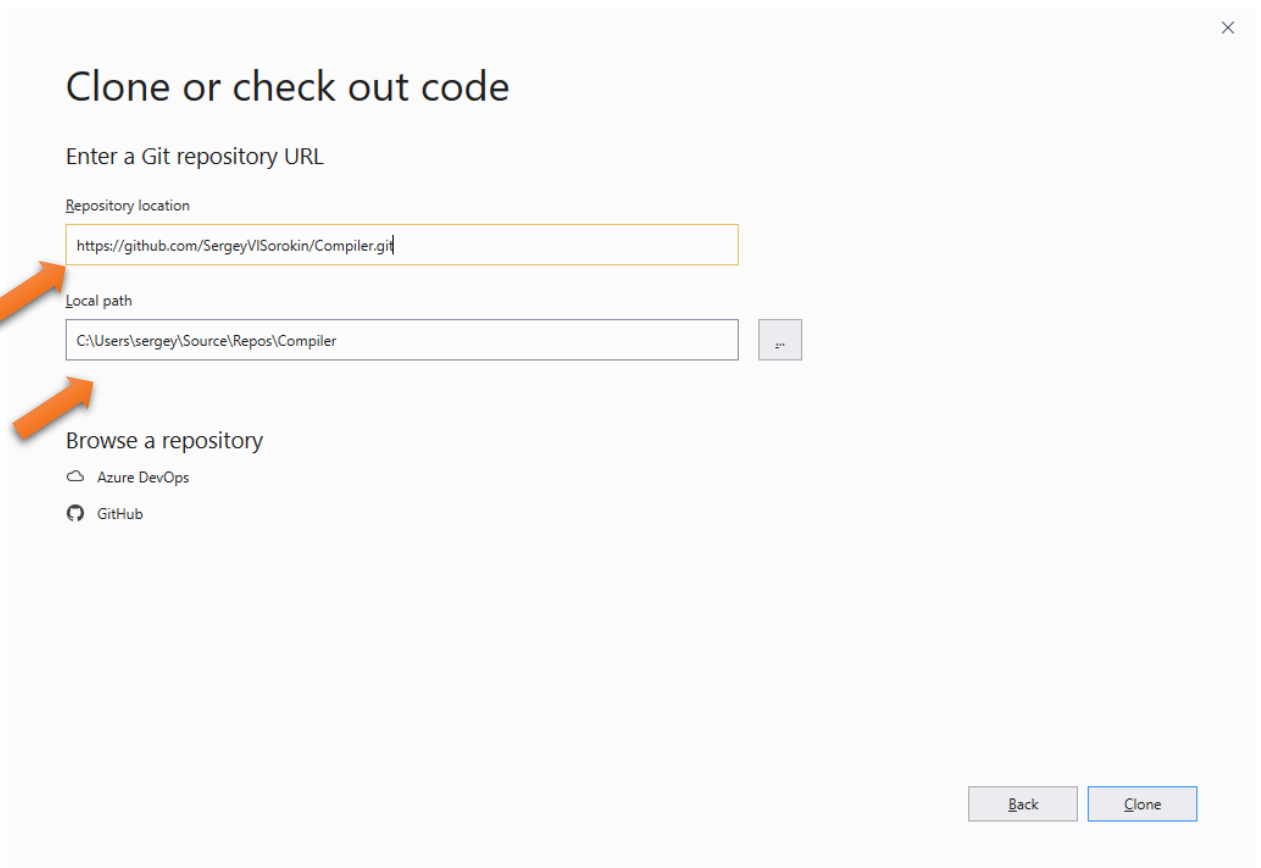


Рис. 5. Ввод размещения репозитория и его локальной копии

5. Создайте проект в локальном репозитории

Откройте созданный локальный репозиторий в Team Explorer в Visual Studio. Если повезёт, там будет открыт правильный репозиторий и будет ссылка «Create new project or solution in this repository», нажмите её.

Если там открыт какой-то другой репозиторий нажмите на иконку с зелёной вилкой «Manage Connections» и выберите созданный репозиторий в списке (рис. 6).

Если у вас открыт правильный репозиторий, но нету ссылки «Create new project ...» просто щёлкните «New...» в списке решений (рис. 6.)

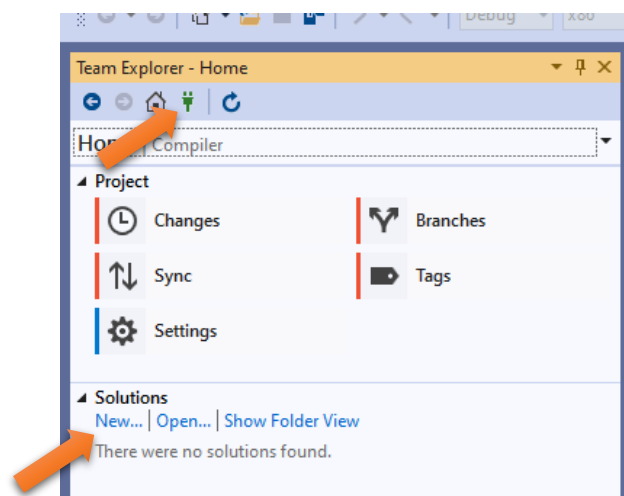


Рис. 6. Создание решения через Team Explorer

Выберите тип проекта, введите название проекта (например Scanner для лексического анализатора) и название решения (можно назвать Compiler для всего компилятора).

Не создавайте решение и проекты в обход Team Explorer, так как тогда необходимые файлы не будут добавлены под контроль версий, и их придётся добавлять руками.

6. Отправьте созданный проект на Github

В Team Explorer нажмите кнопку Changes. В текстовом поле ввода введите описание, что вы сделали (сейчас, например «Создан новый проект»), и нажмите стрелку справа от «Commit All». В меню выберите «Commit All and Push». Кнопка Commit не будет доступна, пока не введён текст (рис. 7).

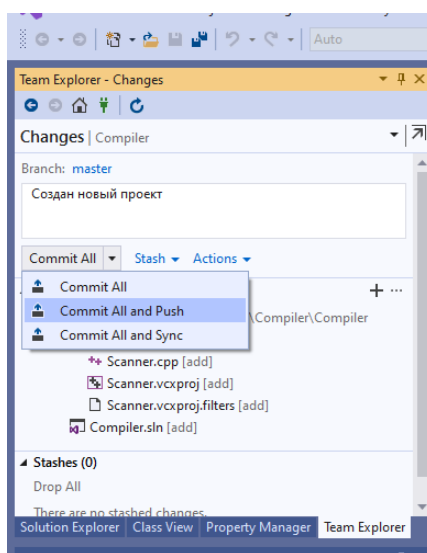


Рис. 7. Ввод описания коммита

Подключение к существующему проекту с другого компьютера

7. Клонировать проект

Всё совершенно аналогично пункту 4. Вы получите проект в том виде, в котором он есть на Github.

Работа в ветках

Для внесения изменений в проекте обычно создаются ветки, работа ведётся в них, и когда изменение завершено, оно переносится в основную ветку проекта, например master.

8. Создание ветки

Чтобы создать новую ветку зайдите в Team Explorer и нажмите «Branches». Нажмите «New Branch» и введите имя ветки (рис. 8). Нажмите «Create Branch».

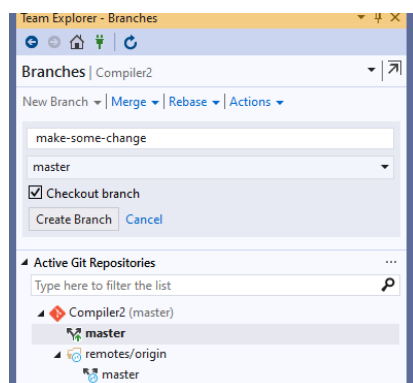


Рис. 8. Создание ветки

Созданная ветка будет показана в нижней части окна. Текущая ветка выделяется жирным шрифтом. Ветки, которые находятся в дереве remotes/origin находятся на сервере (github), которые снаружи – на Вашем компьютере. Переключаться между ветками можно выбрав команду Checkout, которая доступна в всплывающем меню по правой кнопке мыши на имени ветки.

9. Решите задачу

Напишите код, который делает что-то полезное.

10. Передайте свой код на сервер

Перейдите в Team Explorer, зайдите в раздел Changes, сделайте Commit All and Push.

Пояснение:

- Commit сохраняет изменения в локальной репозитории на вашем компьютере. Рекомендуется регулярно делать Commit'ы, тогда, если дальше Вы сделаете ошибку, можно будет всегда откатиться к предыдущему состоянию, сохранённому через Commit. Или, например, переключится на другую ветку, поработать в ней и вернуться.
- Push передаёт изменения на удалённый сервер, с которого Вы исходно взяли свой код.

11. Скачайте изменения других авторов

Зайдите в раздел Branches в Team Explorer. Переключитесь на ветку, изменения которой хотите получить или просто нажмите на ней правую кнопку мыши и выберите Pull.

12. Объедините свою работу с работой других

Зайдите в раздел Branches в Team Explorer.

Нажмите «Merge» и выберите изменения из какой и в какую ветку надо перенести. Например, если Вы сделали какую-то работу в ветке make-some-change, то её нужно выбрать в списке «... from ...», а в «Into ...» скорее всего нужен master (рис. 9).

Нажмите кнопку Merge.

Если изменения, произошедшие со времени создания вашей ветки в самом master'е и в ветке затрагивают разные строки, то объединение будет выполнено полностью автоматически.

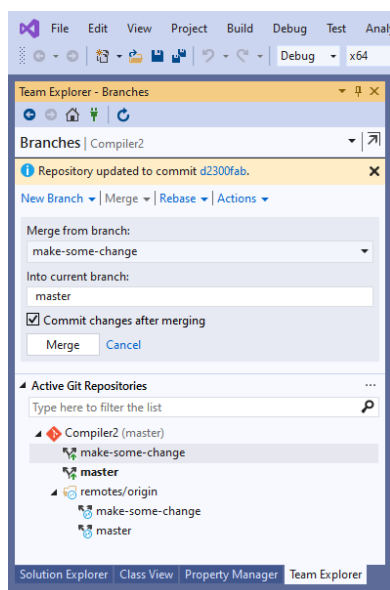


Рис. 9. Объединение веток

Если Git не сможет выполнить объединение, то Вы увидите сообщение о конфликте (рис. 10).

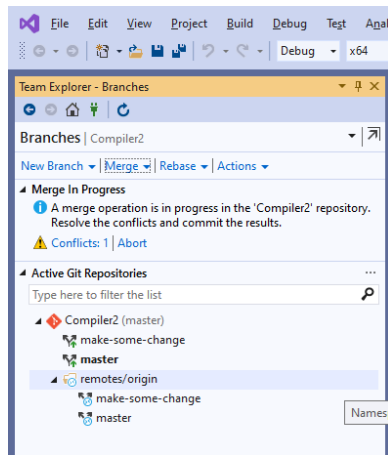


Рис. 10. Конфликт во время слияния

Щёлкните по ссылке «Conflicts..». Вы попадёте в список с местами конфликтов. Если щёлкнуть мышью по конкретному пункту списка, то там раскроется подменю с возможными действиями.

Чтобы разобраться с конфликтом нажмите «Merge» (рис. 11).

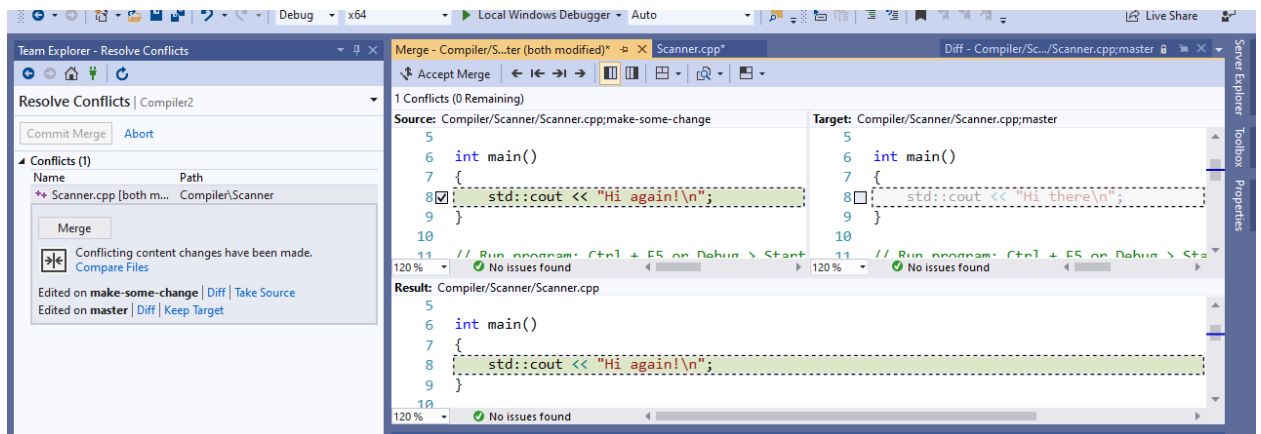


Рис. 11. Отображение конфликта слияния

Какой вариант попадёт в результат слияния можно выбрать «галками» слева от текста, в том числе можно поместить оба варианта один за другим. В случае необходимости можно внести изменения в конечный результат вручную в нижней части окна.

Чтобы принять финальный вариант нажмите «Accept merge».

После решения всех конфликтов обязательно нажмите «Commit Merge» чтобы зафиксировать результат. Чтобы сразу отправить его на сервер используйте «Commit and Push». Впрочем, перед отправкой на сервер не будет лишним проверить работоспособность программы.