



Vogsphere - Руководство

Президентская форма для сдачи заявок

zaz zaz@staff.42.fr

Реферат: В этом руководстве объяснено, как использовать **Vogsphere**, так же известную как **встроенную систему 42**, включая основы **git**. **Git** - это программное обеспечение для управления версиями, используемое в **Vogsphere**.

Глава I. Предисловие

НИКОГДА НЕ ЖДИТЕ ДО ПОСЛЕДНЕЙ МИНУТЫ чтобы клонировать свой репозиторий и начать его использовать. Таким образом, если что-то пойдет не так, у вас будет достаточно времени, чтобы решить эту проблему.

Пожалуйста, обратите внимание, что студенты, которые просят о помощи в отношении **Vogsphere за несколько часов до закрытия хранилищ увидят у себя ответ что-то вроде "Вы должны были прийти намного раньше!", и Они будут заслуживать такого ответа.**

I.1 Git

- Git — это распределенный менеджер управления версиями. Это инструмент, который мы используем для хранения версии изменений ваших файлов практически для каждого проекта в школе 42.
- Вы несете ответственность за приобретение знаний о **Git** и его использовании. Это руководство охватывает только очень ограниченный аспект этого **невероятного программного обеспечения**. Можно приобрести эти знания очень быстро, читая документацию **git** в Интернете. Это сэкономит вам много неприятностей.

I.2 Источники

Вот несколько ссылок, где вы можете найти хорошее качество документации в отношении **Git**:

- Официальная документация: git-scm.com
- Удивительная книга по Git [book_about_git.pdf](#)
- Знакомство с Git и GitHub: ч.1
- Знакомство с Git и GitHub: ч.2
- Git и Bash

Глава II. Операции в Vogsphere

С этого момента руководство считает, что вы прочитали документацию по **Git**.

II.1 В кратце

- При регистрации в проекте автоматически создается уникальный и персональный репозиторий для вашей команды и для этого конкретного проекта.
- Вы можете найти **URL-адрес** вашего репозитория в **интрасети** на странице вашего проекта.
- **URL репозитория** будет выглядеть примерно так:

```
vogsphere@vogsphere.42.fr:pisc-0-001/2013/paris-0-1/j00/yourlogin
```

- Далее, вы должны сделать свежую копию вашего репозитория с помощью команды **git clone**.
- Далее, вы будете работать над этой локальной копией. После этого, или всякий раз, когда вы хотите уведомить Git о своей работе, вы должны:
 - Добавить изменения с помощью команды **git add**.
 - Создать новую пометку этих изменений с помощью команды **git commit**.
- Вы можете проверить состояние локального репозитория в любое время с помощью команды **git status**.
- Совершая свою работу шаг за шагом, вы можете создавать столько изменений, сколько хотите. Прочитайте документацию, чтобы понять, почему это хорошая идея, чтобы сделать это ...
- Чтобы отправить изменения на сторону **Vogsphere** (т.е. в репозиторий, который будет использоваться для оценок), необходимо использовать команду **git push**. При первой же отправке в репозиторий не забудьте указать пункт назначения (**git push origin master**).
- После успешной отправки все ваши изменения будут существовать уже и в копии **Vogsphere** вашего репозитория, и будут видны во время оценки.
- Очевидно, что любые ИЗМЕНЕНИЯ, СОЗДАННЫЕ в вашей локальной копии уже ПОСЛЕ отправки прошлых изменений, не будут существовать в копии **Vogsphere** вашего репозитория пока вы их туда самостоятельно не отправите, это (к счастью) не автоматически. Было бы стыдно усердно работать над проектом и забывать о его продвижении, не так ли?
Последовательность команд:
 - **git add "file.c"**

- `git commit -m "fix"`
- `git push origin master`
- Если вы хотите в любое время проверить, то что было успешно сдано (то есть какие данные доступны в копии вашего репозитория [Vogsphere](#)), вы можете клонировать новую локальную копию вашего репозитория во временную папку. Этот клон будет точной копией того, что вы бы сдали, если бы оценка проводилась прямо сейчас.
- Когда вы работаете в команде (над некоторыми проектами или gush-ами), вы столкнетесь с проблемами синхронизации между локальной копией репозитория каждого члена команды. Существует несколько решений этой проблемы, которые подробно описаны в [документации Git ...](#)

II.2 Пример

Вот краткий пример работы с репозиторием с помощью [Vogspere](#) (очевидно, URL-адреса ваших репозиториев будут отличаться):

```

zaz@blackjack ~ $ cd work
zaz@blackjack work $ git clone vogsphere@vogsphere.42.fr:pisc-0-001/2013/paris-0-1/d00/demo d00
Cloning into 'd00'...
vogsphere: (INFO) This transaction has been started at 2013-06-17 16:19:35, server time.
vogsphere: (INFO) Rights will be determined using this time, so do NOT cut the connection.
vogsphere: (INFO) You have read and write rights on this repository
warning: You appear to have cloned an empty repository.
zaz@blackjack work $ cd d00
zaz@blackjack d00 $ vim test.c
zaz@blackjack d00 $ git add test.c
zaz@blackjack d00 $ git commit -m "Added test.c"
[master (root-commit) be2069b] Added test.c
 1 file changed, 4 insertions(+)
 create mode 100644 test.c
zaz@blackjack d00 $ vim test2.c
zaz@blackjack d00 $ git add test2.c
zaz@blackjack d00 $ git commit -m "Added test2.c"
[master f195334] Added test2.c
 1 file changed, 6 insertions(+)
 create mode 100644 test2.c
zaz@blackjack d00 $ git push origin master
vogsphere: (INFO) This transaction has been started at 2013-06-17 16:22:52, server time.
vogsphere: (INFO) Rights will be determined using this time, so do NOT cut the connection.
vogsphere: (INFO) You have read and write rights on this repository
Counting objects: 9, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 897 bytes, done.
Total 9 (delta 1), reused 0 (delta 0)
To vogsphere@vogsphere.42.fr:pisc-0-001/2013/paris-0-1/d00/demo
 * [new branch]      master -> master

```

```

zaz@blackjack d00 $ echo "/* New modification */" >> test2.c
zaz@blackjack d00 $ git commit -am "Committing a modification without git add, you fool"
[master ba0a82a] Committing a modification without git add, you fool
 1 file changed, 1 insertion(+)
zaz@blackjack d00 $ git push
vogsphere: (INFO) This transaction has been started at 2013-06-17 16:23:35, server time.
vogsphere: (INFO) Rights will be determined using this time, so do NOT cut the connection.
vogsphere: (INFO) You have read and write rights on this repository
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 351 bytes, done.
Total 3 (delta 1), reused 0 (delta 0)
To vogsphere@vogsphere.42.fr:pisc-0-001/2013/paris-0-1/d00/demo
 a6f17c3..ba0a82a  master -> master
zaz@blackjack d00 $

```

II.3 Пожалуйста, помните следующее

Vogsphere применяет строгую квоту в 100 МБ на репозиторий. Будьте очень осторожны, поскольку превышение квоты будет рассматриваться как злоупотребление ресурсами и будет наказано как таковое!

- **Vogsphere** использует билет **Kerberos** для вашей аутентификации. Билет приобретается автоматически при входе в сеанс. Однако через некоторое время этот БИЛЕТ ИСТЕЧЕТ. Если ваш билет **Kerberos** истек и вы не можете использовать **Vogsphere**, то вы можете получить новый билет с помощью команды **kinit**.
- Репозитории для сдачи будут закрыты в точную дату и время его сдачи, которое указано на странице вашего проекта во внутренней сети. У вас по-прежнему будет доступ для чтения к вашему репозиторию, но вы больше не сможете отправлять данные.

- Когда вы подключаетесь к **Vogsphere**, дата и время, используемые для проверки вашего доступа на запись к вашему репозиторию в **Vogsphere**, будут являться датой и временем вашего подключения к Vogsphere. Если слишком много студентов попытаются отправить данные в одно и то же время, ваш запрос будет поставлен в очередь (**Vogsphere** уведомит вас). Не волнуйтесь, если вы не разорвали соединение или сделали **git push** до истечения крайнего срока, ваше сообщение будет принято, если истекли дата и время подачи. Но в любом случае, кто может быть настолько глуп, чтобы сдать свою работу в последний момент ...