Генератор документации Sphinx Documentation

Выпуск 1.0

Дмитрий Мажарцев

мая 20, 2022

Оглавление

В руководстве подробно описан процесс генерации документации с помощью связки reStructuredText, Python Sphinx, GitHub и сервиса Read the Docs.

Примечание: *Последняя правка:* мая 20, 2022

ГЛАВА 1

Предисловие

В последнее время в русскоязычном сегменте интернета все реже и реже попадаются толковые статьи. Особенно это касается статей на техническую тему. Многие из них лишь поверхностно касаются озвученных вопросов, даже не предоставляя никаких ссылок на дополнительную информацию.

С целью хоть чуточку изменить ситуацию в лучшую сторону, я решил задокументировать свой опыт в виде монолитного руководства. Конечно, рассмотреть абсолютно все аспекты не представляется возможным. Поэтому такие места снабжены ссылками на другие источники.

Несмотря на благие намерения, данное руководство можно улучшить и дополнить. Критика, замечания и предложения приветствуются. Также буду рад любому сотрудничеству. Пишите мне на почту (см. *Обратная связь*) или отправляйте Pull Requests (см. *Pull Requests*).

1.1 О чём данное руководство?

Данное руководство посвящено процессу генерации документации с помощью связки reStructuredText, Python Sphinx, GitHub и сервиса Read the Docs. Ключевую роль в этой связке играет генератор документации Python Sphinx, разработанный для документирования языка программирования Python, но активно прижившегося и в других проектах.

Sphinx использует облегченный язык разметки reStructuredText, с помощью которого можно создавать текстовые документы с четкой структурой. В дальнейшем, такие документы могут быть преобразованы в любой другой формат: HTML, LaTeX, ODT, ePub, man и другие. Sphinx расширяет возможности reStructuredText и упрощает процесс генерации в различные форматы.

GitHub позволяет организовать совместную работу над документацией и значительно упрощает процесс отслеживания изменений. Сервис Read the Docs предоставляет бесплатную площадку для публикации документации, сгенерированной с помощью Sphinx. Он автоматизирует процесс создания и загрузки Sphinx-документации после каждой фиксации изменений на GitHub.

1.2 Почему именно Sphinx?

На данный момент я состою в русскоязычном сообществе LibreOffice и курирую работу над написанием и оформлением официальной русскоязычной документации по офисному пакету. У меня большой опыт работы со сложными документами. Я занимался версткой документов и с помощью обычных офисных пакетов, и с помощью LaTeX.

Офисные пакеты хорошо справляются со сложной документацией, они достаточно просты для обычных пользователей, но очень плохо приспособлены для совместной работы. Конечно, LibreOffice имеет встроенные функции контроля версий документов, записи изменений, имеется отличная функция «Составные документы», которая позволяет обеспечить одновременную совместную работу над сложными документами. Но все это не даёт той гибкости, которая есть у систем управления версиями файлов таких, как, например, Git.

LaTeX лишен многих недостатков офисных пакетов, обладает потрясающим качеством генерации документов, но имеет высокий входной порог. Последнее обстоятельство затрудняет совместную работу и во многом перекладывает значительную часть работы на плечи редактора.

Sphinx совмещает в себе качество LaTeX и простоту офисных пакетов. Для совместной работы достаточно иметь обыкновенный текстовый редактор, что освобождает от привязки к каким-либо платформам, избавляет от установки громоздких программ и значительно упрощает работу редактора по оформлению и генерации документации.

В связке Sphinx с GitHub и Read the Docs упрощается процесс отслеживания изменений в документации и её публикация.

1.3 Структура книги

В главе «*Генератор документации Sphinx*» описана установка и основные настройки Sphinx.

В главе «*Стандартный синтаксис разметки reStructuredText*» приведен стандартный синтаксис разметки reStructuredText. Глава «*Конструкции разметки Sphinx*» содержит описание дополнительных конструкции разметки, вносимых Sphinx. Данные конструкции значительно расширяют функционал reStructuredText, но не поддерживаются стандартной разметкой и поэтому не могут быть применены вне Sphinx.

Глава «*Система управления версиями Git*» является справочником по основным командам git.

Глава «*GitHub и Read The Docs*» рассматривает веб-сервис для хостинга IT-проектов GitHub, сервис Read The Docs и их взаимодействие.

В главах «*Известные проблемы*» и «*Часто задаваемые вопросы*» описаны основные проблемы, с которыми я столкнулся.

1.4 Рекомендации

Отдельным авторам и переводчикам документации для работы не нужно иметь установленный Sphinx, а , следовательно, нет необходимости читать многие разделы данного руководства. Им достаточно прочитать раздел *Стандартный синтаксис разметки reStructuredText* и *Конструкции разметки Sphinx*, чтобы изучить необходимый синтаксис. А также ознакомиться с разделом *Работа с GitHub*. Этого минимума хватит для комфортной работы как самих авторов/переводчиков, так и редакторов.

Редакторы, для облегчения своей работы, должны дать авторам и переводчикам рекомендации относительно оформления исходного текста, создания закладок и указателей. В качестве примера смотрите раздел *Соглашения*.

1.5 Авторские права

1.6 Дата публикации и версия программного обеспечения

Опубликовано 29 декабря 2014 года. При написании руководства были использованы:

Программа	Версия
Sphinx	1.2.2
Git	2.1.0

1.7 Обратная связь

Автор Дмитрий Мажарцев

Контакты deemonizer@gmail.com

Блог http://librerussia.blogspot.ru

Адрес Волгоград

Дата 29 декабря 2014 года

[1] http://creativecommons.org/licenses/by-sa/4.0/deed.ru.

ГЛАВА 2

Генератор документации Sphinx

В главе рассмотрена установка и дополнительные настройки Sphinx, позволяющие задать или изменить параметры генерации документации.

2.1 Быстрый старт

Sphinx доступен для всех основных операционных систем, которые поддерживает язык программирования Python.

2.1.1 Установка

С помощью pip:

```
pip install sphinx
```

С помощью easy_install:

```
easy_install sphinx
```

В стандартном репозитории Ubuntu 14.10 есть пакеты `python-sphinx` и `python3-sphinx`.

```
sudo apt-get install python3-sphinx
```

Другие способы установки описаны в разделе Installing Sphinx официальной документации Sphinx.

2.1.2 Создание нового проекта

Создадим директорию для нового проекта и перейдем в неё. Для этого в ОС Linux необходимо выполнить следующие команды в терминале:

```
mkdir MyProject
cd MyProject/
```

Для инициализации проекта необходимо выполнить команду `sphinx-quickstart` :

```
sphinx-quickstart
```

Программа задаст ряд вопросов. Все настройки можно будет позже изменить в файле `conf.py`.

```
> Сделать ли раздельные папки исходников и готовых страниц - Да
> Separate source and build directories (y/N) [n]: y

> Название проекта. Для начала лучше вводить на латинице.
> Project name:

> Имя автора/авторов. Для начала лучше вводить на латинице.
> Author name(s):

> Номер релиза проекта
> Project release []:

> Язык проекта
> Project language [en]: ru
```

После ответа на вопросы будут созданы файлы `index.rst`, `conf.py`, `Makefile`, `_build`, `_static`, `_templates`.

```
.
 Makefile
 _build
 _templates
 _static
 conf.py
 index.rst
```

Makefile — содержит инструкции для генерации результирующего документа командой make.

_build — директория, в которую будут помещены файлы в определенном формате после того, как будет запущен процесс их генерации.

_static — в эту директорию помещаются все файлы, не являющиеся исходным кодом (например, изображения). Позже создаются связи этих файлов с директорией build.

conf.py — содержит конфигурационные параметры Sphinx, включая те, которые были выбраны при запуске sphinx-quickstart в окне терминала.

index.rst — это корень проекта. Он соединяет документацию воедино, если она разделена на несколько файлов[1] .

[1] IBM developerWorks Россия: Простое и удобное создание документации в Sphinx

2.1.3 Файл index

В каталоге проекта находится файл `index.rst`, который служит для объединения всех файлов в один проект. `index.rst` имеет достаточно простую структуру. Если открыть `index.rst` в простом текстовом редакторе, то будет отображено примерно следующее содержание:

```
.. 3 documentation master file, created by
sphinx-quickstart on Fri Dec 26 19:44:30 2014.
You can adapt this file completely to your liking, but it should at least
contain the root `toctree` directive.

 Welcome to 3's documentation!
 ==============================

 Contents:

 .. toctree::
    :maxdepth: 2



 Indices and tables
 ==================

 * :ref:`genindex`
 * :ref:`modindex`
 * :ref:`search`
```

Первый абзац, который начинается с двух точек, содержит комментарий. Две идущие подряд точки `..` перед абзацем означают комментарий. Также они служат объявлением многих команд.

Подробнее синтаксис разметки рассматривается в следующих главах, а пока рассмотрим директиву `.. toctree::`. Данная директива объединяет отдельные файлы в единый проект.

Содержимое `index.rst` не должно включать много информации и в нём обязательно должна присутствовать директива `.. toctree::`.

Чтобы включить в проект другие файлы, необходимо прописать названия этих файлов в `.. toctree::`. Для примера создадим в корне проекта файл `example1.rst` и `example2.rst` со следующим содержанием:

```
Это пример
==========
```

Теперь включим их в проект в файле `index.rst`, добавив названия файлов к директиве `.. toctree::`.

```
Оглавление:

.. toctree::
   :maxdepth: 2

   example1
   example2
```

Обратите внимание, что название файла пишется без расширения. Также важен отступ и пустая строка. Подробнее директива `.. toctree::` рассматривается в разделе *Автоматическое содержание* главы

Конструкции разметки Sphinx.

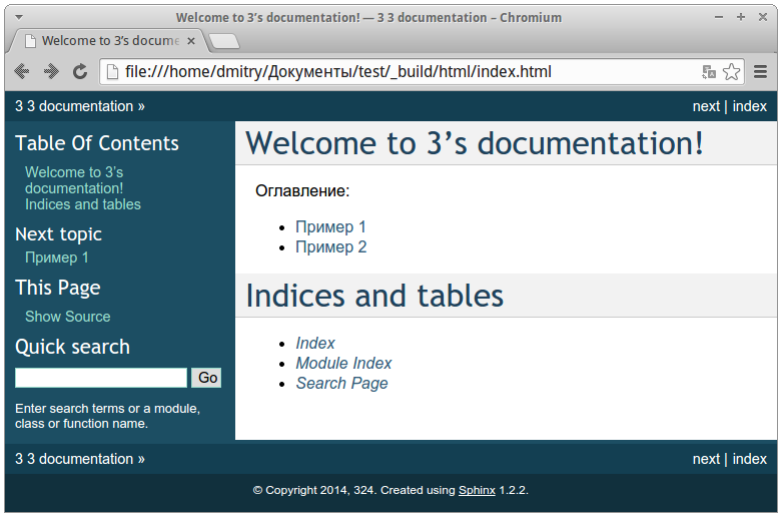
Сохраним `index.rst` и произведем генерацию документа в формат HTML.

2.1.4 Генерация документа

Для генерации документации в HTML формат необходимо выполнить в командной строке команду `make html`. Аналогичным образом можно выполнить генерацию в другие форматы, например, `make epub`.

```
cd MyProject/
make html
```

Произойдет сборка HTML, выходные файлы будут помещены в директорию `_build/html/`. Перейдем в неё и откроем файл `index.html` в браузере.



Получив совсем немного исходных данных, Sphinx сумел создать нечто большее. Мы получили несложную компоновку, содержащую информацию о документации проекта, раздел поиска, содержание, заметки об авторских правах, включая имя и дату, а также нумерацию страниц.

Обратите внимание на раздел поиска: Sphinx проиндексировал все файлы и с помощью JavaScript создал статический сайт, на котором можно искать нужную информацию.

На снимке показана стандартная тема оформления документации. Она может быть изменена. Также можно настроить локализацию и прочие параметры.

Файл конфигурации `conf.py` позволяет настроить много дополнительных параметров генерации документации.

2.1.5 Добавление иллюстраций

Для добавления изображений в документы, необходимо предварительно поместить изображения в папку `_static`. В файл изображения добавляются директивой `.. image::` или `.. figure::`:

```
.. image:: _static/favicon.png
```

Подробнее смотрите раздел *Изображения и иллюстрации*.

2.1.6 Автоматическая сборка

Для автоматической сборки документации используйте sphinx-autobuild.

Для установки выполните команду:

```
pip install https://github.com/lepture/python-livereload/archive/master.zip
```

Или:

```
pip install sphinx-autobuild
```

Для запуска автоматической сборки выполните команду:

```
sphinx-autobuild <папка-с-проектом-Sphinx> <папка-с-проектом-Sphinx>/_build/html
```

В первом параметре указывается директория с исходными rst-файлами, во втором указывается папка с выходными данными.

В моем случае команда выглядела следующим образом:

```
sphinx-autobuild docs docs/_build/html
```

Посмотреть сгенерированную документацию можно открыв в браузере адрес `http://127.0.0.1:8000`.

Для интеграции в Makefile добавьте цель:

```
livehtml:
sphinx-autobuild -b html $(ALLSPHINXOPTS) $(BUILDDIR)/html
```

Подробнее смотрите официальную документацию sphinx-autobuild.

2.2 Файл конфигурации

Все настройки сборки документации находятся в файле `conf.py` в корне проекта Sphinx.

2.3 Общие настройки

Общие настройки задаются в разделе `General configuration` файла `conf.py`.

2.3.1 Изменение названия и копирайта

```
# General information about the project.
project = 'Sphinx-ru'
copyright = '2014, Dmitry Mazhartsev'
```

2.3.2 Строки Unicode

Использование кириллических символов в названиях проекта и других строках, может приводить к ошибкам генерации. В версии Sphinx для Python 3 таких проблем не наблюдается. В версии для Python 2.7 перед каждой кириллической строкой необходимо ставить `u`.

```
project = u'Мой проект'
copyright = u'2014, Дмитрий Мажарцев'
```

2.3.3 Версии публикации

Изменить параметры `version` и `release`:

```
# The short X.Y version.
version = '1'
# The full version, including alpha/beta/rc tags.
release = '1'
```

2.3.4 Настройка локализации

Многие шаблоны тем имеют несколько локализаций, в зависимости от этого некоторые элементы оформления могут быть автоматически локализованы, например, заголовки блоков предупреждений меняются в зависимости от языка.

Настройки языка задаются в разделе `General configuration` файла `conf.py` в строке `language`:

```
language = 'ru'
```

Для корректной генерации ePub в разделе `Options for Epub output` файла `conf.py` есть строка `epub_language` :

```
epub_language = 'ru'
```

2.3.5 Настройка отображения даты

Формат отображения даты зависит от выбранного языка (см. *Настройка локализации*). Но можно задать и собственный формат.

```
today = ''
today_fmt = '%B %d, %Y'
```

Из этих настроек берется информация для автозамены `|today|` (см. *Автозамены Sphinx (Подстановки)*).

2.3.6 Подключение расширений

Список расширений для Sphinx приведен на странице Sphinx Extensions официальной документации.

Подключаемые расширения прописываются в файле `conf.py`, в строке `extensions`.

```
extensions = [
'sphinx.ext.mathjax',
'sphinx.ext.graphviz',
]
```

2.3.7 Режим отображения формул

Формулы в Sphinx могут отображаться с помощью одного из двух расширений: `sphinx.ext.imgmath` или `sphinx.ext.mathjax`. Первое расширение отображает формулы как изображения в png формате. Другое расширение использует JavaScript библиотеку `mathjax` для отображения формул.

Я предпочитаю использовать `sphinx.ext.mathjax`, так как оно позволяет масштабировать формулы без потери качества отображения. Но для сервиса Read The Docs нужно использовать `sphinx.ext.imgmath`.

Указать Sphinx какое расширение использовать можно в конфигурационном файле `conf.py`, в строке `extensions`.

```
 extensions = [
 'sphinx.ext.mathjax',
]

 или

 extensions = [
 'sphinx.ext.imgmath',
]
```

Предупреждение: Использовать одновременно оба расширения нельзя.

2.4 Генерация в формат HTML

Настройки генерации в формат HTML задаются в разделе `Options for HTML output` файла `conf.py`.

2.4.1 Добавление favicon

В разделе `Options for HTML output` раскомментировать строку и прописать путь к файлу favicon:

```
html_favicon = '_static/favicon.ico'
```

2.4.2 Метаданные. Тег META

Имеется возможность добавлять метаданные каждой из страниц. Это не относится к настройкам файла `conf.py`, а добавляется непосредственно в rst файлы с помощью директивы `.. meta::`. Подробнее смотрите раздел *Метаданные. Тег META*.

2.4.3 Смена HTML-темы

В строке `html_theme` указать название используемой темы:

```
html_theme = 'sphinx_rtd_theme'
```

Раскомментировать строку *html_ theme_ path* и прописать в ней путь к HTML-теме:

```
html_theme_path = ['.']
```

В Sphinx есть ряд встроенных тем. Для их подключения достаточно написать название темы в строке `html_theme`, путь указывать не надо. Названия стандартных тем:

- default
- sphinxdoc
- scrolls
- agogo
- traditional
- nature
- haiku
- pyramid

Некоторые из перечисленных тем поддерживают дополнительные настройки. Подробнее смотрите раздел HTML theming support официальной документации Sphinx.

Примечание: В Read The Docs при значении `html_theme = 'default'` используется тема Read The Docs.

Некоторые сторонние темы:

- Read the Docs Theme

- Alabaster
- Sphinx Bootstrap Theme

2.5 Генерация в формат LaTeX

Настройки генерации в формат LaTeX задаются в разделе `Options for LaTeX output` файла `conf.py`.

2.5.1 Преамбула

В файле `conf.py` можно указать дополнительные параметры преамбулы:

```
# Additional stuff for the LaTeX preamble.
'preamble': '\\usepackage[utf8]{inputenc}',
'babel': '\\usepackage[russian]{babel}',
'cmappkg': '\\usepackage{cmap}',
'fontenc': '\usepackage[T1,T2A]{fontenc}',
'utf8extra':'\\DeclareUnicodeCharacter{00A0}{\\nobreakspace}',
}
```

2.5.2 Язык и кодировка

Для корректной генерации в формат LaTeX кириллических документов необходимо указать в файле `conf.py` дополнительные параметры преамбулы, смотрите пример в разделе *Преамбула*.

2.5.3 Уровни заголовков в содержании

Параметр `:maxdepth:` не распространяется на LaTeX-документы. Глубина оглавления в LaTeX контролируется его внутренним счетчиком, который можно настроить в файле конфигурации Sphinx `conf.py`, указав в преамбуле значение `\setcounter{tocdepth}{2}`.

2.6 Генерация в формат ePub

Для генерации в формат ePub используются настройки `conf.py` в разделе `Options for Epub output`. Также частично используются настройки из раздела генерации в HTML `Options for HTML output`.

2.6.1 Настройка языка

Для корректной генерации ePub в разделе `Options for Epub output` файла `conf.py` есть строка `epub_language` :

```
epub_language = 'ru
```

2.6.2 Отключение копирайта

В разделе `Options for HTML output` раскомментировать и установить значение `False` строке `html_show_copyright`

```
# If true, "(C) Copyright ..." is shown in the HTML footer.
# Default is True.

html_show_copyright = False
```

2.6.3 Отключение надписи «Created using Sphinx»

В разделе `Options for HTML output` раскомментировать и установить значение `False` строке `html_show_sphinx`

```
# If true, "Created using Sphinx" is shown in the HTML footer.
# Default is True.

html_show_sphinx = False
```

2.6.4 Настройка отображения URL-адресов

URL-адреса могут отображаться в ePub в нескольких режимах: в строке(inline), внизу страницы(footnote) и быть оформлены в виде гиперссылок(no). Для настройки отображения необходимо раскомментировать строку `epub_show_urls` и установить нужное значение:

```
# How to display URL addresses: 'footnote', 'no', or 'inline'.
epub_show_urls = 'inline'
```

2.6.5 Настройка глубины содержания

Раскомментировать и установить нужное значение. По умолчанию в оглавление входят заголовки глубиной до 3-х уровней:

```
# The depth of the table of contents in toc.ncx.
epub_tocdepth = 3
```

2.6.6 Настройка названия и заголовка

```
# Bibliographic Dublin Core info.
epub_title = 'Sphinx-ru'
epub_author = 'Dmitry Mazhartsev'
epub_publisher = 'Dmitry Mazhartsev'
epub_copyright = '2014, Dmitry Mazhartsev'
```

Для решения проблем с кириллическими строками смотрите раздел *Строки Unicode*.

ГЛАВА 3

Стандартный синтаксис разметки reStructuredText

В главе приведен стандартный синтаксис разметки reStructuredText. Python Sphinx значительно расширяет возможности reStructuredText и вносит дополнительные конструкции. Так как данная глава может быть полезна всем пользователям reStructuredText, конструкции работающие только в Sphinx вынесены в следующую главу.

3.1 Что такое reStructuredText?

reStructuredText (сокращение: **ReST**, расширение файла: **.rst**) — облегчённый язык разметки, который может быть преобразован в различные форматы — HTML, ePub, PDF и другие.

Документы с разметкой ReST являются обычными текстовыми файлами. С такими файлами очень легко работать посредством системы управления Git, которая позволяет отслеживать все вносимые изменения, легко принимать или отклонять их.

Помимо этого, документы в формате .rst можно открывать и редактировать в любом простом текстовом редакторе (например, в Блокноте). Это позволяет работать над документацией в любых условиях, на любых платформах, без необходимости использовать специализированное программное обеспечение.

Самое главное, что ReST позволяет сосредоточиться исключительно на структуре документа и не отвлекаться на его оформление.

ReST аналогичен языку разметки Markdown, но обладает более расширенным синтаксисом, особенно вкупе с генератором документации Sphinx. ReST используется во многих проектах, например, на сайте GitHub. Также его используют многие генераторы статических сайтов такие, как: Hyde, Pelican и другие.

3.2 Редакторы reStructuredText

Документы с разметкой ReST можно создавать в любом обычном текстовом редакторе, но существует ряд специализированных редакторов ReST с возможностью предпросмотра и другими удобными функциями.

3.2.1 ReText

ReText (https://github.com/retext-project/retext) — редактор Markdown и reStructuredText для Linux. Есть возможность установки ReText в ОС Windows, инструкция находится здесь. Для Mac OS X репозиторий Homebrew: https://github.com/samueljohn/homebrew-python

Основные возможности редактора:

- Полная поддержка Markdown и reStructuredText, а также расширений Python-Markdown;
- Экспорт в HTML, PDF, ODT из коробки, а также возможность создавать свои собственные экспортные расширения (например, есть расширение для загрузки в Google Drive);
- Поддержка вкладок;
- Поддержка CSS-стилей и подсветка синтаксиса;
- Проверка орфографии (в том числе и для русского языка);
- Два движка просмотра: основанный на QTextBrowser и основанный на WebKit.
- Поддержка математических формул (с синтаксисом LaTeX).

Предупреждение: ReText не распознает конструкции, специфичные для Sphinx.

Примечание: Данное руководство написано с помощью ReText.

3.2.2 SublimeText

SublimeText (https://www.sublimetext.com/) — универсальный редактор большего количества форматов как для Linux так и для Windows. По умолчанию не поддерживает reStructuredText. Но есть специальные плагины:

RestructuredText Improved (https://packagecontrol.io/packages/RestructuredText%20Improved) плагин добавляющий в SublimeText подсветку синтаксиса, а так же возможность перехода по заголовками используя комбинацию `Control-R`

Restructured Text (RST) Snippets (https://github.com/mgaitan/sublime-rst-completion) плагин добавляющий в SublimeText сниппеты основных конструкций, возможность геренерации Html/Pdf/Docx, и самое главное это возможность удобной работы с таблицами

GitGutter или **Modific** (https://github.com/jisaacks/GitGutter, https://github.com/gornostal/Modific) Данные плагины подсвечивают строки измененные последним коммитом, другими словами diff tools в режиме реального времени.

3.2.3 Online reStructuredText editor

Online reStructuredText editor (http://rst.ninjs.org) — простой онлайн-редактор reStructuredText.

3.2.4 NoTex.ch

NoTex.ch (https://www.notex.ch/) — онлайн-редактор с поддержкой reStructuredText, Markdown, LaTex и др. Поддерживает экспорт в PDF, HTML, EPUB, txt и LaTex.

3.2.5 rstext.me

rstext.me (http://rstext.me/) — онлайн-редактор reStructuredText с возможностью экспорта в разные форматы.

3.3 Синтаксис reStructuredText

3.3.1 Базовые концепции

Синтаксис reStructuredText опирается на следующие концепции:

- Отступы и пробелы имеют значение для команд разметки[1], но не имеют значения внутри текста (10 пробелов будут отображены как один);
- В командах (директивах) используется символ обратной кавычки «`», который располагается на клавише с буквой ё и символом ~. Использование обычных одинарных кавычек в командах не приведет к желаемым результатам.

[1] Не важно как делается отступ — пробелами или клавишей Tab, главное, чтобы они были одинакового размера.

3.3.2 Абзацы

Абзацы в ReST (reStructuredText) отделяются друг от друга пустой строкой:

```
Первый абзац...

Строки параграфов начинаются от левой границы
и отделяются параграфы друг от друга пустой строкой.
```

3.3.3 Заголовки

ReST поддерживает несколько уровней заголовков. Заголовки первого уровня (главы) подчеркиваются символом равно =. Заголовки второго уровня (подглавы) подчеркиваются символом короткого тире или минуса -. Заголовки третьего уровня (подпункта) подчеркиваются символом тильды ~. Для параграфов допускается использовать подчеркивание символами двойных кавычек "

Заголовки подчеркиваются (или отбиваются сверху и снизу) с помощью небуквенных и нецифровых 7-битных ASCII символов. Рекомендуется использовать: «= ` : ' " ~ ^ _ * + # < >». Отбивка должна быть не короче текста заголовка.

```
Заголовок 1 уровня
==================

Заголовок 2 уровня
------------------

Заголовок 3 уровня
~~~~~~~~~~~~~~~~~~

Заголовок 4 уровня
""""""""""""""""""""
```

3.3.4 Начертание

Чтобы выделить текст **жирным** начертанием или *курсивным* используется обособление звездочками:

```
**жирный текст**

*курсив текст*
```

Внимание: Не допускается наличие пробелов между выделяемым словом и звездочкой, например, команда `** жирный текст**` не даст нужного эффекта.

Начертание текста `«как есть»` достигается обособлением двумя обратными кавычками:

```
``«как есть»``
```

3.3.5 Нумерованные списки

Нумерованные списки создаются с помощью символа решетки с точкой `#.`:

```
#. Один
#. Два
#. Три

Или:
5. Пять
6. Шесть
#. Семь
```

Результат:

1. Один
2. Два
3. Три

Или:

5. Пять
6. Шесть
7. Семь

3.3.6 Маркированные списки

Маркированные списки создаются с помощью символа звездочки `*` или дефиса `-`. Пробелы после маркера обязательны:

```
* Один
* Два
* Три
```

Результат:

- Один
- Два
- Три

3.3.7 Вложенные списки

```
* Первый уровень
    * Второй уровень
        * Третий уровень
```

Результат:

- **Первый уровень**
    - **Второй уровень**
        * Третий уровень

```
#. Один
    * Маркер
#. Два
    #. Номер
```

Результат:

1. **Один**

    - Маркер

2. **Два**

    1. Номер

3.3.8 Верхний и нижние индексы

Верхние и нижние индексы добавляются с помощью команд `:sub:` и `:sup:`.

```
H\ :sub:`2`\ O
E = mc\ :sup:`2`
```

Результат:

- H_2O
- $E = mc^2$

Другой способ с помощью автозамены:

```
Химическая формула воды - |H2O|.

.. |H2O| replace:: H\ :sub:`2`\ O
```

Химическая формула воды — H_2O.

3.3.9 Определения

В ReST можно набрать два типа определений:

```
:Первый: В прямоугольном треугольнике квадрат длины
         гипотенузы равен сумме квадратов длин катетов.

Второй
    В прямоугольном треугольнике квадрат длины
    гипотенузы равен сумме квадратов длин катетов.
```

Результат:

Первый В прямоугольном треугольнике квадрат длины гипотенузы равен сумме квадратов длин катетов.

Второй В прямоугольном треугольнике квадрат длины гипотенузы равен сумме квадратов длин катетов.

3.3.10 Цитаты

Для вставки цитат используется отступ, сделанный с помощью клавиши *Tab*:

```
Основной текст:

    Цитата неизвестного человека

    --Аноним
```

Результат:

Основной текст:

> Цитата неизвестного человека
>
> —Аноним

3.3.11 Эпиграф

```
.. epigraph::

   *«Если бы двери восприятия были чисты, всё
   предстало бы человеку таким, как оно есть - бесконечным»*

   -- Уильям Блэйк
```

Результат:

> *«Если бы двери восприятия были чисты, всё предстало бы человеку таким, как оно есть — бесконечным»*
>
> — Уильям Блэйк

Оформление эпиграфа зависит от настроек HTML-темы или используемого шаблона LaTeX.

В американской типографике, в отличие от европейской, не принято отбивать тире пробелами. Чтобы получить пробел между тире и автором я использовал функцию *Автозамены (Подстановки)*. В моем случае код эпиграфа выглядит так:

```
.. epigraph::

   *«Если бы двери восприятия были чисты, всё
   предстало бы человеку таким, как оно есть - бесконечным»*

   -- |nbsp| Уильям Блэйк

   .. |nbsp| unicode:: U+00A0
```

3.3.12 Сноски

Сноски могут быть разного вида:

```
Числовая сноска [5]_.

.. [5] Сюда ведет числовая сноска.

Сноски с автоматической [#]_ нумерацией [#]_.

.. [#] Это первая сноска.
.. [#] Это вторая сноска.

Автосимвол сносок используйте вот так [*]_ и [*]_.

.. [*] Это первый символ.
.. [*] Это второй символ.
```

Результаты:

Числовая сноска[5].

Сноски с автоматической[2] нумерацией[3].

Автосимвол сносок используйте вот так*[0] и†[0].

```
Ссылки на цитаты выглядят так [CIT2002]_.

.. [CIT2002] Это цитата
(как часто используемая в журналах).
```

Ссылки на цитаты выглядят так [CIT2002].

При экспорте в PDF сноски автоматически располагаются в конце страницы. Чтобы цитата располагалась в конце HTML-страницы, необходимо команду сноски располагать в конце *.rst* файла [CIT2003].

3.3.13 Комментарии

В ReST можно оставлять комментарии, которые отображаются только в исходном файле ReST. Комментарии создаются с помощью двух точек в начале предложения ... Для создания многострочных комментариев необходимо соблюдать отступ:

```
.. Это комментарий
   Многострочный комментарий
```

[5] Сюда ведет числовая сноска.
[2] Это первая сноска.
[3] Это вторая сноска.
[0] Это первый символ.
[0] Это второй символ.

3.3.14 Листинги (исходный код)

Если обособление обратными кавычками используется для визуального выделения команд в абзацах, то для примеров частей исходного кода используется команда из двух двоеточий `::`:

```
Посмотрим на исходный код:
::

    Пример исходного кода
```

Предупреждение: Пустая строка между командой `::` и примером кода, а также отступ перед ним, обязательны.

Существуют другие способы ввода команды `::`, например:

```
Посмотрим на исходный код: ::

    Пример исходного кода
```

Или так:

```
Посмотрим на исходный код::

    Пример исходного кода
```

В данном случае команда `::` будет верно истолкована, а двоеточие в тексте поставлено автоматически. Это более лаконичная форма записи.

Для вставки блоков исходного кода с подсветкой синтаксиса и нумерацией строк в Sphinx используются специальные команды, подробнее смотрите раздел *Примеры исходного кода с подсветкой синтаксиса*.

3.3.15 Автозамены (Подстановки)

Язык *reStructuredText* — очень гибкий язык разметки, который поддерживает функцию автозамены (подстановки).

```
Язык |ReST| - очень гибкий язык разметки (подстановки).

.. |ReST| replace:: *reStructuredText*
```

Для удобства я в начале каждого файла делаю список автозамен.

3.3.16 Использование символов юникод (unicode)

С функцией автозамены связана функция вставки символов unicode:

```
Copyright |copy| 2015, |LibreRussia (TM)| |---| все права защищены.

.. |copy| unicode:: 0xA9 .. знак копирайта
.. |LibreRussia (TM)| unicode:: LibreRussia U+2122 .. символ торговой марки
.. |---| unicode:: U+02014 .. длинное тире
```

Получится такой результат:

Copyright © 2015, LibreRussia™ — все права защищены.

3.3.17 Дата и время

```
.. |date| date:: %d.%m.%Y
.. |time| date:: %H:%M

Текущая дата |date| и время |time|
```

Результат: Текущая дата 20.05.2022 и время 06:26 (на момент генерации документа).

Sphinx добавляет дополнительные команды автозамены, которые не требуют объявления. Подробнее о них написано в следующей главе.

3.3.18 Вставка текста из других файлов

ReST позволяет вставлять текст из других файлов:

```
.. include:: имя_файла
```

3.3.19 Черта (Линия)

Иногда возникает необходимость визуально отделить абзац, для этого можно воспользоваться чертой, достаточно поставить подряд несколько дефисов (не меньше 4-х), также можно воспользоваться нижним подчеркиванием:

```
--------

--------
```

Предупреждение: Символы черты должны быть отбиты пустыми строками до и после.

Предупреждение: Черта не должна завершать документ. Черта, расположенная в самом конце документа может вызывать ошибки при сборке.

3.3.20 Ссылки

Внешние ссылки создаются так:

```
1. Внешние ссылки выглядят так: ссылка_.

.. _ссылка: http://librerussia.blogspot.ru/

2. Если несколько слов, тогда так: `ссылка в несколько слов`_.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
.. _`ссылка в несколько слов`: http://librerussia.blogspot.ru/

3. `Более компактная запись ссылок <http://librerussia.blogspot.ru/>`_
```

Результат:

1. Внешние ссылки выглядят так: ссылка.
2. Если несколько слов, тогда так: ссылка в несколько слов.
3. Более компактная запись ссылок

Внутренние ссылки делаются так:

```
Внутренние ссылки делаются так_

.. _так:
```

Ссылками также являются и заголовки разделов, например, *Таблицы* :

```
Ссылка на раздел создается так `Таблицы`_ .
Достаточно в обратных кавычках написать название заголовка.
```

Sphinx расширяет возможности создания ссылок, в том числе позволяет ссылаться на заголовки в других документах. Подробнее читайте раздел *Перекрестные ссылки*.

3.3.21 Изображения и иллюстрации

Вставка изображения между слов осуществляться с помощью функции автозамены:

```
Вставка изображения между слов |кубик-рубика| осуществляться с помощью функции␣
↪автозамены:

.. |кубик-рубика| image:: _static/favicon.ico
```

Вставка изображений между абзацами осуществляется так:

```
.. figure:: _static/favicon.png
       :scale: 300 %
       :align: center
       :alt: Альтернативный текст

       Подпись изображения

       Легенда изображения.
```

Рис. 1: Подпись изображения
Легенда изображения.

Параметр `:scale:` устанавливает масштаб изображений.

Параметр `:align:` устанавливает обтекание текстом, может принимать опции `left`, `center` или `right`.

Ещё один способ:

```
.. image:: picture.jpeg
   :height: 100px
   :width: 200 px
   :scale: 50 %
   :alt: alternate text
   :align: right
```

3.3.22 Таблицы

Создавать таблицы можно несколькими способами:

```
.. table:: Заголовок таблицы (Внимание! Отступ таблицы относительно
           команды ..``table::`` обязателен)

    +------------------------+------------+----------+----------+
    | Header row, column 1   | Header 2   | Header 3 | Header 4 |
    | (header rows optional) |            |          |          |
    +========================+============+==========+==========+
    | body row 1, column 1   | column 2   | column 3 | column 4 |
    +------------------------+------------+----------+----------+
    | body row 2             | Cells may span columns.          |
    +------------------------+------------+---------------------+
    | body row 3             | Cells may  | - Table cells       |
    +------------------------+ span rows. | - contain           |
    | body row 4             |            | - body elements.    |
    +------------------------+------------+---------------------+
```

Важно: Отступ таблицы относительно команды `.. table::` обязателен

Результат:

Таблица 1: Заголовок таблицы (Внимание! Отступ таблицы относительно команды `.. table::` обязателен)

Header row, column 1 (header rows optional)	Header 2	Header 3	Header 4
body row 1, column 1	column 2	column 3	column 4
body row 2	Cells may span columns.		
body row 3	Cells may span rows.	• Table cells • contain • body elements.	
body row 4			

Простая таблица:

```
.. table:: Простая таблица
   =====  =====  =======
```

(continues on next page)

(продолжение с предыдущей страницы)

```
   A      B    A and B
=====  =====  =======
False  False  False
True   False  False
False  True   False
True   True   True
=====  =====  =======
```

Результат:

Таблица 2: Простая таблица

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

Ещё один пример:

```
.. table:: Простая таблица со сложной шапкой

   =====  =====  ======
      Inputs     Output
   ------------  ------
     A      B    A or B
   =====  =====  ======
   False  False  False
   True   False  True
   False  True   True
   True   True   True
   =====  =====  ======
```

Результат:

Таблица 3: Простая таблица со сложной шапкой

Inputs		Output
A	B	A or B
False	False	False
True	False	True
False	True	True
True	True	True

Ещё один тип таблицы — CSV-таблица:

```
.. csv-table:: CSV-таблица
   :header: "Treat", "Quantity", "Description"
   :widths: 15, 10, 30

   "Albatross", 2.99, "On a stick!"
   "Crunchy Frog", 1.49, "If we took the bones out, it wouldn't be
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  crunchy, now would it?"
  "Gannet Ripple", 1.99, "On a stick!"
```

Результат:

Таблица 4: CSV-таблица

Treat	Quantity	Description
Albatross	2.99	On a stick!
Crunchy Frog	1.49	If we took the bones out, it wouldn't be crunchy, now would it?
Gannet Ripple	1.99	On a stick!

Примечание: Смотрите также статью reStructuredText (ReST): Быстрый способ ввода таблиц

Ещё один тип таблицы — таблица в виде списка:

```
.. list-table:: Таблица в виде списка
   :widths: 15 10 30
   :header-rows: 1

   * - Treat
     - Quantity
     - Description
   * - Albatross
     - 2.99
     - On a stick!
   * - Crunchy Frog
     - 1.49
     - If we took the bones out, it wouldn't be
       crunchy, now would it?
   * - Gannet Ripple
     - 1.99
     - On a stick!
```

Таблица 5: Таблица в виде списка

Treat	Quantity	Description
Albatross	2.99	On a stick!
Crunchy Frog	1.49	If we took the bones out, it wouldn't be crunchy, now would it?
Gannet Ripple	1.99	On a stick!

3.3.23 Формулы

Вставка формул осуществляется командой `.. math::`. Для ввода формул используется синтаксис LaTeX:

```
.. math::

   \alpha_t(i) = P(O_1, O_2, ... O_t, q_t = S_i \lambda)
```

Результат:

$$\alpha_t(i) = P(O_1, O_2, \ldots O_t, q_t = S_i \lambda)$$

Sphinx расширяет возможности отображения формул, добавляя возможность ссылаться на них. Подробнее в разделе *Вставка формул*. Также смотрите раздел *Некорректно отображаются формулы на Read The Docs*.

Примечание: Таблица математических символов

3.3.24 Блоки примечаний и предупреждений

Блоки примечаний и предупреждений используются для сообщения дополнительной информации. Локализация заголовков и оформление блоков зависит от используемого шаблона. В стандартном шаблоне, используемом на сайте ReadTheDocs.org все блоки имеют собственное оформление, а локализация заголовков зависит от выбранного языка. Также язык настраивается в файле конфигурации Sphinx `conf.py`.

Внимание: Блок **Внимание**, команда: `.. attention::`

Осторожно: Блок **Осторожно**, команда: `.. caution::`

Опасно: Блок **Опасно**, команда: `.. danger::`

Ошибка: Блок **Ошибка**, команда: `.. error::`

Подсказка: Блок **Подсказка**, команда: `.. hint::`

Важно: Блок **Важно**, команда: `.. important::`

Примечание: Блок **Примечание**, команда: `.. note::`

Совет: Блок **Совет**, команда: `.. tip::`

Предупреждение: Блок **Предупреждение**, команда: `.. warning::`

Код блоков выглядит так:

```
.. tip:: Блок **Совет**, команда: ``.. tip::``
```

3.3.25 Содержание

На основе заголовков ReST автоматически создает оглавление, которое вставляется командой `.. contents:::`

```
.. contents:: Оглавление
   :depth: 2

или

.. contents:: Содержание
   :depth: 3
```

Параметр `:depth:` задает уровни заголовков, которые будут включены в оглавление.

Результат:

Оглавление

или

Содержание

3.3.26 Метаданные. Тег META

Имеется возможность добавлять метаданные каждой из страниц непосредственно в rst файлы с помощью директивы `.. meta::`:

```
.. meta::
   :description: The reStructuredText plaintext markup language
   :keywords: plaintext, markup language
```

Будет преобразовано в:

```
<meta name="description"
content="The reStructuredText plaintext markup language">
<meta name="keywords" content="plaintext, markup language">
```

Другие атрибуты:

```
.. meta::
   :description lang=en: An amusing story
   :description lang=fr: Une histoire amusante
```

```
.. meta::
   :http-equiv=Content-Type: text/html; charset=ISO-8859-1
```

Подробнее смотрите раздел HTML-Specific официальной документации reStructuredText.

ГЛАВА 4

Конструкции разметки Sphinx

Sphinx содержит дополнительные конструкции разметки, значительно расширяющие функционал, но которые не поддерживаются стандартной разметкой reStructuredText.

4.1 Автоматическое содержание

Стандартная разметка ReST поддерживает создание в отдельных документах автоматического содержания на основе заголовков. Sphinx расширяет данную функцию и позволяет автоматически создавать общее оглавление для группы документов.

Файл `index.rst` обычно содержит автоматическое оглавление, созданное командой `.. toctree::`:

```
.. toctree::
   :maxdepth: 2
   :numbered:
   :hidden:

   имя_документа1
   имя_документа1
   имя_документа1
```

Команда `.. toctree::` имеет несколько параметров:

- `:maxdepth:` — уровни заголовков, включаемых в оглавление;
- `:numbered:` — нумерация всех пунктов оглавления;
- `:hidden:` — позволяет скрыть оглавление.

После параметров через пустую строку, с отступами, идут названия включаемых файлов, без расширения. Данные названия будут автоматически преобразованы в заголовки разделов.

Параметр `:maxdepth:` не распространяется на LaTeX-документы. Глубина оглавления в LaTeX контролируется его внутренним счетчиком, который можно настроить в файле конфигурации Sphinx `conf.py`, указав в преамбуле значение `\setcounter{tocdepth}{2}`.

Параметр `:hidden:` позволяет Sphinx'у быть в курсе структуры документа, но при этом не отображать оглавление. Удобно, если ссылки на разделы будут указаны, например, на боковой панели.

Подробнее об автоматическом оглавлении в Sphinx смотрите в разделе «The TOC tree» официальной документации Sphinx.

4.2 Примеры исходного кода с подсветкой синтаксиса

Sphinx поддерживает вставку примеров исходного когда с подсветкой синтаксиса на разных языках программирования. Вставка листингов осуществляется командой `.. code-block:: <название яыка>`:

```
.. code-block:: python
   :emphasize-lines: 1-3,5

   def some_function():
       interesting = False
       print 'This line is highlighted.'
       print 'This one is not...'
       print '...but this one is.'
```

Результат:

```
def some_function():
    interesting = False
    print 'This line is highlighted.'
    print 'This one is not...'
    print '...but this one is.'
```

Пример вставки листинга на языке Ruby:

```
.. code-block:: ruby
  :linenos:

  Some more Ruby code.
```

Результат:

```
1 Some more Ruby code.
```

Команда `.. code-block::` имеет следующие параметры:

- `:linenos:` — добавляет нумерацию строк;
- `:emphasize-lines:` — включает подсветку отдельных строк, допускается перечисление одиночных строк через запятую или групп строк через тире.

4.2.1 Вставка примеров кода из файла

Также Sphinx поддерживает вставку кода непосредственно из файла скрипта:

```
.. literalinclude:: example.rb
  :language: ruby
  :emphasize-lines: 3,9-11
  :encoding: latin-1
  :start-after: 2
  :end-before: 15
  :linenos:
```

Параметры `:start-after:` и `:end-before:` позволяют указать с какой по какую строки приводить листинг.

Дополнительную информацию смотрите в разделе «Showing code examples» официальной документации Sphinx.

4.3 Вставка формул

Вставка формул в предложение:

```
Формула в предложении :math:`a^2 + b^2 = c^2`.
```

Формула в предложении $a^2 + b^2 = c^2$.

Выравнивание формул относительно знака равно осуществляется с помощью знака `&`. Перенос строк с помощью `\\`:

```
 .. math::

(a + b)^2  &=  (a + b)(a + b) \\
           &=  a^2 + 2ab + b^2
```

$$\begin{aligned}(a + b)^2 &= (a + b)(a + b) \\ &= a^2 + 2ab + b^2\end{aligned}$$

4.3.1 Нумерация формул

Для нумерации формул необходимо использовать параметр `:label:`:

```
.. math:: e^{i\pi} + 1 = 0
   :label: euler

Формула :eq:`euler` представляет собой Тождество Эйлера.
```

$$e^{i\pi} + 1 = 0 \tag{4.1}$$

Формула (4.1) представляет собой Тождество Эйлера.

Расположение номера относительно формулы зависит от настроек HTML-темы.

Подробнее смотрите главу Math support in Sphinx официальной документации Sphinx.

4.3.2 Отображение формул

Подробнее смотрите раздел *Режим отображения формул*. Также смотрите раздел *Некорректно отображаются формулы на Read The Docs*.

4.3.3 Вставка графиков

Для вставки графиков используются дополнительные расширения, список которых приведен на странице Sphinx Extensions официальной документации Sphinx.

Также смотрите раздел *Подключение расширений*.

4.4 Перекрестные ссылки

Sphinx позволяет создавать перекрестные ссылки между отдельными .rst файлами, подключенными в файле `index.rst`.

4.4.1 Ссылки на разделы

Для того, чтобы создать ссылку на другой раздел, необходимо сначала установить закладку перед этим разделом. Например, для ссылки на пункт *Таблицы* из раздела *Стандартный синтаксис разметки reStructuredText* я использовал следующие команды:

```
.. _rst-markup-label:

Стандартный синтаксис разметки reStructuredText
===============================================


.. _table-label:

Таблицы
~~~~~~~
```

Таким образом, я установил закладки. Теперь можно сослаться на них:

```
Например, для ссылки на пункт :ref:`table-label` из
раздела :ref:`rst-markup-label` я использовал следующие команды:
```

Ссылка осуществляется с помощью команды `:ref:` и названия закладки в обратных кавычках. Закладки автоматически преобразуются в названия разделов, поэтому между закладкой и заголовком ничего не должно находиться.

Аналогичным образом осуществляется ссылка на изображения и таблицы. Для формул используется немного иной синтакси.

4.4.2 Ссылки на изображения

Для ссылки на изображение перед ним также надо поставить закладку, которая автоматически будет преобразовываться в подпись изображения:

```
.. _my-favicon:

.. figure:: img/favicon.png
      :scale: 300 %
      :align: center
      :alt: Альтернативный текст

      Подпись изображения

      Легенда изображения.
```

Теперь сделаем ссылку на изображение *Подпись изображения*:

```
Теперь сделаем ссылку на изображение :ref:`my-favicon`:
```

4.4.3 Ссылки на таблицы

Ссылки на таблицы осуществляются по тому же принципу. Сначала перед таблицей устанавливается закладка, которая потом автоматически преобразуется в название таблицы.

```
.. _cvs-table:

.. csv-table:: CSV-таблица
   :header: "Treat", "Quantity", "Description"
   :widths: 15, 10, 30

   "Albatross", 2.99, "On a stick!"
   "Crunchy Frog", 1.49, "If we took the bones out, it wouldn't be
   crunchy, now would it?"
   "Gannet Ripple", 1.99, "On a stick!"
```

Ссылка на *CSV-таблица*:

```
Ссылка на :ref:`cvs-table`:
```

4.4.4 Ссылки на формулы

Ссылка на формулы осуществляется немного иным способом, с помощью команды `:eq:`. Подробнее смотрите пункт *Нумерация формул*.

4.5 Дополнительные конструкции

4.5.1 Глоссарий

Sphinx позволяет создавать глоссарий с автоматической сортировкой. Элементы глоссария также автоматически попадают в алфавитный указатель.

```
.. glossary::
   :sorted:

   Трансценденция
      Философский термин, характеризующий то, что
      принципиально недоступно опытному познанию
      или не основано на опыте.

   Бозон
      Частица с целым значением спина.
```

Результат:

Бозон Частица с целым значением спина.

Трансценденция Философский термин, характеризующий то, что принципиально недоступно опытному познанию или не основано на опыте.

За автоматическую сортировку отвечает параметр `:sorted:`.

4.5.2 Аббревиатуры

Аббревиатуры вставляются следующим образом, например, LIFO (last-in, first-out):

```
:abbr:`LIFO (last-in, first-out)`
```

4.5.3 Пункты меню

Для обозначения пунктов меню используются команды `menuselection` и `guilabel`:

```
:menuselection:`Файл --> О&ткрыть`
:guilabel:`&Открыть`
```

- *Файл → Открыть*
- *Открыть*

Символ `&` устанавливает в зависити от темы HTML следующему за ним символу нижнее подчеркивание.

4.5.4 Автозамены Sphinx (Подстановки)

Sphinx вводит ряд автозамен, которые не требуют объявления, их значения берутся из конфигурационного файла `conf.py`.

```
Номер релиза: |release|

Номер версии: |version|

Текущая дата: |today|
```

Номер релиза: 1.0

Номер версии: 1.0

Текущая дата: мая 20, 2022

О настройке этих параметров смотрите в пунктах *Версии публикации* и *Настройка отображения даты*.

4.5.5 Смотрите также

```
.. seealso:: Блок с дополнительной информацией.
```

См.также:

Блок с дополнительной информацией. См. также *Блоки примечаний и предупреждений*.

4.5.6 Боковая врезка

Боковая врезка добавляетяс командой `.. sidebar::`

```
.. sidebar:: Боковая врезка

   Оформление врезки зависит от используемой HTML-темы.
```

Боковая врезка

Оформление врезки зависит от используемой HTML-темы.

4.5.7 Рубрики

Рубрики создаются командой `.. rubric::` и используются для создания заголовков, не включаемых в общее содержание.

```
.. rubric:: Пример рубрики

Текст рубрики
```

Пример рубрики

Текст рубрики

4.5.8 Горизонтальный список

```
.. hlist::
   :columns: 3

   * A list of
   * short items
   * that should be
   * displayed
   * horizontally
```

- A list of
- short items
- that should be
- displayed
- horizontally

Примечание: Здесь приведен не полный перечень дополнительных конструкций Sphinx, подробнее в разделе Inline markup и Paragraph-level markup официальной документации Sphinx.

4.5.9 Документация по языкам программирования

Для создания документации по языкам программирования Sphinx имеет специальные команды. Например, для описания тех или иных функций языка может использоваться команда `.. function::` :

```
.. function:: pyfunc()

   Описание функции Python.
```

`pyfunc()`

Описание функции Python.

Эта команда автоматически добавляет функцию в genindex.

Похожие конструкции имеются и для других языков программирования, подробнее смотрите раздел Sphinx Domains официальной документации Sphinx.

4.6 Указатель

Sphinx автоматически генерирует алфавитный указатель, на основе команд `.. glossary::`, `.. function::` и некоторых других (подробнее смотрите раздел Sphinx Domains официальной документации Sphinx).

Но можно и вручную внести элементы в алфавитный указатель с помощью команды `.. index::`.

```
.. index:: Указатель
```

Главная запись индекса предваряется восклицательным знаком:

```
.. index::  ! Указатель
```

Короткая запись:

```
.. index:: BNF, grammar, syntax, notation
```

Ссылка на указатель имеет вид :ref:`genindex` (genindex).

Примечание: Частичный перевод документации Sphinx. Разметка уровня параграфа + Вставка кода

ГЛАВА 5

Система управления версиями Git

В этой главе рассматривается система управления версиями Git. Она не является полным руководством по Git и вряд ли может быть использована в качестве учебника. Скорее это справочник по основным командам.

Для изучения Git с нуля рекомендую прочитать официальное руководство Pro Git (http://git-scm.com/book/ru) или пройти интерактивный русскоязычный тур Git How To (http://githowto.com/ru).

5.1 Установка Git

Установка из исходных кодов описана в официальном руководстве Git (http://git-scm.com/book/ru).

5.1.1 Установка в Ubuntu

```
$ sudo apt-get install git
```

5.1.2 Установка в Fedora

```
$ yum install git-core
```

5.1.3 Установка в OpenSUSE

```
$ zypper in git-core
```

5.1.4 Установка в Mac с помощью графического инсталятора Git

```
http://sourceforge.net/projects/git-osx-installer/
```

5.1.5 Установка в Mac с помощью MacPorts

```
$ sudo port install git-core +svn +doc +bash_completion +gitweb
```

5.1.6 Установка в Windows

```
http://msysgit.github.com/
```

5.2 Первичная настройка

В состав Git входит утилита `git config`, которая позволяет просматривать и устанавливать параметры, контролирующие все аспекты работы Git и его внешний вид. Эти параметры могут быть сохранены в трёх местах:

- Файл `/etc/gitconfig` содержит значения, общие для всех пользователей системы и для всех их репозиториев. Если при запуске `git config` указать параметр `--system`, то параметры будут читаться и сохраняться именно в этот файл.
- Файл `~/.gitconfig` хранит настройки конкретного пользователя. Этот файл используется при указании параметра `--global`.
- Конфигурационный файл в каталоге Git'а (`.git/config`) в том репозитории, где вы находитесь в данный момент. Эти параметры действуют только для данного конкретного репозитория. Настройки на каждом следующем уровне подменяют настройки из предыдущих уровней, то есть значения в `.git/config` перекрывают соответствующие значения в `/etc/gitconfig`.

В системах семейства Windows Git ищет файл `.gitconfig` в каталоге `$HOME` (`C:\Documents and Settings\$USER` или `C:\Users\$USER` для большинства пользователей). Кроме того Git ищет файл `/etc/gitconfig`, но уже относительно корневого каталога MSys, который находится там, куда вы решили установить Git, когда запускали инсталлятор.

5.2.1 Установка имени и электронной почты

Установка имени и адреса электронной почты. Каждый коммит в Git'е содержит эту информацию, и она включена в передаваемые коммиты и не может быть далее изменена:

```
git config --global user.name "Your Name"
git config --global user.email "your_email@whatever.com"
```

5.2.2 Параметры установки окончаний строк

Также, для пользователей Unix/Mac:

```
git config --global core.autocrlf input
git config --global core.safecrlf true
```

5.2.3 Выбор редактора

По умолчанию Git использует стандартный редактор, установленный в системе. Для выбора другого редактора необходимо выполнить команду:

```
$ git config --global core.editor emacs
```

5.2.4 Утилита сравнения

По умолчанию Git использует стандартную утилиту сравнения, для её смены необходимо выполнить команду:

```
$ git config --global merge.tool vimdiff
```

5.2.5 Просмотр настроек

Просмотр всех настроек выполняется командой `git config --list`:

```
$ git config --list
user.name=Scott Chacon
user.email=schacon@gmail.com
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
...
```

Также вы можете проверить значение конкретного ключа, выполнив `git config {ключ}`:

```
$ git config user.name
Scott Chacon
```

5.2.6 Псевдонимы в Git

Можно настроить псевдонимы (alias) для любой команды с помощью `git config`.

```
$ git config --global alias.co checkout
$ git config --global alias.br branch
$ git config --global alias.ci commit
$ git config --global alias.st status
```

Это означает, что, например, вместо набирания `git commit`, вам достаточно набрать только `git ci`.

5.3 Основные команды

5.3.1 Получение справки

```
$ git help <команда>
$ git <команда> --help
$ man git-<команда>
```

5.3.2 Создание репозитория в существующем каталоге

```
$ git init
```

5.3.3 Клонирование существующего репозитория

```
$ git clone <адрес репозитория>
```

Например:

```
$ git clone git://github.com/schacon/grit.git
```

Клонирование в другой каталог осуществляется командой:

```
$ git clone git://github.com/schacon/grit.git <имя другого каталога>
```

5.3.4 Определение состояния файлов

```
$ git status
```

5.3.5 Отслеживание новых файлов

```
$ git add <имя файла>
```

5.3.6 Игнорирование файлов

Настройки игнорирования файлов, которые не должны обрабатываться git, находятся в файле .gitignore: в корне репозитория. Пример файла .gitignore:

```
# комментарий - эта строка игнорируется
# не обрабатывать файлы, имя которых заканчивается на .a
*.a
# НО отслеживать файл lib.a, несмотря на то, что мы игнорируем все .a файлы с помощью␣
↪предыдущего правила
!lib.a
# игнорировать только файл TODO находящийся в корневом каталоге, не относится к файлам␣
↪вида subdir/TODO
/TODO
# игнорировать все файлы в каталоге build/
build/
# игнорировать doc/notes.txt, но не doc/server/arch.txt
doc/*.txt
# игнорировать все .txt файлы в каталоге doc/
doc/**/*.txt
```

5.3.7 Просмотр индексированных и неиндексированных изменений

Для просмотра непроиндексированных изменений:

```
$ git diff
```

Для просмотра проиндексированных изменений, которые войдут в следующий коммит:

```
git diff --cached
```

Для просмотра изменений в отдельном файле:

```
git diff <имя файла>
```

5.3.8 Фиксация изменений

```
$ git commit -m "Комментарий"
```

Внимание: Изменения, для которых не выполнена команда `git add` не будут зафиксированы.

5.3.9 Игнорирование индексации

Добавление параметра `-a` в команду `git commit` заставляет Git автоматически индексировать каждый уже отслеживаемый на момент коммита файл, позволяя вам обойтись без `git add`:

```
git commit -a -m "Комментарий"
```

5.3.10 Удаление файлов

Полное удаление файла и из под наблюдения Git и с жесткого диска:

```
git rm <имя файла>
```

Если файл был изменён и уже проиндексирован, необходимо использовать принудительное удаление с помощью параметра `-f`.

Удаление файла и из под наблюдения Git, но не с жесткого диска:

```
git rm --cached <имя файла>
```

5.3.11 Перемещение файлов

```
$ git mv <имя файла> <новое имя файла>
```

Эквивалентно выполнению следующих команд:

```
$ mv <имя файла>  <новое имя файла>
$ git rm <имя файла>
$ git add <новое имя файла>
```

5.3.12 Просмотр истории коммитов

По умолчанию, без аргументов, `git log` выводит список коммитов созданных в данном репозитории в обратном хронологическом порядке. То есть самые последние коммиты показываются первыми.

```
git log
```

Вывод в собственном формате:

```
git log --pretty=format:"%h - %an, %ar : %s"
```

Команда `log` содержит много других дополнительных параметров. Подробнее о них написано в официальном руководстве Git (http://git-scm.com/book/ru).

5.3.13 Изменение последнего коммита

```
git commit --amend
```

5.3.14 Отмена индексации файла

```
git reset HEAD <имя файла>
```

5.3.15 Отмена изменений файла

```
git checkout -- <имя файла>
```

5.3.16 Просмотр меток (тегов)

```
$ git tag
```

Поиск меток(тегов) по шаблону:

```
$ git tag -l 'v1.4.2.*'
```

5.3.17 Создание легковесных меток(тегов)

```
git tag <метка>
```

5.3.18 Создание аннотированных меток(тегов)

```
git tag -a <метка> -m '<комментарий>'
```

5.3.19 Выставление меток(тегов) позже

Также возможно помечать уже пройденные коммиты. Предположим, что история коммитов выглядит следующим образом:

```
$ git log --pretty=oneline
9fceb02d0ae598e95dc970b74767f19372d61af8 updated rakefile
964f16d36dfccde844893cac5b347e7b3d44abbc commit the todo
8a5cbc430f1a9c3d00faaeffd07798508422908a updated readme
```

Для отметки коммита укажите его контрольную сумму (или её часть) в конце команды:

```
$ git tag -a v1.2 -m 'version 1.2' 9fceb02
```

5.3.20 Обмен метками(тегами)

По умолчанию, команда `git push` не отправляет метки на удалённые серверы. Необходимо явно отправить (push) метки на общий сервер после того, как они были созданы. Это делается так же, как и выкладывание в совместное пользование удалённых веток — нужно выполнить `git push origin [имя метки]`.

Чтобы отправить все метки за один раз, нужно использовать опцию `--tags`:

```
$ git push origin --tags
```

5.3.21 Удаление меток(тегов)

```
git tag -d <имя метки>
```

5.3.22 Отображение удалённых репозиториев

```
$ git remote -v
```

5.3.23 Добавление удалённых репозиториев

```
$ remote add [сокращение] [url]:
```

5.3.24 Извлечение данных из удаленного репозитория

Без слияния:

```
$ git fetch [имя удал. сервера]
```

Со слиянием:

```
$ git pull [имя удал. сервера]
```

5.3.25 Отправка данных в удаленный репозиторий

```
git push [удал. сервер] [ветка]
```

5.3.26 Инспекция удалённого репозитория

```
$ git remote show [удал. сервер]
```

5.3.27 Удаление и переименование удалённых репозиториев

Переименование:

```
$ git remote rename <старое имя> <новое имя>
```

Удаление:

```
$ git remote rm <имя репозитория>
```

5.3.28 Создание новой ветки

```
$ git branch <имя ветки>
```

Чтобы создать ветку и сразу же перейти на неё, необходимо выполнить команду `git checkout` с ключом `-b`:

```
$ git checkout -b <имя ветки>
```

5.3.29 Переход на другую ветку

```
$ git checkout <имя ветки>
```

5.3.30 Слияние веток

```
$ git merge <имя ветки>
```

5.3.31 Удаление ветки

```
$ git branch -d <имя ветки>
```

Для удаления не слитых веток необходимо использовать команду с аргументом `-D`:

```
$ git branch -D <имя ветки>
```

5.3.32 Состояние веток

Показать текущую ветку:

```
$ git branch
```

Показать последний коммит на каждой из веток:

```
$ git branch -v
```

Посмотреть ветки, которые уже слиты с текущей:

```
$ git branch --merged
```

Посмотреть ветки, которые ещё не слиты с текущей:

```
$ git branch --no-merged
```

5.3.33 Перемещение изменений между ветками

При помощи команды `rebase` можно взять все изменения, которые попали в коммиты на одной из веток, и повторить их на другой. Для этого сначала надо переместиться на ветку (команда `git checkout <имя ветки>`), в которую будут переноситься изменения, а затем выполнить команду:

```
$ git rebase <имя ветки из которой переносятся изменения>
```

5.3.34 Отслеживание веток

```
$ git checkout -b [ветка] [удал. сервер]/[ветка]
```

Для Git версии 1.6.2 или более поздних, можно также воспользоваться сокращением `--track`:

```
$ git checkout --track origin/serverfix
```

Чтобы настроить локальную ветку с именем, отличным от имени удалённой ветки, можно легко использовать первую версию с другим именем локальной ветки:

```
$ git checkout -b sf origin/serverfix
```

5.3.35 Удаление веток на удалённом сервере

```
$ git push [удал. сервер] :[ветка]
```

5.3.36 Прятанье

```
$ git stash
```

Вывести список спрятанного:

```
$ git stash list
stash@{0}: WIP on master: 049d078 added the index file
stash@{1}: WIP on master: c264051... Revert "added file_size"
stash@{2}: WIP on master: 21d80a5... added number to log
```

Применить спрятанное:

```
$ git stash apply stash@{2}
```

Создание ветки из спрятанных изменений:

```
$ git stash branch <имя ветки>
```

Примечание: В главе описаны основные команды Git. Это далеко не полный перечень его возможностей. Исчерпывающую информацию по работе и настройке Git можно получить в официальном руководстве Pro Git (http://git-scm.com/book/ru).

ГЛАВА 6

GitHub и Read The Docs

В главе описывается взаимодействие двух сервисов и приводятся рекомендации по их настройке.

6.1 Работа с GitHub

GitHub обладает богатым набором функций. В данном разделе будут описаны только основные функции, необходимые для совместной работы над документацией. За дополнительной информацией обратитесь к следующим материалам:

- Официальное руководство Pro Git
- Интерактивный русскоязычный тур Git How To
- UnixLab: GitHub и pastebin: что и зачем?
- Github — без командной строки

6.1.1 Что такое GitHub

GitHub (https://github.com) — веб-сервис для хостинга IT-проектов и их совместной разработки, основанный на системе контроля версий Git. Сервис абсолютно бесплатен для проектов с открытым исходным кодом. Для создания закрытых проектов предлагаются различные платные тарифные планы.

GitHub позволяет:

- Размещать исходный код программ;
- Комментировать и обсуждать правки;
- Следить за обновлениями проектов;
- Копировать и скачивать репозитории проектов;
- Объединять репозитории;

- Создавать небольшие Вики проектов;
- Редактировать файлы непосредственно через веб-интерфейс;
- Создавать статичные HTML-страницы проектов;
- Для платных аккаунтов доступно создание приватных репозиториев, доступных ограниченному кругу пользователей.

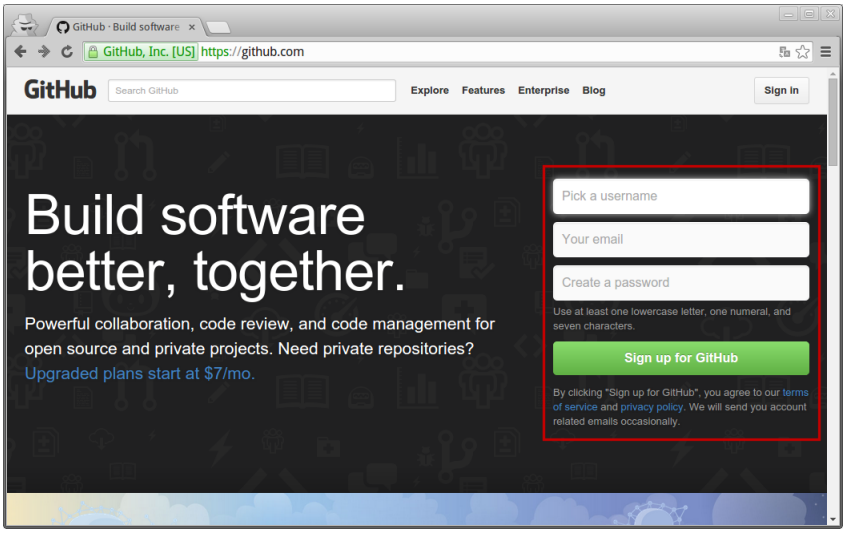
Кроме Git, сервис поддерживает получение и редактирование кода через SVN и Mercurial.

Также на сайте есть pastebin-сервис gist.github.com для быстрой публикации фрагментов кода.

Для комфортной работы с GitHub, необходимо уметь работать с системой контроля версий Git, подробнее смотрите главу *Система управления версиями Git*.

6.1.2 Регистрация

Процедура регистрации проста, бесплатна и не занимает много времени. Достаточно заполнить три поля: логин, адрес электронной почты и пароль. На почту будет выслан запрос на подтверждение регистрации.



6.1.3 Создание репозитория

GitHub — это не простой хостинг файлов, он завязан на систему управления версиями файлов и служит для выгрузки локальных пользовательских git-репозиториев.

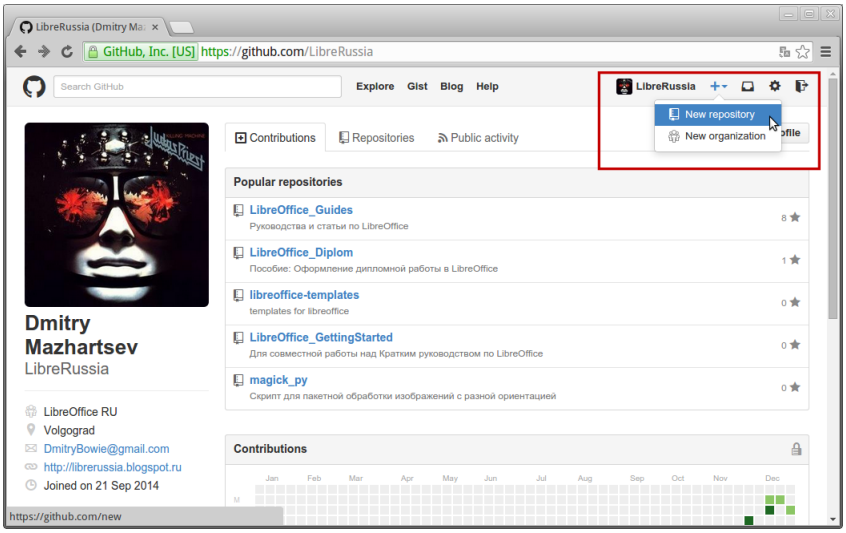
Для создания репозитория нажмите значок `+` и выберите пункт `New repository`.

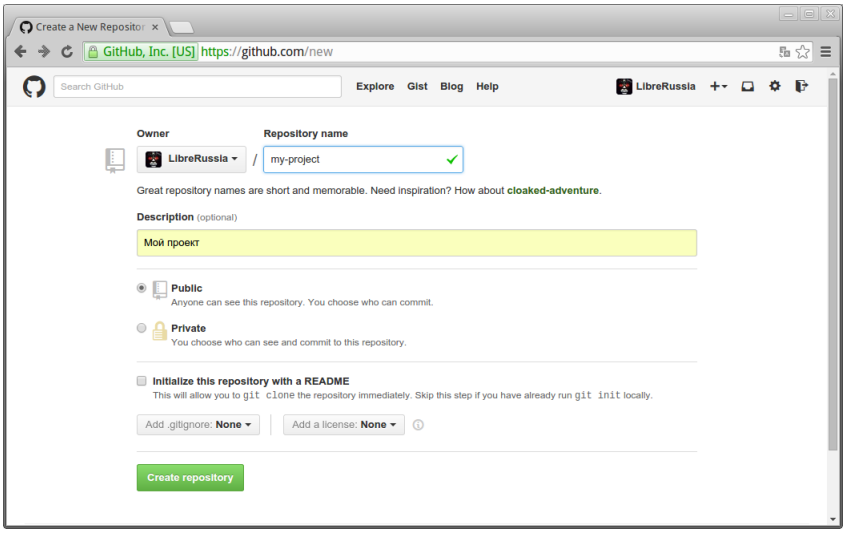
Будет открыта страница, на которой задается название репозитория и его описание.

Чтобы создать новый репозиторий на компьютере и выгрузить его в репозиторий на GitHub, выполните в командной строке:

```
touch README.md
git init
git add README.md
git commit -m "first commit"
```

(continues on next page)


LibreRussia (Dmitry Ma
GitHub, Inc. [US] https://github.com/LibreRussia
Search GitHub
Explore Gist Blog Help
LibreRussia
New repository
New organization
Contributions
Repositories
Public activity
Popular repositories
LibreOffice_Guides
Руководства и статьи по LibreOffice
LibreOffice_Diplom
Пособие: Оформление дипломной работы в LibreOffice
libreoffice-templates
templates for libreoffice
LibreOffice_GettingStarted
Для совместной работы над Кратким руководством по LibreOffice
magick_py
Скрипт для пакетной обработки изображений с разной ориентацией
Dmitry Mazhartsev
LibreRussia
LibreOffice RU
Volgograd
DmitryBowie@gmail.com
http://librerussia.blogspot.ru
Joined on 21 Sep 2014
Contributions
https://github.com/new


Create a New Repositor
GitHub, Inc. [US] https://github.com/new
Search GitHub
Explore Gist Blog Help
LibreRussia
Owner
Repository name
LibreRussia
my-project
Great repository names are short and memorable. Need inspiration? How about cloaked-adventure.
Description (optional)
Мой проект
Public
Anyone can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with a README
This will allow you to git clone the repository immediately. Skip this step if you have already run git init locally.
Add .gitignore: None
Add a license: None
Create repository

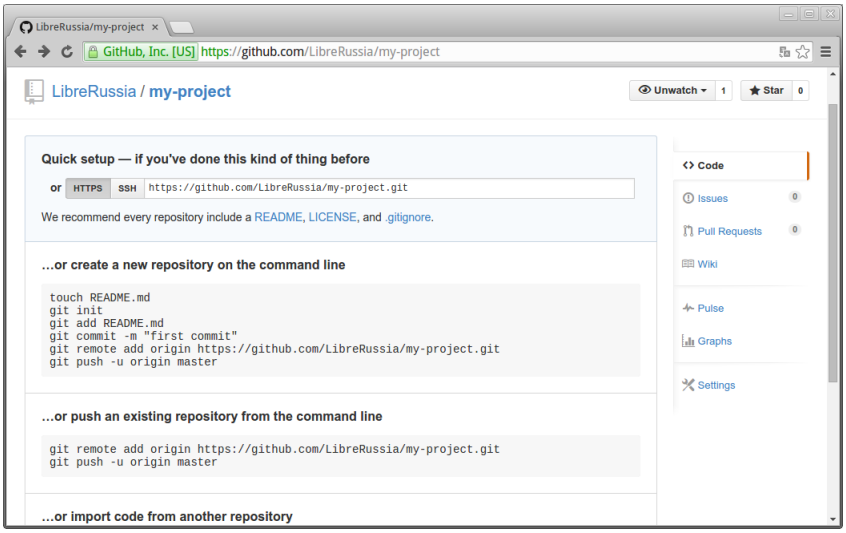
(продолжение с предыдущей страницы)

```
git remote add origin https://github.com/LibreRussia/my-project.git
git push -u origin master
```

Примечание: Используйте URL-адрес вашего репозитория для команды `remote`.

Чтобы выгрузить в репозиторий на GitHub уже существующий репозиторий выполните в командной строке:

```
git remote add origin https://github.com/LibreRussia/my-project.git
git push -u origin master
```



Выгружать файлы в репозиторий на GitHub могут только те пользователи, которые имеют право на запись. Загрузить же копию репозитория на компьютер может любой пользователей, если это не ограничено настройками приватности репозитория (доступно в платных тарифах).

Также имеется возможность создавать совместные для нескольких аккаунтов репозитории. В целях безопасности я предпочитаю не давать никому доступ к основному репозиторию и работаю через систему форков (см. *Копирование репозитория (Fork)*).

6.1.4 Копирование репозитория (Fork)

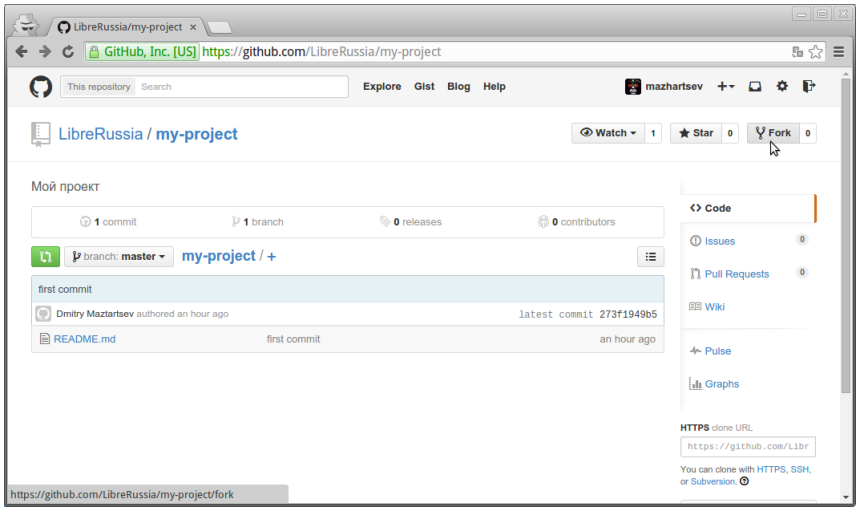
Любой желающий может сделать полную копию(англ. *Fork* — разветвление) чужого репозитория и работать над ней независимо от основного. После внесения всех изменений, можно предложить их в основной репозиторий (сделать, так называемый, *Pull Requests*).

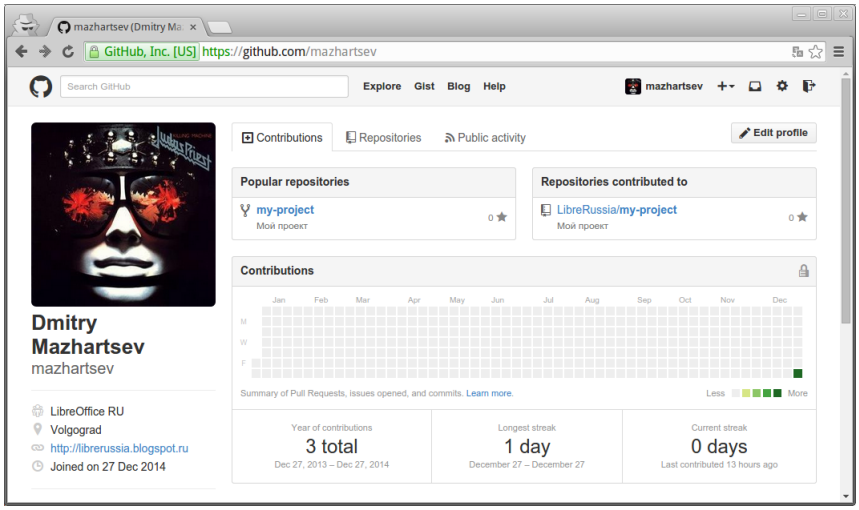
Чтобы создать форк, необходимо зайти в репозиторий и нажать кнопку `Fork`.

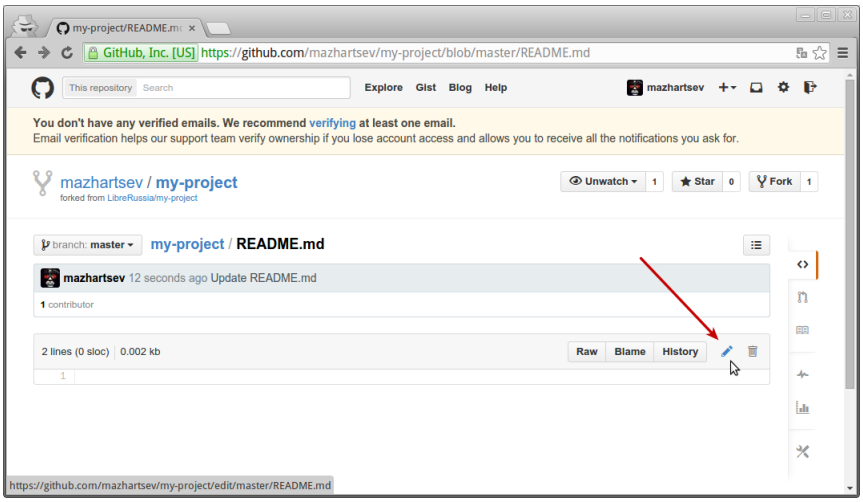
После чего в аккаунте пользователя появится полная копия репозитория.

В качестве примера я сделал форк репозитория `My_Project` пользователя LibreRussia. Теперь в моем аккаунте (mazhartsev) есть его полная копия, с которой я могу спокойно работать не опасаясь покорёжить основной.

Теперь внесем изменения в файл `README.md`, выбрав его и нажав значок с карандашом.

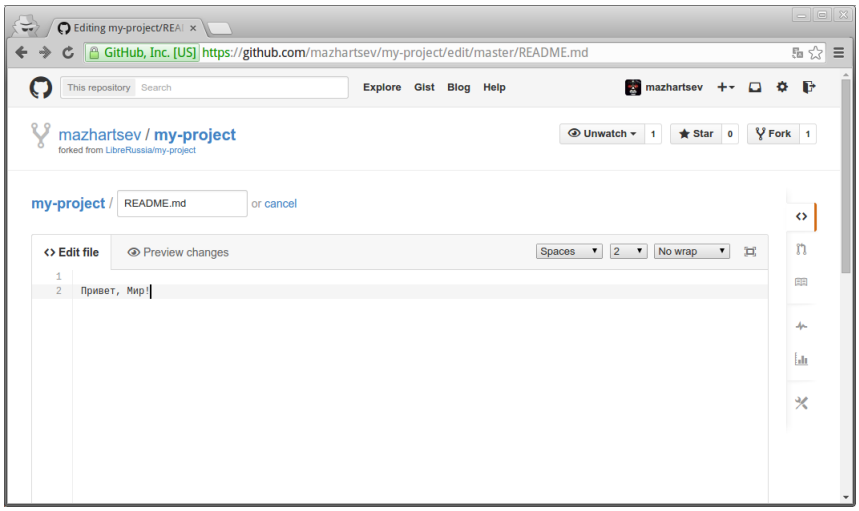

LibreRussia/my-project
GitHub, Inc. [US] https://github.com/LibreRussia/my-project
This repository Search
Explore Gist Blog Help
mazhartsev
LibreRussia / my-project
Watch 1
Star 0
Fork 0
Мой проект
1 commit
1 branch
0 releases
0 contributors
branch: master
my-project / +
first commit
Dmitry Maztartsev authored an hour ago
latest commit 273f1949b5
README.md
first commit
an hour ago
Code
Issues 0
Pull Requests 0
Wiki
Pulse
Graphs
HTTPS clone URL
https://github.com/Libr
You can clone with HTTPS, SSH, or Subversion.
https://github.com/LibreRussia/my-project/fork


mazhartsev (Dmitry Ma
GitHub, Inc. [US] https://github.com/mazhartsev
Search GitHub
Explore Gist Blog Help
mazhartsev
Contributions
Repositories
Public activity
Edit profile
Popular repositories
my-project
Мой проект
0
Repositories contributed to
LibreRussia/my-project
Мой проект
0
Contributions
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Summary of Pull Requests, issues opened, and commits. Learn more.
Less More
Dmitry Mazhartsev
mazhartsev
LibreOffice RU
Volgograd
http://liberussia.blogspot.ru
Joined on 27 Dec 2014
Year of contributions
3 total
Dec 27, 2013 – Dec 27, 2014
Longest streak
1 day
December 27 – December 27
Current streak
0 days
Last contributed 13 hours ago


my-project/README.m
GitHub, Inc. [US] https://github.com/mazhartsev/my-project/blob/master/README.md
This repository Search
Explore Gist Blog Help
mazhartsev
You don't have any verified emails. We recommend verifying at least one email.
Email verification helps our support team verify ownership if you lose account access and allows you to receive all the notifications you ask for.
mazhartsev / my-project
forked from LibreRussia/my-project
Unwatch 1
Star 0
Fork 1
branch: master
my-project / README.md
mazhartsev 12 seconds ago Update README.md
1 contributor
2 lines (0 sloc) 0.002 kb
Raw Blame History
1
https://github.com/mazhartsev/my-project/edit/master/README.md

Одна из дополнительных приятных функций GitHub — возможность редактировать текстовые файлы через браузер.

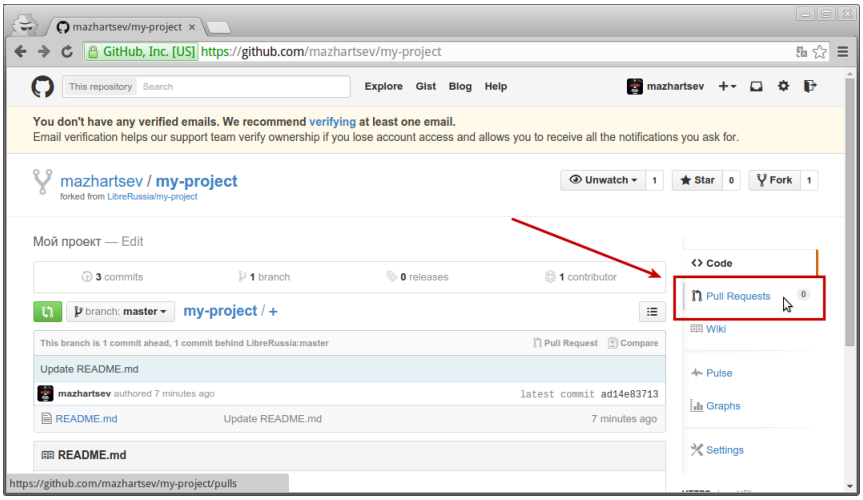
Добавим строку `Привет, Мир!` и сохраним файл.



Теперь предложим наши изменения в основной репозиторий, то есть сделаем *Pull Requests*.

6.1.5 Pull Requests

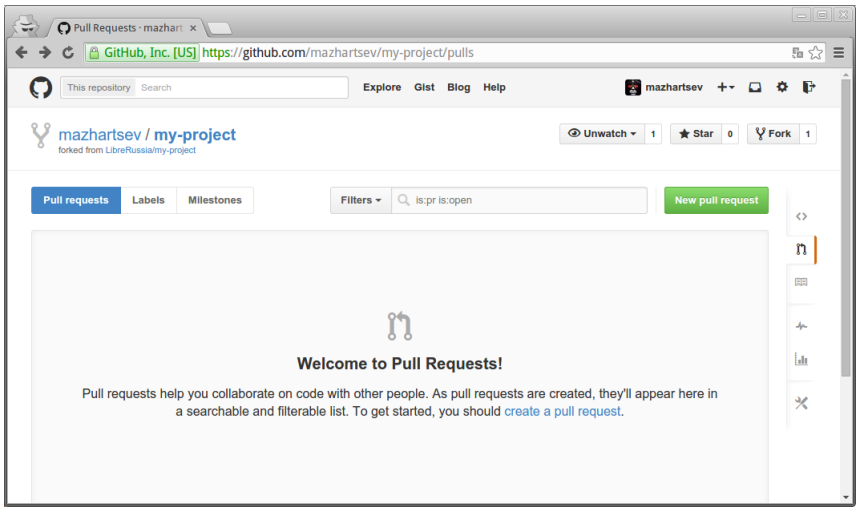
Для того, чтобы предложить свои изменения в основной репозиторий, необходимо перейти в раздел Pull Requests и нажать кнопку `New pull request`.

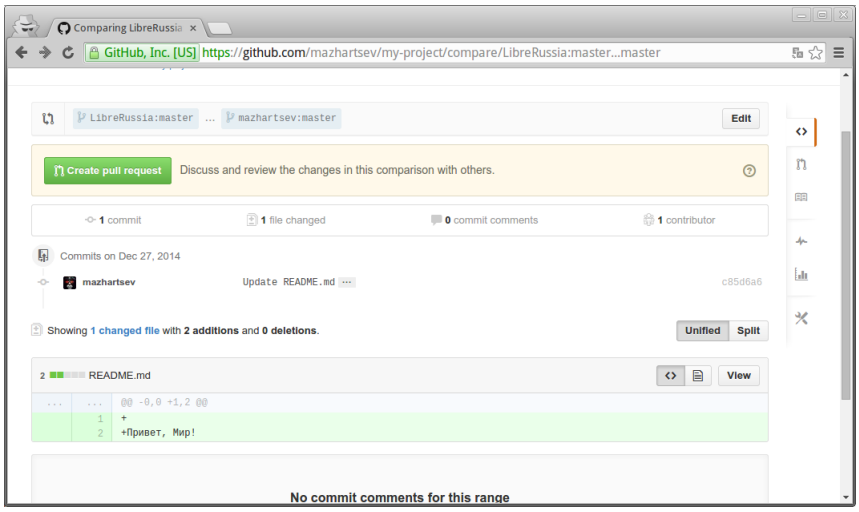


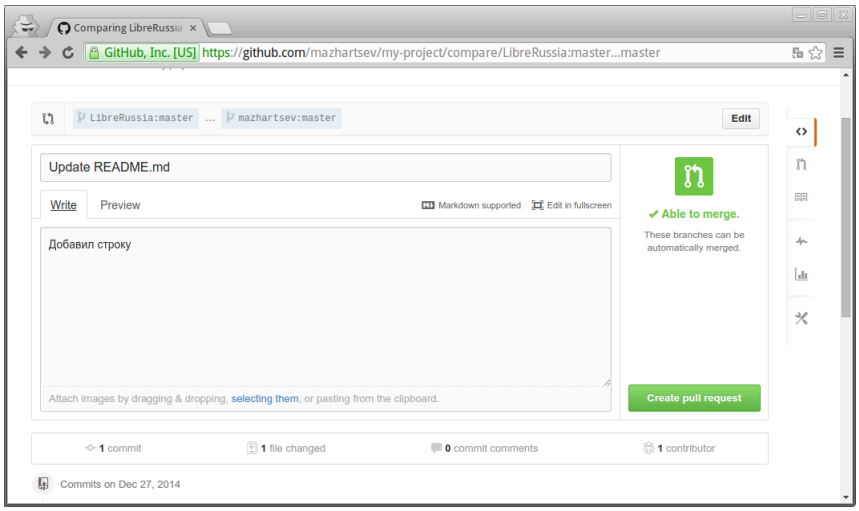
Будет открыта страница, показывающая внесенные изменения. Нажимаем `Create pull request`.

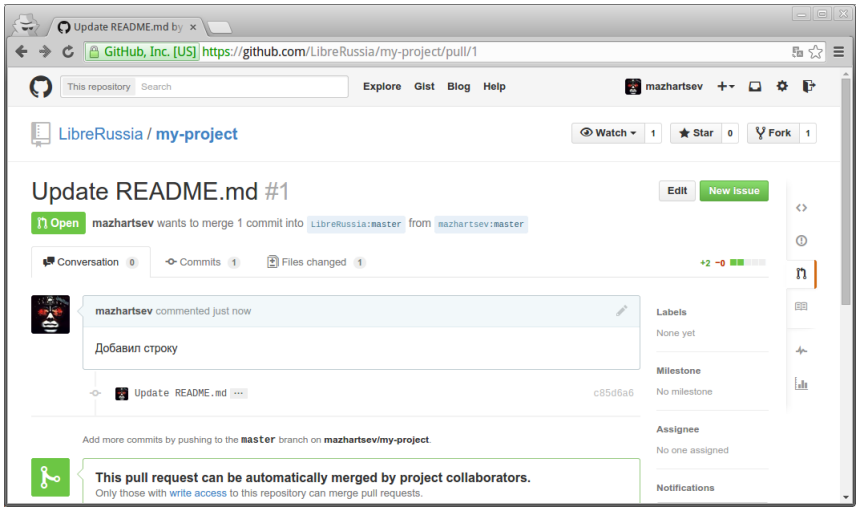
Далее будет предложено оставить комментарий к предлагаемым изменениям. Подробно описывать все внесенные изменения не надо, Git сам покажет их.

Внесенные изменения отправлены на рассмотрение владельцу основного репозитория, который получит соответствующие уведомления на электронный адрес и в интерфейсе GitHub. Для демонстрации я перейду из своего аккаунта в аккаунт LibreRussia.


Pull Requests · mazhart
GitHub, Inc. [US] https://github.com/mazhartsev/my-project/pulls
This repository Search
Explore Gist Blog Help
mazhartsev
mazhartsev / my-project
forked from LibreRussia/my-project
Unwatch 1
Star 0
Fork 1
Pull requests
Labels
Milestones
Filters
is:pr is:open
New pull request
Welcome to Pull Requests!
Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should create a pull request.

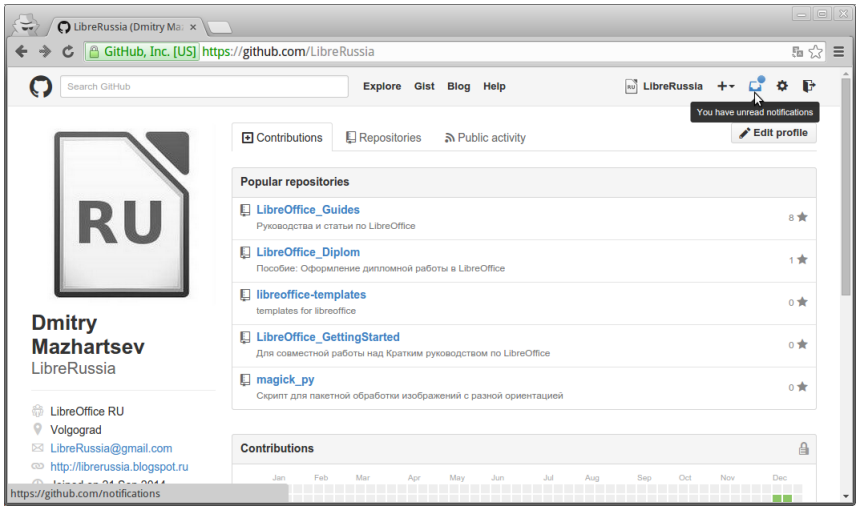

Comparing LibreRussia
GitHub, Inc. [US] https://github.com/mazhartsev/my-project/compare/LibreRussia:master...master
LibreRussia:master ... mazhartsev:master
Edit
Create pull request
Discuss and review the changes in this comparison with others.
1 commit
1 file changed
0 commit comments
1 contributor
Commits on Dec 27, 2014
mazhartsev
Update README.md
c85d6a6
Showing 1 changed file with 2 additions and 0 deletions.
Unified
Split
README.md
View
@@ -0,0 +1,2 @@
+
+Привет, Мир!
No commit comments for this range


Comparing LibreRussia
GitHub, Inc. [US] https://github.com/mazhartsev/my-project/compare/LibreRussia:master...master
LibreRussia:master ... mazhartsev:master
Edit
Update README.md
Write
Preview
Markdown supported
Edit in fullscreen
Добавил строку
Attach images by dragging & dropping, selecting them, or pasting from the clipboard.
Able to merge.
These branches can be automatically merged.
Create pull request
1 commit
1 file changed
0 commit comments
1 contributor
Commits on Dec 27, 2014



Принятие Pull Requests

Уведомление о предложенных изменениях приходят на электронную почту владельца репозитория и отображаются в интерфейсе GitHub.



Откроем уведомления.

Чтобы выполнить слияние (англ. *Merge*) предложенных изменений нажмите кнопку `Merge pull request`.

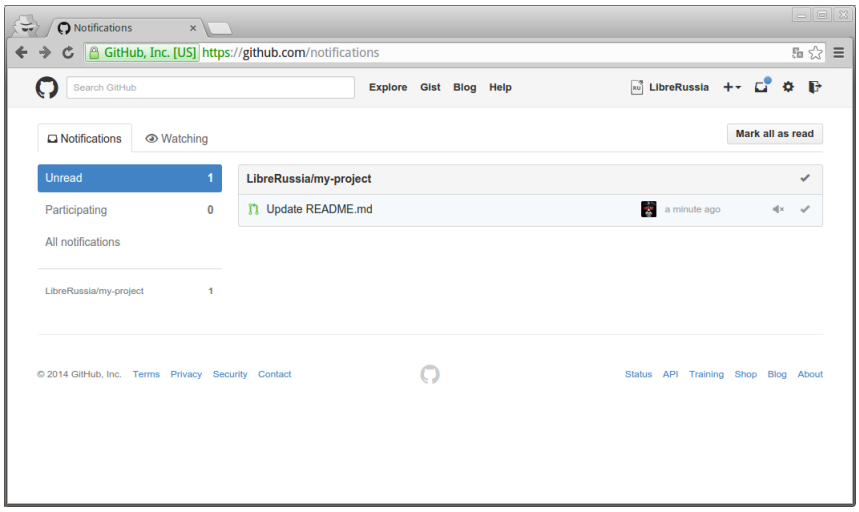
Будет открыта страница сравнения версий. Левая колонка отображает начальную версию файла, которая сейчас находится в основном репозитории. Правая колонка отображает изменения внесенные в новый файл. Плюсами отмечены добавленные строки.

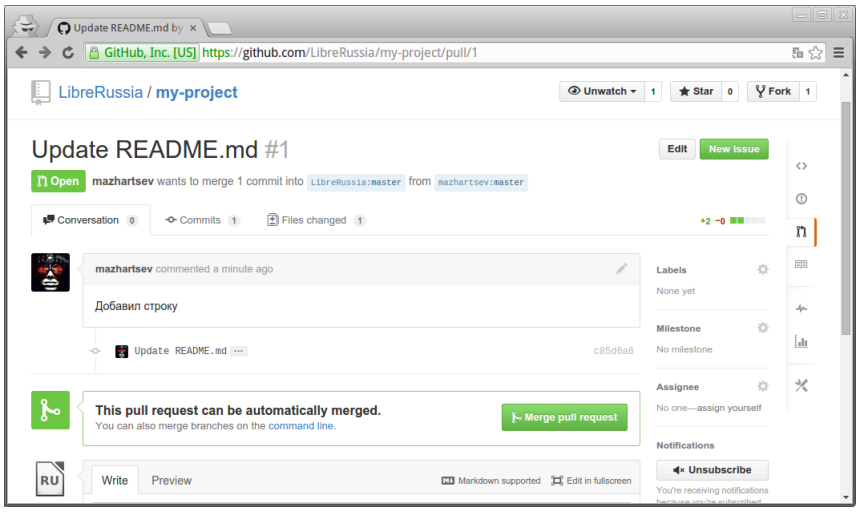
Теперь проверим файл в основном репозитории.

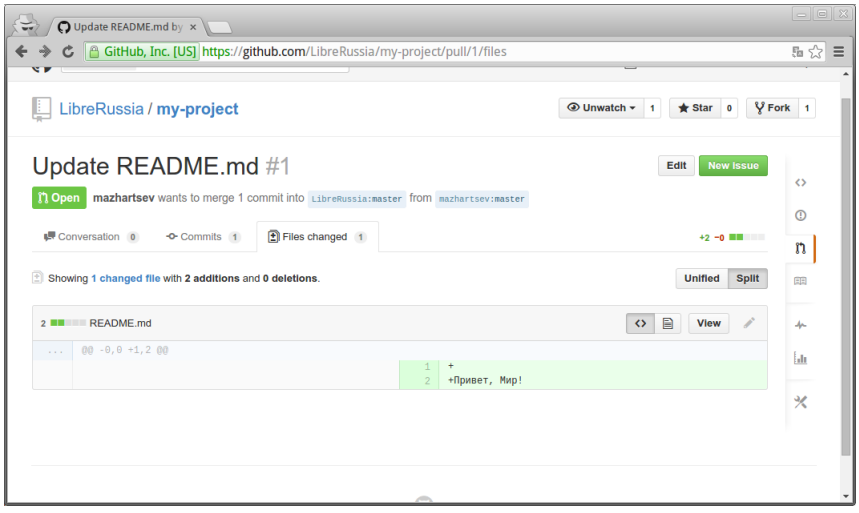
Как видно, файл успешно изменен и в нём появилась строка, добавленная другим пользователем.

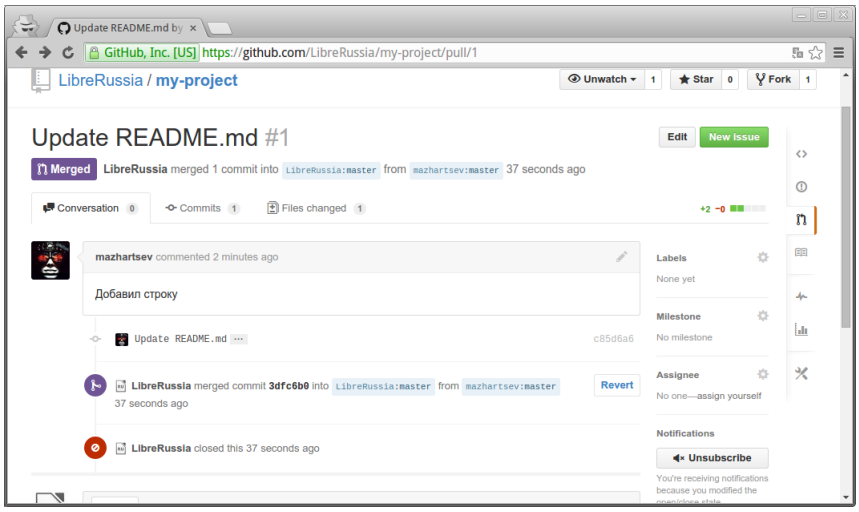
В любой момент можно отменить сделанные изменения и откатиться к предыдущим версиям файла. Все это больше относится к функциям самой системы управления версиями Git.

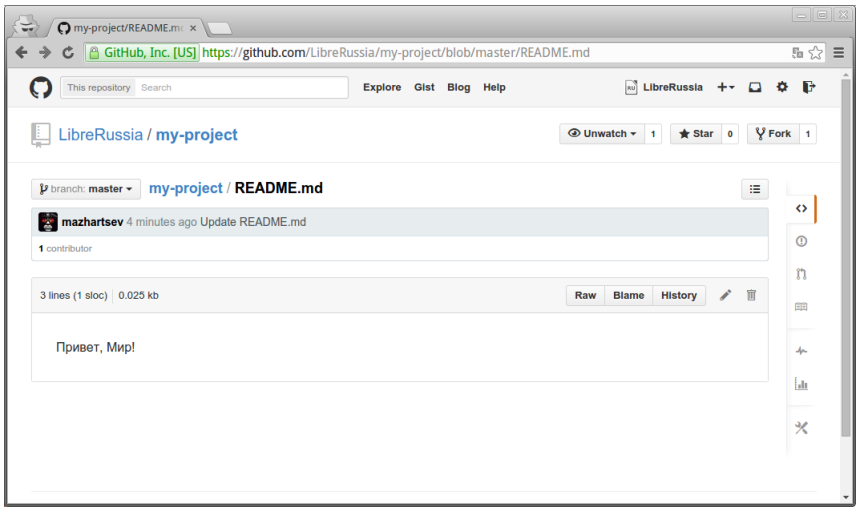
Подробнее о слияниях смотрите главу *Ветвление в Git* из официального руководства Pro Git.

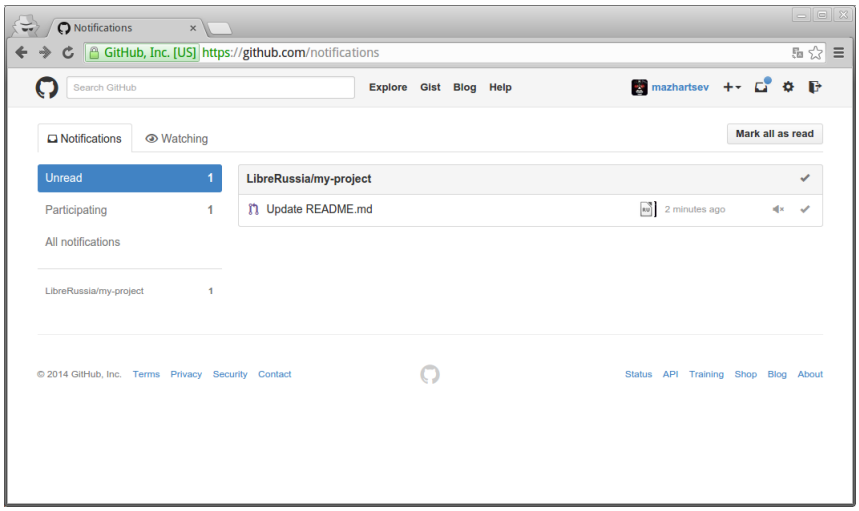

Notifications
GitHub, Inc. [US] https://github.com/notifications
Search GitHub
Explore Gist Blog Help
LibreRussia
Notifications
Watching
Mark all as read
Unread 1
Participating 0
All notifications
LibreRussia/my-project 1
LibreRussia/my-project
Update README.md
a minute ago
© 2014 GitHub, Inc. Terms Privacy Security Contact
Status API Training Shop Blog About


Update README.md by
GitHub, Inc. [US] https://github.com/LibreRussia/my-project/pull/1
LibreRussia / my-project
Unwatch 1
Star 0
Fork 1
Update README.md #1
Edit
New Issue
Open
mazhartsev wants to merge 1 commit into LibreRussia:master from mazhartsev:master
Conversation 0
Commits 1
Files changed 1
+2 −0
mazhartsev commented a minute ago
Добавил строку
Update README.md
c85d6a6
Labels
None yet
Milestone
No milestone
Assignee
No one—assign yourself
Notifications
Unsubscribe
This pull request can be automatically merged.
You can also merge branches on the command line.
Merge pull request
RU
Write
Preview
Markdown supported
Edit in fullscreen


Update README.md by
GitHub, Inc. [US] https://github.com/LibreRussia/my-project/pull/1/files
LibreRussia / my-project
Unwatch 1
Star 0
Fork 1
Update README.md #1
Edit
New Issue
Open
mazhartsev wants to merge 1 commit into LibreRussia:master from mazhartsev:master
Conversation 0
Commits 1
Files changed 1
+2 −0
Showing 1 changed file with 2 additions and 0 deletions.
Unified
Split
2
README.md
View
@@ -0,0 +1,2 @@
+
+Привет, Мир!


Update README.md by
GitHub, Inc. [US] https://github.com/LibreRussia/my-project/pull/1
LibreRussia / my-project
Unwatch 1
Star 0
Fork 1
Update README.md #1
Edit
New issue
Merged
LibreRussia merged 1 commit into LibreRussia:master from mazhartsev:master 37 seconds ago
Conversation 0
Commits 1
Files changed 1
+2 −0
mazhartsev commented 2 minutes ago
Добавил строку
Update README.md
c85d6a6
LibreRussia merged commit 3dfc6b0 into LibreRussia:master from mazhartsev:master
37 seconds ago
Revert
LibreRussia closed this 37 seconds ago
Labels
None yet
Milestone
No milestone
Assignee
No one—assign yourself
Notifications
Unsubscribe
You're receiving notifications because you modified the


my-project/README.md
GitHub, Inc. [US] https://github.com/LibreRussia/my-project/blob/master/README.md
This repository Search
Explore Gist Blog Help
LibreRussia
LibreRussia / my-project
Unwatch 1
Star 0
Fork 1
branch: master
my-project / README.md
mazhartsev 4 minutes ago Update README.md
1 contributor
3 lines (1 sloc) 0.025 kb
Raw Blame History
Привет, Мир!


Notifications
GitHub, Inc. [US] https://github.com/notifications
Search GitHub
Explore Gist Blog Help
mazhartsev
Notifications
Watching
Mark all as read
Unread 1
Participating 1
All notifications
LibreRussia/my-project
Update README.md
2 minutes ago
LibreRussia/my-project 1
© 2014 GitHub, Inc. Terms Privacy Security Contact
Status API Training Shop Blog About

Слияние копий репозиториев похоже на слияние двух веток, подробнее смотрите *Ветвление*.

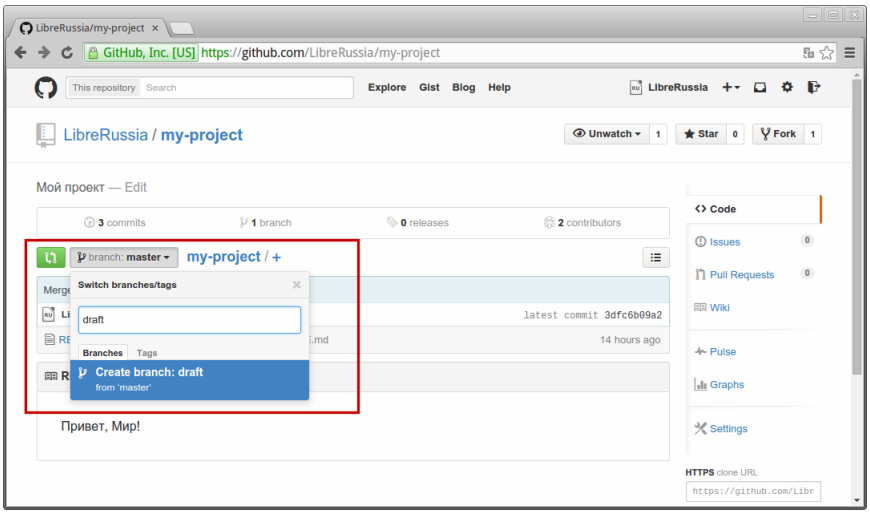
Статьи на тему Pull Requests:

- UnixLab: Как сделать pull request
- Pull request'ы на GitHub или Как мне внести изменения в чужой проект
- Кнопка слияния на GitHub (Merge)

6.1.6 Ветвление

Внутри одного репозитория можно создавать параллельные ветви (англ. *Branch*). Ветвление означает, что вы отклоняетесь от основной линии разработки и продолжаете работу, не вмешиваясь в основную линию. В любой момент можно выполнить слияние веток. Подробнее о ветвлении смотрите главу *Ветвление в Git* из официального руководства Pro Git.

Ветки напоминают функции *Копирование репозитория (Fork)* и *Pull Requests*, но при этом всё происходит в рамках одного репозитория.



Данная функция очень полезна при создании документации, так как именно на ней основывается возможность вести одновременно несколько версий руководства (см. *Несколько версий документации*).

6.1.7 Обход блокировки GitHub

Не так давно случился неприятный инцидент с блокировкой GitHub на территории России. Не буду давать какую-либо оценку данному событию, но на случай повторения ситуации приведу способы обхода блокировки.

Я воспользовался утилитой `torsocks`, которая устанавливается вместе с Tor. Чтобы установить Tor в Debian/Ubuntu выполните в терминале:

```
sudo apt-get install tor
```

Пакет tor включает программу torsocks. Если к команде запуска программы (например, git) приписать torsocks, то её сетевая активность (включая разрешение доменов) пойдёт через Tor. Сетевая активность, которую нельзя пропустить через Tor, будет отсекаться (например, UDP). Данный способ не работает в Windows.

```
torsocks git push
torsocks git pull
```

Другие способы обхода смотрите в статьях:

- Краткая инструкция: GitHub через Tor
- Обход блокировки github.com в Windows с помощью DNSCrypt
- Краткая инструкция: GitHub через I2P

6.1.8 GitHub и совместная работа над документацией

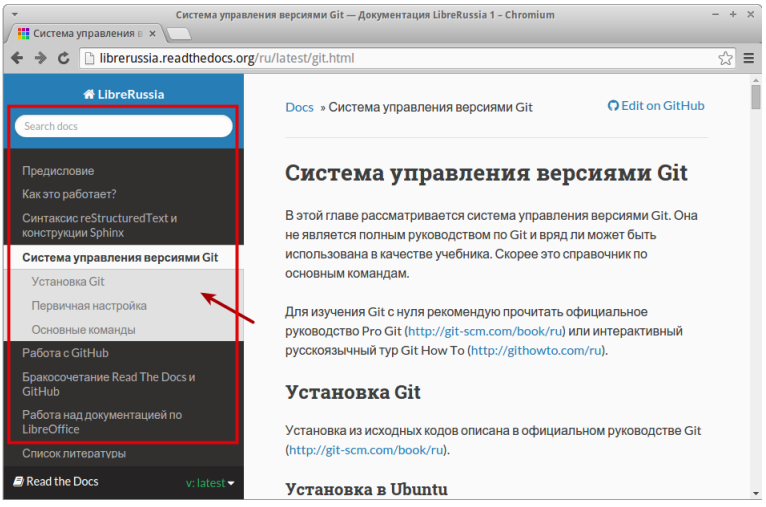
GitHub позволяет обезопасить процесс создания документации. С одной стороны, это дополнительная резервная копия, которая не будет утеряна. С другой, возможность в случае необходимости откатиться к предыдущим изменениям. Система веток позволяет одновременно работать над несколькими версиями руководства. С помощью функции Pull Requests все желающие могут подключиться к совместной работе и предлагать свои изменения. А редактор без лишних хлопот может безопасно принимать и контролировать все изменения в документации.

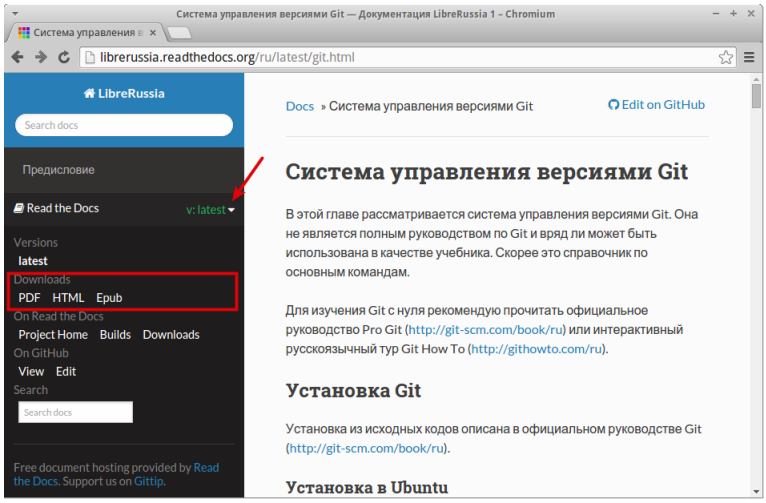
6.2 Работа с Read The Docs

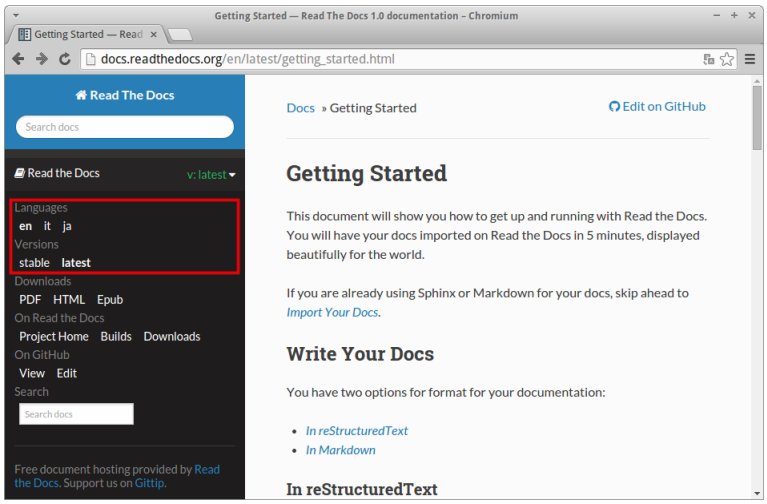
Read the Docs (https://readthedocs.org) служит для хранения и публикации документации, позволяет легко в ней ориентироваться и делает её доступной для полноценного поиска. Можно импортировать документацию из проекта, используя любую известную систему контроля версий: Mercurial, Git, Subversion и Bazaar.

Read The Docs поддерживает webhooks , так что документация может быть обновлена сразу после добавления в репозиторий нового кода. Также есть поддержка версий, поэтому можно собирать отдельные варианты (в том числе отдельные локализации) документации для тэгов и веток, которые есть в репозитории.

- Сайт Read the Docs
- Документация Read the Docs

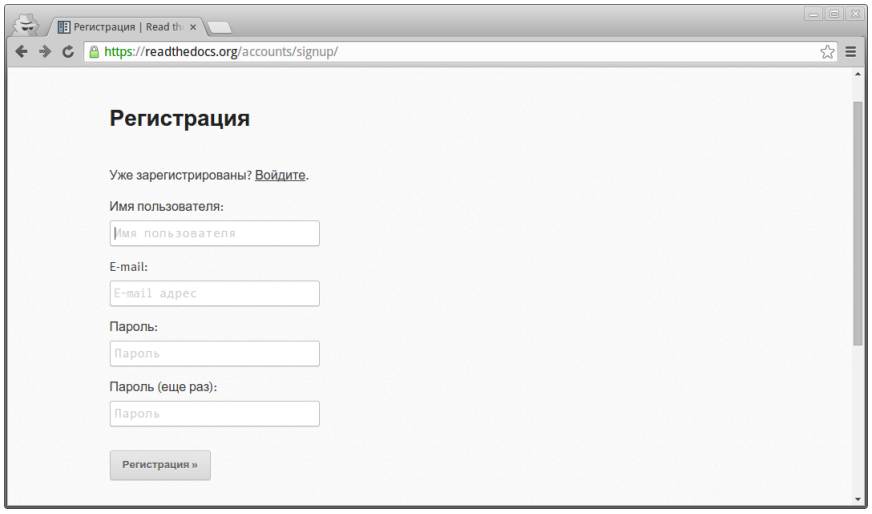



Система управления версиями Git — Документация LibreRussia 1 – Chromium
librerussia.readthedocs.org/ru/latest/git.html
LibreRussia
Search docs
Предисловие
Read the Docs
v: latest
Versions
latest
Downloads
PDF HTML Epub
On Read the Docs
Project Home Builds Downloads
On GitHub
View Edit
Search
Search docs
Free document hosting provided by Read the Docs. Support us on Gittip.
Docs » Система управления версиями Git
Edit on GitHub
Система управления версиями Git
В этой главе рассматривается система управления версиями Git. Она не является полным руководством по Git и вряд ли может быть использована в качестве учебника. Скорее это справочник по основным командам.
Для изучения Git с нуля рекомендую прочитать официальное руководство Pro Git (http://git-scm.com/book/ru) или интерактивный русскоязычный тур Git How To (http://githowto.com/ru).
Установка Git
Установка из исходных кодов описана в официальном руководстве Git (http://git-scm.com/book/ru).
Установка в Ubuntu


Getting Started — Read The Docs 1.0 documentation – Chromium
docs.readthedocs.org/en/latest/getting_started.html
Read The Docs
Search docs
Read the Docs
v: latest
Languages
en it ja
Versions
stable latest
Downloads
PDF HTML Epub
On Read the Docs
Project Home Builds Downloads
On GitHub
View Edit
Search
Search docs
Free document hosting provided by Read the Docs. Support us on Gittip.
Docs » Getting Started
Edit on GitHub
Getting Started
This document will show you how to get up and running with Read the Docs. You will have your docs imported on Read the Docs in 5 minutes, displayed beautifully for the world.
If you are already using Sphinx or Markdown for your docs, skip ahead to Import Your Docs.
Write Your Docs
You have two options for format for your documentation:
In reStructuredText
In Markdown
In reStructuredText

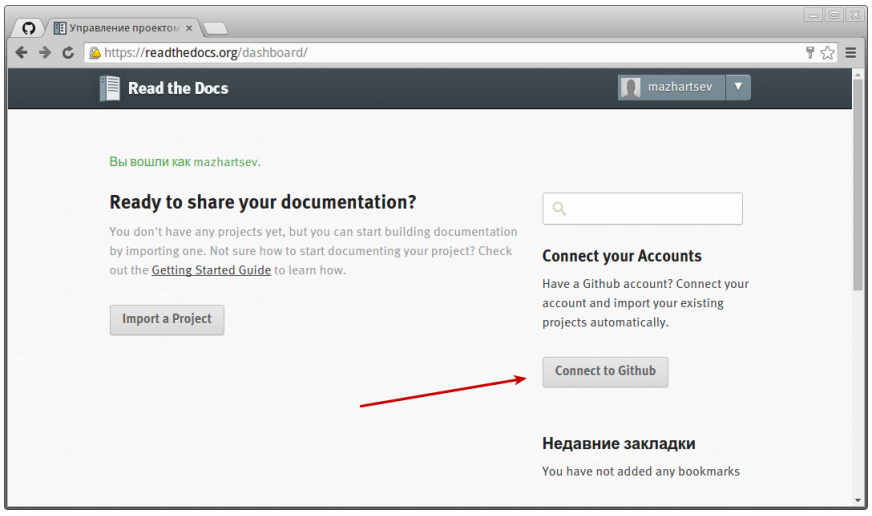
6.2.1 Регистрация

Read the Docs полностью бесплатен, регистрация на нём не занимает много времени.



6.2.2 Привязка к GitHub

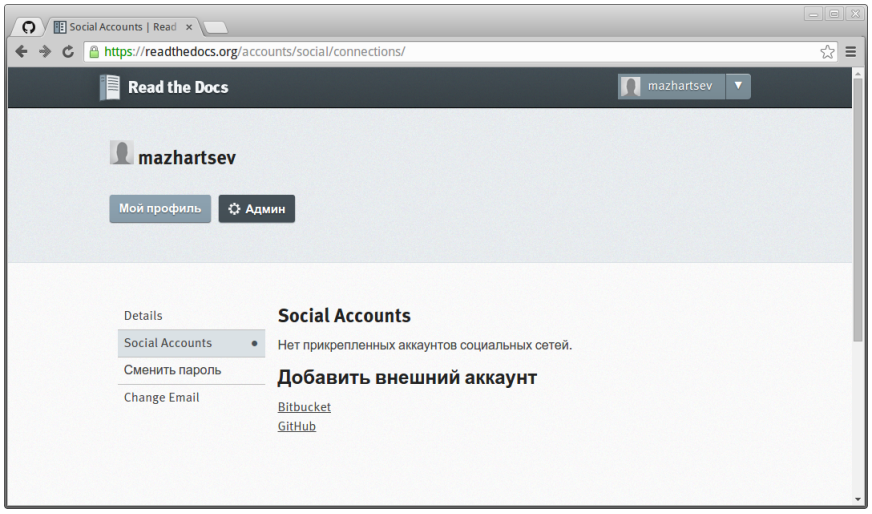
После регистрации можно привязать свой аккаунт на GitHub или Bitbucket.

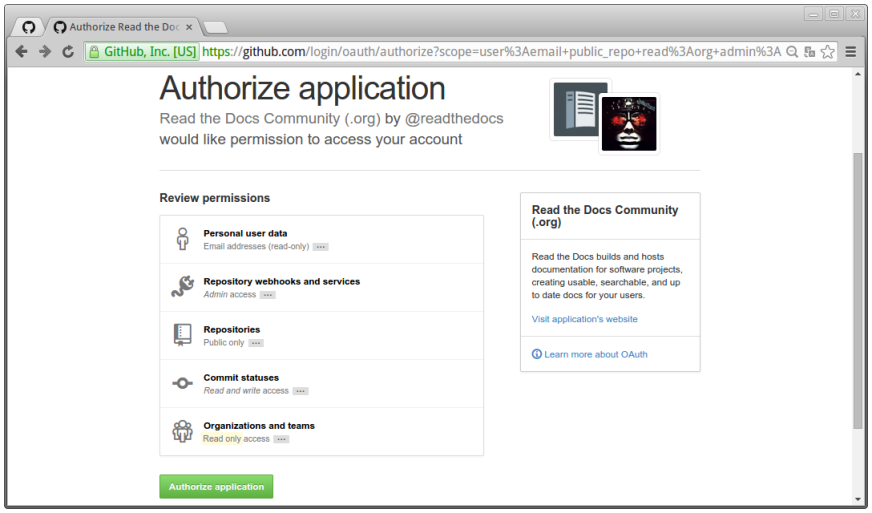


6.2.3 Создание проекта

Прежде чем выгрузить свой проект на Read the Docs, его необходимо создать и загрузить на GitHub. Структура репозитория должна содержать папку `docs`, в которой и будет находиться проект Sphinx (т.е. команду `sphinx-quickstart` нужно запускать в папке `docs`).

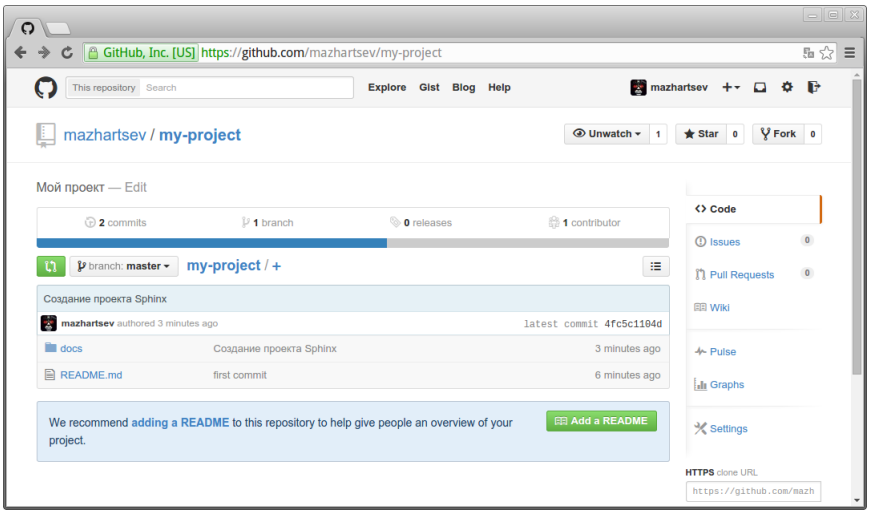
```
$ cd /path/to/project
$ mkdir docs
```


Social Accounts | Read
https://readthedocs.org/accounts/social/connections/
Read the Docs
mazhartsev
mazhartsev
Мой профиль
Admin
Details
Social Accounts
Сменить пароль
Change Email
Social Accounts
Нет прикрепленных аккаунтов социальных сетей.
Добавить внешний аккаунт
Bitbucket
GitHub


Authorize Read the Doc
GitHub, Inc. [US] https://github.com/login/oauth/authorize?scope=user%3Aemail+public_repo+read%3Aorg+admin%3A
Authorize application
Read the Docs Community (.org) by @readthedocs would like permission to access your account
Review permissions
Personal user data
Email addresses (read-only)
Repository webhooks and services
Admin access
Repositories
Public only
Commit statuses
Read and write access
Organizations and teams
Read only access
Authorize application
Read the Docs Community (.org)
Read the Docs builds and hosts documentation for software projects, creating usable, searchable, and up to date docs for your users.
Visit application's website
Learn more about OAuth

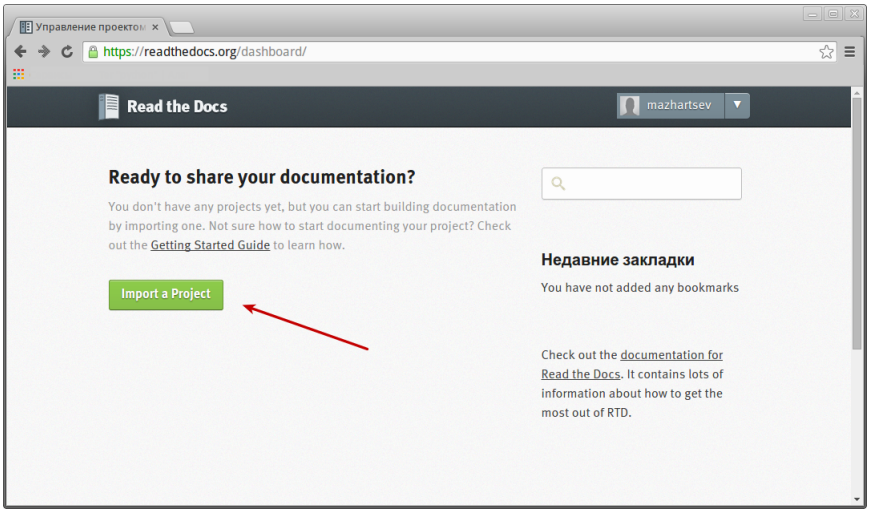
```
$ cd docs
$ sphinx-quickstart
$ make html
```

После создания Sphinx-проекта и генерации документации, загружаем репозиторий на GitHub.



6.2.4 Импорт проекта

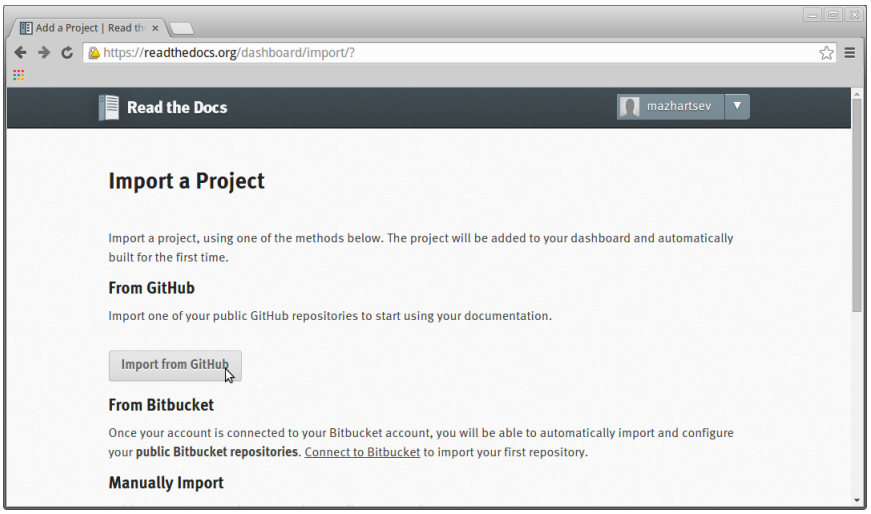
После создания проекта и выгрузки его на GitHub в панели управления Read the Docs нажимаем `Import a project`.

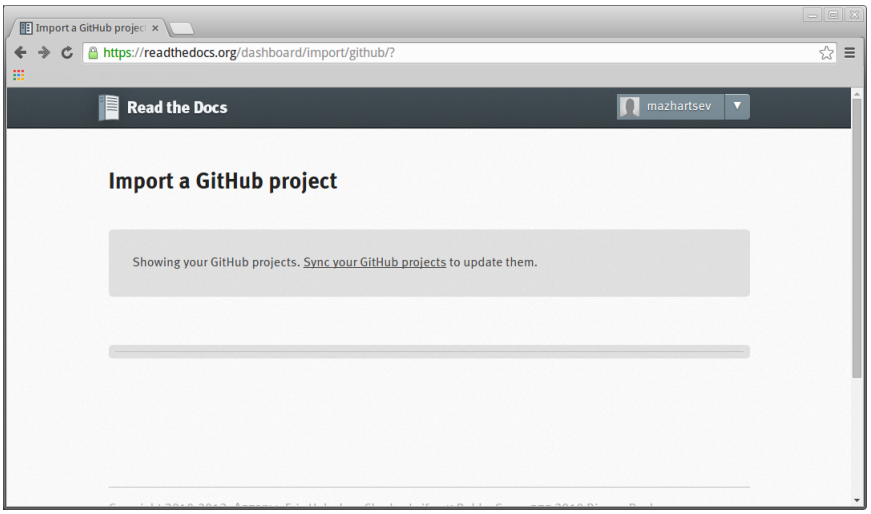


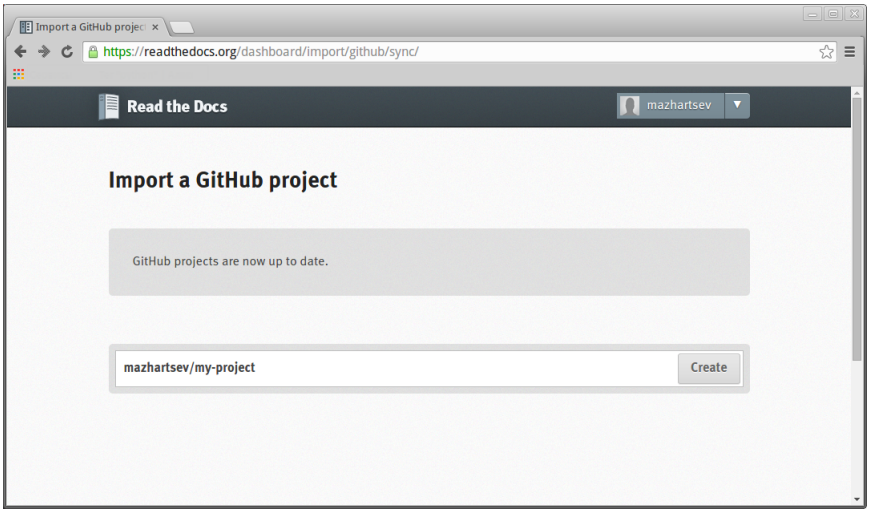
Синхронизируемся с GitHub и выбираем репозиторий, документация из которого будет опубликована на Read the Docs. Нажимаем кнопку `Create`.

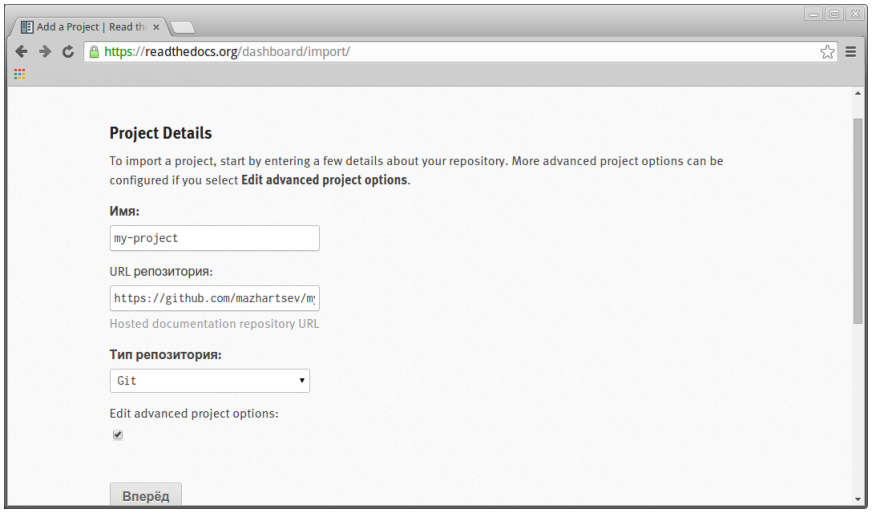
Далее будет предложено выбрать *Имя проекта* (это будет часть URL-адреса документации *Имя-проекта.readthedocs.org*), *Тип репозитория* и также можно установить галочку *Edit advanced project options* для включения расширенных настроек.

При включении расширенных настроек на следующей странице будет предложено дать расширенное описание проекта и выбрать локализацию. Выбор локализации позволяет создавать мультиязычную

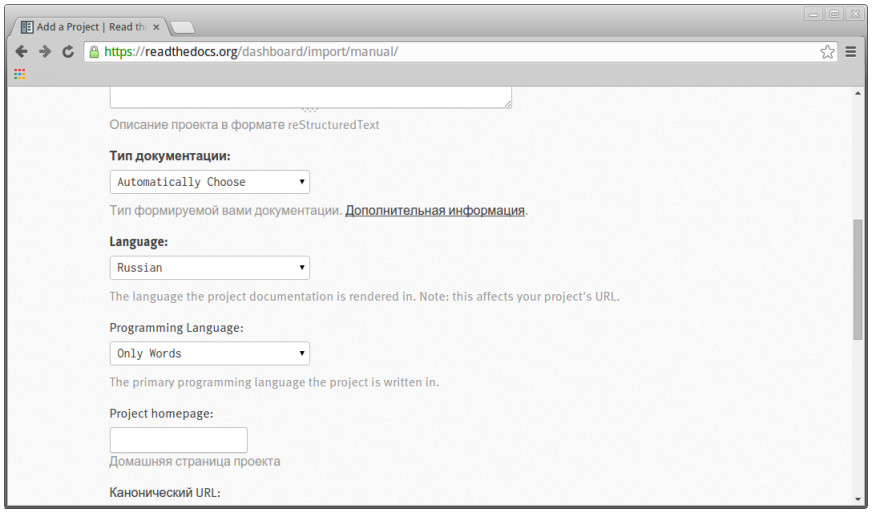

Add a Project | Read th
https://readthedocs.org/dashboard/import/?
Read the Docs
mazhartsev
Import a Project
Import a project, using one of the methods below. The project will be added to your dashboard and automatically built for the first time.
From GitHub
Import one of your public GitHub repositories to start using your documentation.
Import from GitHub
From Bitbucket
Once your account is connected to your Bitbucket account, you will be able to automatically import and configure your public Bitbucket repositories. Connect to Bitbucket to import your first repository.
Manually Import


Import a GitHub projec
https://readthedocs.org/dashboard/import/github/?
Read the Docs
mazhartsev
Import a GitHub project
Showing your GitHub projects. Sync your GitHub projects to update them.


Import a GitHub projec
https://readthedocs.org/dashboard/import/github/sync/
Read the Docs
mazhartsev
Import a GitHub project
GitHub projects are now up to date.
mazhartsev/my-project
Create



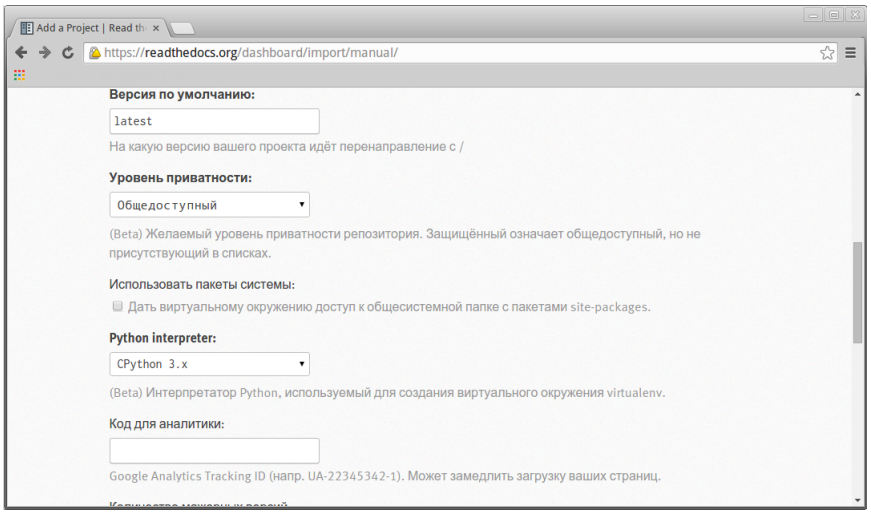
документацию.

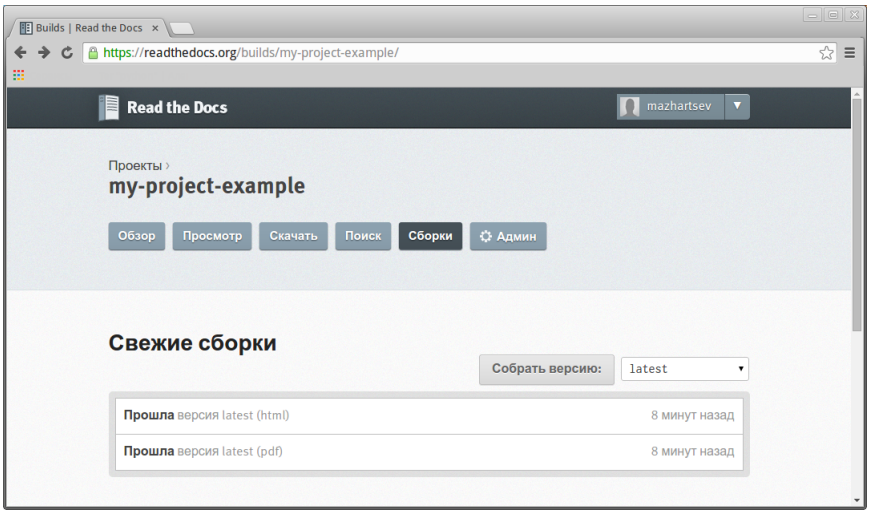


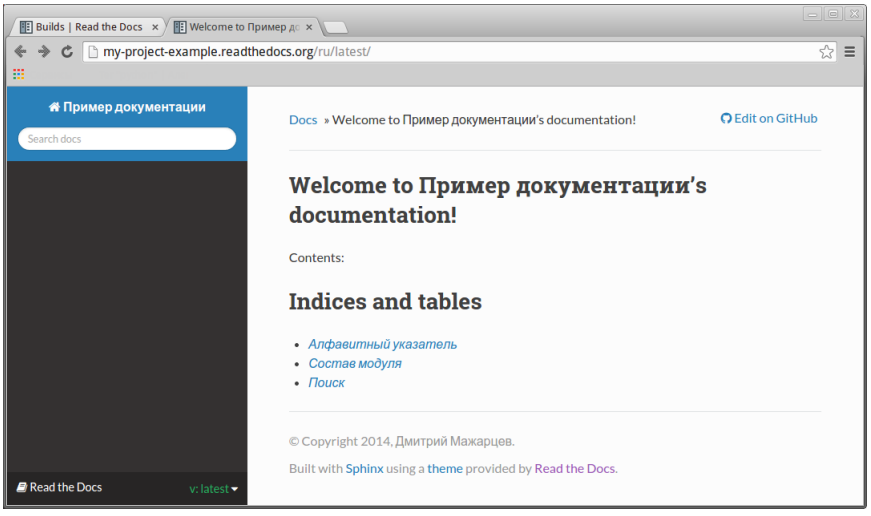
Дальше будет предложено выбрать ещё ряд настроек таких, как версия интерпретатора (я предпочитаю собирать 3-й версией, меньше проблем с кириллицей), ветку репозитория, на основе которой будет сгенерирована версия руководства, а также стандартную версию руководства, на которую будет происходить перенаправление по умолчанию.

После выбора всех настроек будет произведена первая сборка руководства.

Так выглядит стандартная тема оформления Read the Docs, при желании можно использовать любую другую (см. *Смена HTML-темы*):

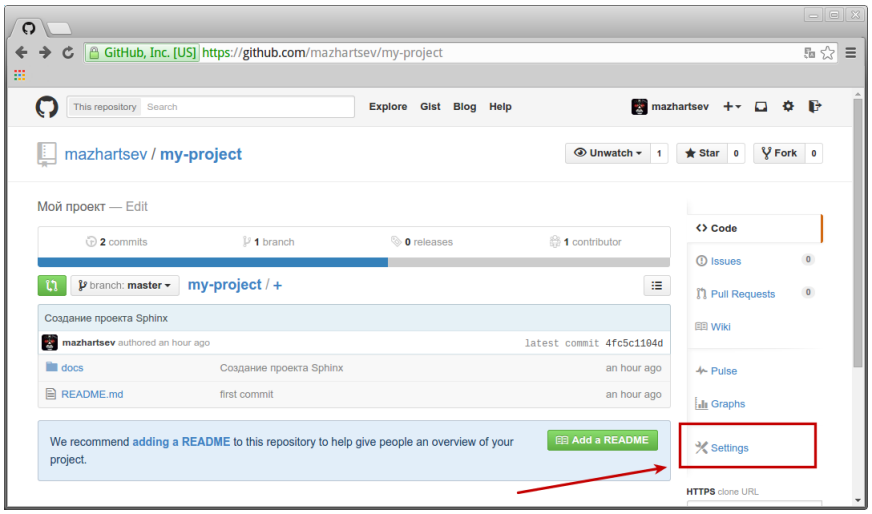

Add a Project | Read th
https://readthedocs.org/dashboard/import/manual/
Версия по умолчанию:
latest
На какую версию вашего проекта идёт перенаправление с /
Уровень приватности:
Общедоступный
(Beta) Желаемый уровень приватности репозитория. Защищённый означает общедоступный, но не присутствующий в списках.
Использовать пакеты системы:
Дать виртуальному окружению доступ к общесистемной папке с пакетами site-packages.
Python interpreter:
CPython 3.x
(Beta) Интерпретатор Python, используемый для создания виртуального окружения virtualenv.
Код для аналитики:
Google Analytics Tracking ID (напр. UA-22345342-1). Может замедлить загрузку ваших страниц.


Builds | Read the Docs
https://readthedocs.org/builds/my-project-example/
Read the Docs
mazhartsev
Проекты ›
my-project-example
Обзор
Просмотр
Скачать
Поиск
Сборки
Админ
Свежие сборки
Собрать версию:
latest
Прошла версия latest (html)
8 минут назад
Прошла версия latest (pdf)
8 минут назад

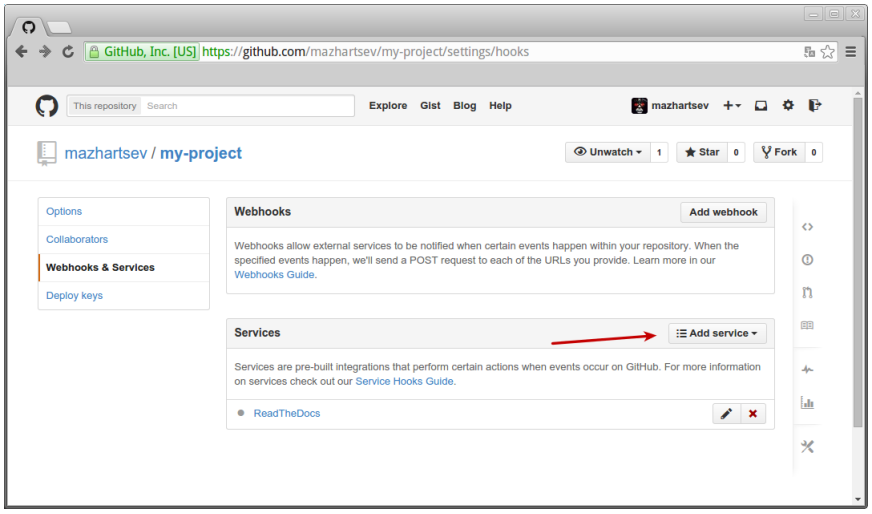

Builds | Read the Docs
Welcome to Пример д
my-project-example.readthedocs.org/ru/latest/
Пример документации
Search docs
Docs » Welcome to Пример документации's documentation!
Edit on GitHub
Welcome to Пример документации's documentation!
Contents:
Indices and tables
Алфавитный указатель
Состав модуля
Поиск
© Copyright 2014, Дмитрий Мажарцев.
Built with Sphinx using a theme provided by Read the Docs.
Read the Docs
v: latest

6.2.5 Автоматическая публикация

Для включения автоматической публикации документации на Read The Docs при обновлении репозитория GitHub, перейдите в репозиторий проекта на GitHub в раздел `Settings`.



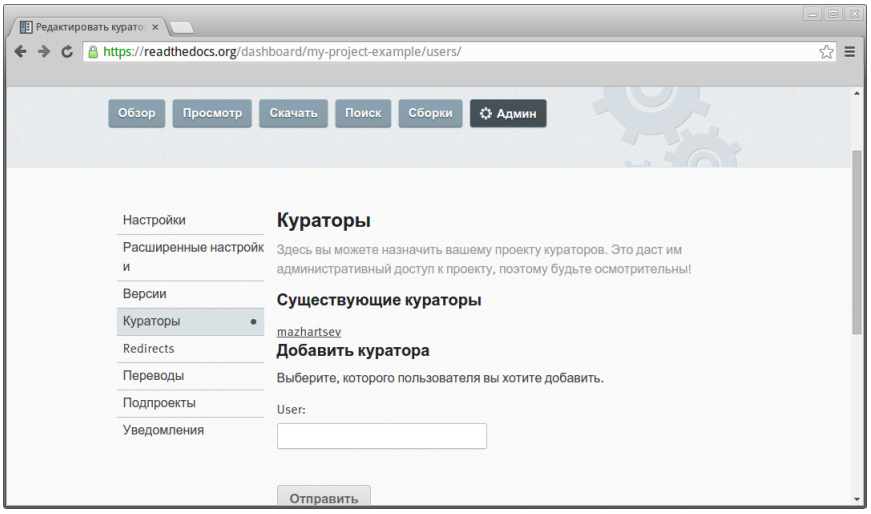
Выберите пункт `Webhooks & Services` и добавьте сервис Read the Docs.



Не путайте автоматическую публикацию с автоматической сборкой (генерацией) самой документации Sphinx, об этом подробнее смотрите раздел *Автоматическая сборка*.

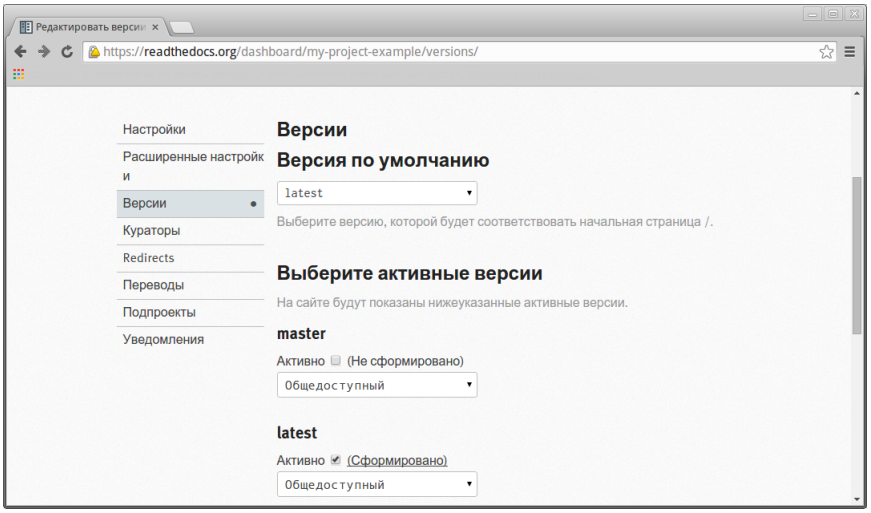
6.2.6 Настройка

В панели управления проектом на Read the Docs в разделе `Admin` можно изменить настройки проекта, добавить новые версии документации, назначить кураторов и т.д.



6.2.7 Несколько версий документации

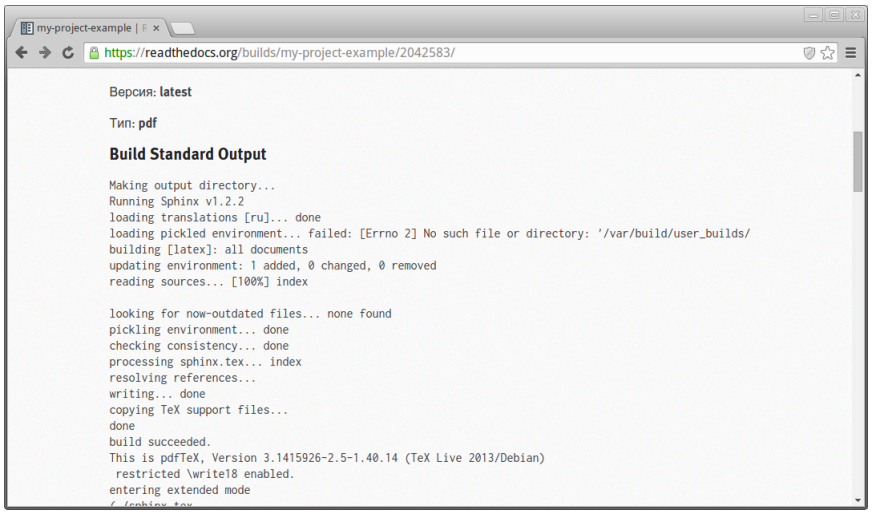
Версии документации создаются на основе веток Git-репозитория, подробнее смотрите *Ветвление*.



6.2.8 Ошибки сборки

В некоторых случаях могут возникать ошибки при сборке документации. Причины ошибок могут быть различны. Стандартный вывод интерпретатора содержит подробную информацию обо всех ошибках.

Наиболее часто возникаю ошибки при сборке PDF. При сборке используется генерация в LaTeX. Стандартные настройки шаблона LaTeX, используемого Read the Docs, не поддерживают кириллицу. Подробнее смотрите раздел *Ошибки при сборке PDF на Read The Docs*.


my-project-example
https://readthedocs.org/builds/my-project-example/2042583/
Версия: latest
Тип: pdf
Build Standard Output
Making output directory...
Running Sphinx v1.2.2
loading translations [ru]... done
loading pickled environment... failed: [Errno 2] No such file or directory: '/var/build/user_builds/
building [latex]: all documents
updating environment: 1 added, 0 changed, 0 removed
reading sources... [100%] index
looking for now-outdated files... none found
pickling environment... done
checking consistency... done
processing sphinx.tex... index
resolving references...
writing... done
copying TeX support files...
done
build succeeded.
This is pdfTeX, Version 3.1415926-2.5-1.40.14 (TeX Live 2013/Debian)
restricted \write18 enabled.
entering extended mode

ГЛАВА 7

Известные проблемы

В данной главе приведены известные проблемы, с которыми я столкнулся за время эксплуатации Sphinx.

7.1 Кириллические символы в алфавитном указателе

Кириллические элементы алфавитного указателя не сортируются по алфавиту и попадают в раздел `символы`. Решение проблемы пока мною не найдено.

7.2 Перекрестные ссылки в LaTeX

Подписи к таблицам и рисункам в HTML отображаются без названия категории «Таблица» или «Рисунок», но в LaTeX-документах к подписям автоматически добавляется и категория с номером. Нумерация также осуществляется автоматически.

При создании перекрестных ссылок на иллюстрации и таблицы ссылка принимает название этих объектов, что не очень удобно если названия длинные. Обычно в документации принято делать ссылки в формате «см. Рисунок N».

В LaTeX-документах ссылки на рисунки и таблицы также преобразуются в названия, а не в категории.

Способ обхода этой проблемы описан здесь. У меня не было времени его протестировать. Буду благодарен, если кто-нибудь поделится своим опытом на этот счёт.

7.3 Масштабирование изображений в LaTeX

Столкнулся с проблемой масштабирования изображений в LaTeX. При использовании директивы `.. image::` с параметрами `:height:` или `:width` масштабирование изображений в LaTeX не происходило. При использовании директивы `.. figure::` с параметром `:scale:` все работает как надо, изображения масштабируются.

7.4 Ошибки при сборке PDF на Read The Docs

При сборке PDF используется генерация в LaTeX. Стандартные настройки шаблона LaTeX, используемого Read the Docs, не поддерживают кириллицу. Чтобы включить поддержку кириллицы, необходимо поправить преамбулу LaTeX в файле настроек Sphinx-проекта `conf.py`. Подробнее смотрите пример *Преамбула*.

7.5 Проблемы с отображением листингов и таблиц в ePub

Многие проблемы с отображением тех или иных фрагментов текста в ePub связаны непосредственно с просмотрщиками электронных книг. Например, плагин к браузеру Firefox для просмотра ePub отлично отображает все элементы. На Android в программе CoolReader некорректно отображаются листинги. В программе FBreader листинги отображаются хорошо, а вот таблицы сбиваются.

7.6 Некорректно отображаются формулы на Read The Docs

У меня возникли проблемы с отображением формул на Read The Docs при использовании расширения `sphinx.ext.mathjax`, пересобрав документацию с расширением `sphinx.ext.pngmath` все стало в порядке. Подробнее смотрите раздел *Режим отображения формул*.

Также проблема с отображением формул может появиться при использовании символов греческого алфавита (, и т.д.) внутри формулы. Так как Sphinx использует для набора синтаксис LaTeX, то набирать данные символы надо следующим образом: `\alpha`, `\lambda` и т.д. (смотрите Перечень команд LaTeX для набора греческих букв).

```
.. math::

   \alpha_t(i) = P(O_1, O_2, ... O_t, q_t = S_i \lambda)
```

Результат:

$$\alpha_t(i) = P(O_1, O_2, \ldots O_t, q_t = S_i \lambda)$$

ГЛАВА 8

Часто задаваемые вопросы

8.1 Как добавить подпись к рисунку?

Смотрите раздел *Изображения и иллюстрации*.

8.2 Как добавить подпись к таблице?

Смотрите раздел *Таблицы*.

8.3 Как сделать ссылку на главу или раздел?

Ссылки на разделы внутри одного документа делаются путем указания в обратных кавычках названия этого раздела, например, *Как сделать ссылку на главу или раздел?*.

```
... например, `Как сделать ссылку на главу или раздел?`_
```

Для создания ссылок между разными документами смотрите раздел *Перекрестные ссылки*.

8.4 Как вставить содержание в разделе (документе)?

Смотрите раздел *Содержание*.

8.5 Как сделать перекрестную ссылку на таблицу или рисунок?

Смотрите раздел *Перекрестные ссылки*.

8.6 Как сделать перекрестную ссылку в виде категории?

Смотрите раздел *Перекрестные ссылки в LaTeX*.

8.7 Вставка графиков

Для вставки графиков используются дополнительные расширения, список которых приведен на странице Sphinx Extensions официальной документации Sphinx.

Также смотрите раздел *Подключение расширений*.

8.8 Как настроить автоматическую генерацию документации

Подробнее смотрите раздел *Автоматическая сборка* и *Автоматическая публикация*.

8.9 Как сделать принудительный разрыв строки?

Возникла у меня необходимость сделать принудительный разрыв строки в HTML. Для этой цели используется автозамена и директива `raw`:

```
.. |br| raw:: html

   <br />

Длинная строка с разрывом в |br| этом месте.
```

Длинная строка с разрывом в этом месте.

8.10 Есть более быстрый способ делать таблицы?

Смотрите статью reStructuredText (ReST): Быстрый способ ввода таблиц

ГЛАВА 9

Соглашения

Данный раздел касается тех проектов, в которых я задействован. Здесь приведены некоторые общие соглашения и рекомендации относительно оформления документации.

1. Для обозначения пунктов меню используйте команды `menuselection` и `guilabel`.

```
:menuselection:`&Файл --> &Открыть`
:guilabel:`&Открыть`
```

- *Файл → Открыть*
- *Открыть*

Символ `&` устанавливает в зависити от темы HTML следующему за ним символу нижнее подчеркивание. Обязательно сверяйтесь с подчеркиваниями в пунктах меню программы.

2. Для примеров исходного кода на различных языках программирования, используйте конструкцию `.. code-block::` (см. *Примеры исходного кода с подсветкой синтаксиса*).
   - Не используйте нумерацию строк в листингах короче 3-х строк.
   - Для листингов не примеров исходного кода на языках программирования допускается использовать более простую конструкцию `::` (см. *Листинги (исходный код)*). При этом лучше использовать более явный стиль команды `::`, когда она располагается на новой строке:

```
Посмотрим на исходный код:
::

    Пример исходного кода
```

3. Пользуйтесь нумерованными сносками (см. *Сноски*).
4. Вставляйте изображения с помощью директивы `.. figure::` (вместо `.. image::`), для масштабирования используйте параметр `:scale:`, а не `:height:` и `:width` (см. *Масштабирование изображений в LaTeX*).
5. Добавляйте подписи к таблицам и рисункам (см. *Изображения и иллюстрации* и *Таблицы*).

6. Делайте осмысленные комментарии к коммитам.
7. В любой непонятной ситуации задавайте вопросы редактору, не импровизируйте.

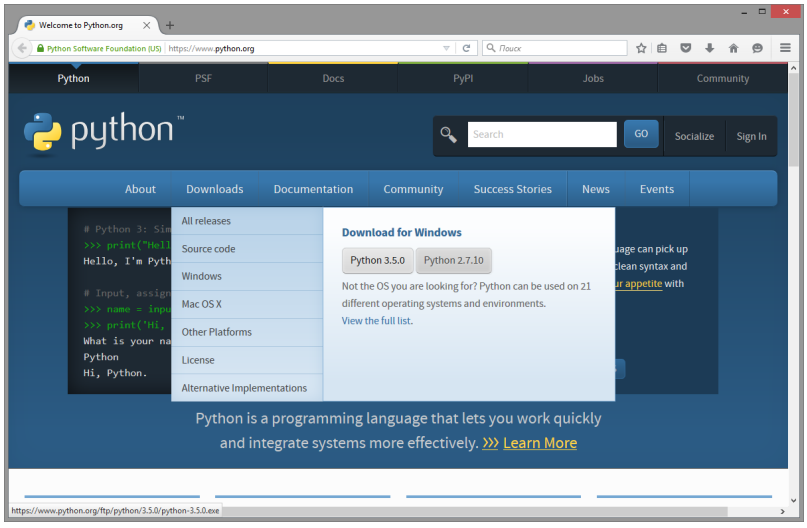
ГЛАВА 10

Приложение I – Установка и настройка Python Sphinx в ОС Windows

10.1 Установка

В данном разделе описана установка и настройка Python Sphinx в операционной системе Windows.

1. Скачайте установочный файл интерпретатора языка программирования Python версии `3.5` – https://www.python.org.

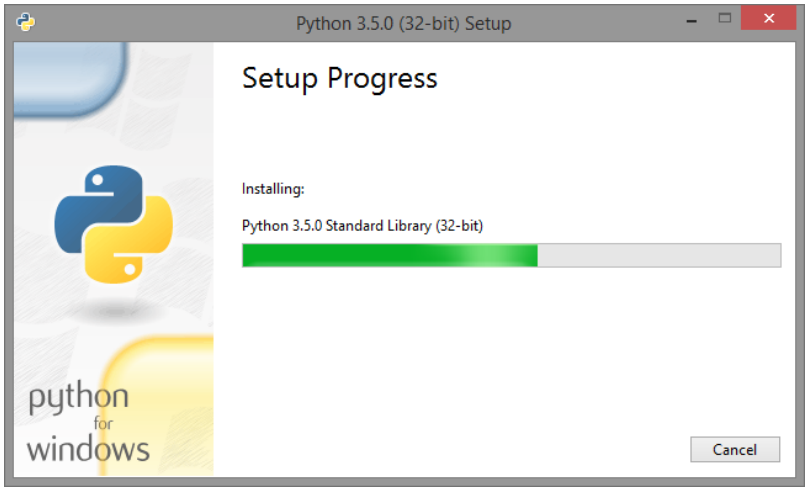


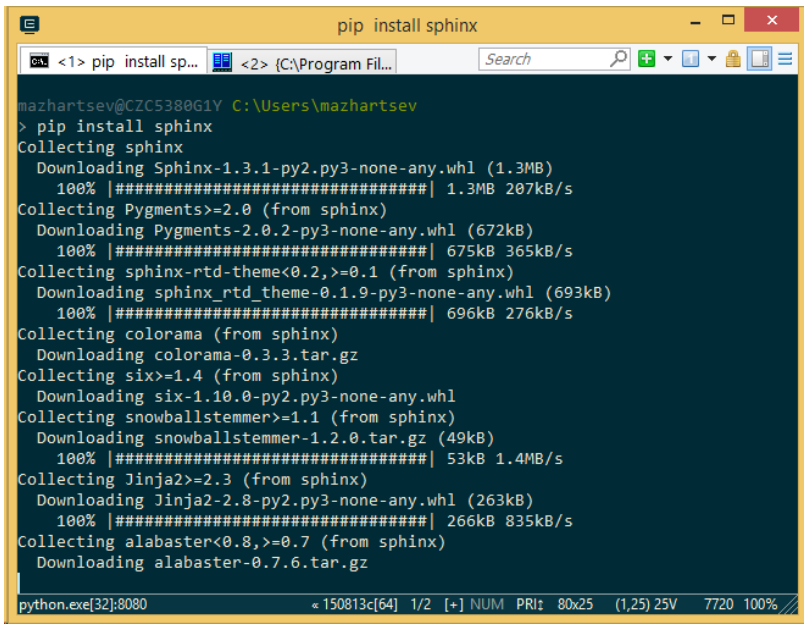
2. Запустите файл установки. На первом шаге поставьте галочку напротив **Add Python 3.5 to PATH** и нажмите **Install Now**.

3. Дождитесь завершения установки интерпретатора.

4. После установки интерпретатора необходимо установить модуль Python Sphinx, для этого откройте командную строку *Пуск* > *cmd* и выполните команду `pip install sphinx`.


Python 3.5.0 (32-bit) Setup
Install Python 3.5.0 (32-bit)
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.
Install Now
C:\Users\mazhartsev\AppData\Local\Programs\Python\Python35-32
Includes IDLE, pip and documentation
Creates shortcuts and file associations
Customize installation
Choose location and features
python for windows
Install launcher for all users (recommended)
Add Python 3.5 to PATH
Cancel


Python 3.5.0 (32-bit) Setup
Setup Progress
Installing:
Python 3.5.0 Standard Library (32-bit)
python for windows
Cancel


pip install sphinx
<1> pip install sp...
<2> {C:\Program Fil...
Search
mazhartsev@CZC5380G1Y C:\Users\mazhartsev
> pip install sphinx
Collecting sphinx
Downloading Sphinx-1.3.1-py2.py3-none-any.whl (1.3MB)
100% |################################| 1.3MB 207kB/s
Collecting Pygments>=2.0 (from sphinx)
Downloading Pygments-2.0.2-py3-none-any.whl (672kB)
100% |################################| 675kB 365kB/s
Collecting sphinx-rtd-theme<0.2,>=0.1 (from sphinx)
Downloading sphinx_rtd_theme-0.1.9-py3-none-any.whl (693kB)
100% |################################| 696kB 276kB/s
Collecting colorama (from sphinx)
Downloading colorama-0.3.3.tar.gz
Collecting six>=1.4 (from sphinx)
Downloading six-1.10.0-py2.py3-none-any.whl
Collecting snowballstemmer>=1.1 (from sphinx)
Downloading snowballstemmer-1.2.0.tar.gz (49kB)
100% |################################| 53kB 1.4MB/s
Collecting Jinja2>=2.3 (from sphinx)
Downloading Jinja2-2.8-py2.py3-none-any.whl (263kB)
100% |################################| 266kB 835kB/s
Collecting alabaster<0.8,>=0.7 (from sphinx)
Downloading alabaster-0.7.6.tar.gz
python.exe[32]:8080

На этом установка Python Sphinx закончена, можно переходить к сборке документации. Сборка осуществляется командой:

```
sphinx-build -b html папка\с\исходными_файлами папка\для\сконвертированных_файлов
```

Например:

```
sphinx-build -b html source build\html
```

10.2 Сборка документации

Перед началом сборки создайте папку `docs`. Для примера создадим на рабочем столе папку `Руководство`, а в ней папку `docs`. Затем откройте командную строку *Пуск* > *cmd* и перейдите в папку с руководством, выполнив команду:

```
cd Desktop\Руководство\docs
```

Примечание: Подробнее о работе с командной строкой Windows смотрите Руководство по командной строке Windows.

Выполните команду `sphinx-quickstart`. Программа задаст ряд вопросов. Все настройки можно будет позже изменить в файле `conf.py`.

```
> Корневой каталог документации. По умолчанию текущий каталог.
> Root path for the documentation [.]: Enter

> Сделать ли раздельные папки исходников и готовых страниц - Да
> Separate source and build directories (y/n) [n]: y

> Префикс для директорий с шаблонами и статическими файлами.
> Name prefix for templates and static dir [_]:

> Название проекта. Для начала лучше вводить на латинице.
> Project name:

> Имя автора/авторов. Для начала лучше вводить на латинице.
> Author name(s):

> Версия проекта
> Project version:

> Номер релиза проекта
> Project release [1]:

> Расширение исходного файла. По умолчанию .rst.
> Source file suffix [.rst]:

> Имя мастер-документа. По умолчанию index.rst.
> Name of your master document (without suffix) [index]:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
> Генерировать ePub версию документации?
> Do you want to use the epub builder (y/n) [n]:

> Автоматически вставлять docstrings из модулей
> autodoc: automatically insert docstrings from modules (y/n) [n]:

>
> doctest: automatically test code snippets in doctest blocks (y/n) [n]:

>
> intersphinx: link between Sphinx documentation of different projects (y/n) [n]:

>
> todo: write "todo" entries that can be shown or hidden on build (y/n) [n]:

>
> coverage: checks for documentation coverage (y/n) [n]:

> Использовать модуль pngmath для вставки формул в формате png
> pngmath: include math, rendered as PNG images (y/n) [n]:

> Использовать модуль mathjax для вставки формул в формате MathJax
> mathjax: include math, rendered in the browser by MathJax (y/n) [n]: y

>
> ifconfig: conditional inclusion of content based on config values (y/n) [n]:

> Влючить ссылки на исходный код в документации
> viewcode: include links to the source code of documented Python objects (y/n) [n]:

> Создать Makefile - да
> Create Makefile? (y/n) [y]:

> Сделать ли файл .bat, - да
> Create Windows command file? (Y/n) [y]: y
```

После выполнения стартового скрипта в папке `docs` появится следующая структура:

```
docs
 build
|        html
|                ...
|                index.html
 source
|        _templates
|        _static
|        conf.py
|        index.rst
 Makefile
```

`Makefile` — содержит инструкции для генерации результирующего документа командой make.

`build` — директория, в которую будут помещены файлы в определенном формате после того, как будет

запущен процесс их генерации.

`source` — директория, в которой располагаются исходные файлы.

`index.rst` — это корень проекта. Он соединяет документацию воедино, если она разделена на несколько файлов.

`_static` — в эту директорию помещаются все файлы, не являющиеся исходным кодом (например, изображения). Позже создаются связи этих файлов с директорией build.

`conf.py` — содержит конфигурационные параметры Sphinx, включая те, которые были выбраны при запуске sphinx-quickstart в окне терминала.

Чтобы выполнить сборку документации, перейдите в папку `docs` и выполните команду:

```
sphinx-build -b html source build\html
```

Будет выполнена сборка документации, в терминале появится информация о ходе сборки:

```
PS C:\Users\mazhartsev\Desktop\Руководство\docs> sphinx-build -b html source build\html
Running Sphinx v1.3.1
making output directory...
loading translations [ru]... done
loading pickled environment... not yet created
building [mo]: targets for 0 po files that are out of date
building [html]: targets for 1 source files that are out of date
updating environment: 1 added, 0 changed, 0 removed
reading sources... [100%] index
looking for now-outdated files... none found
pickling environment... done
checking consistency... done
preparing documents... done
writing output... [100%] index
generating indices... genindex
writing additional pages... search
copying static files... done
copying extra files... done
dumping search index in Russian (code: ru) ... done
dumping object inventory... done
build succeeded.
```

Собранные html-файлы появятся в папаке `build\html`.

Совет: Для удобства можно создать bat-файл, содержащий строку `sphinx-build -b html source build\html`. Подробнее смотрите Руководство по командной строке Windows.

Далее смотрите пункт Файл index в разделе *Генератор документации Sphinx*.

ГЛАВА 11

Русскоязычное сообщество LibreOffice

Последние несколько лет я являюсь участником русскоязычного сообщества LibreOffice. Поскольку именно эта моя деятельность послужила мотивацией для изучения Sphinx и написания данного руководства, то я просто обязан вставить раздел с ссылками на русскоязычные ресурсы сообщества.

11.1 Новости

- Группа ВКонтакте: http://vk.com/libreoffice
- Твиттер: http://twitter.com/LibreOffice_ru
- Facebook: https://www.facebook.com/ru.libreoffice.org

11.2 Поддержка

Форум поддержки пользователей LibreOffice и Apache OpenOffice: http://forumooo.ru

Форум ведет свою историю со времен OpenOffice.org и накопил огромную базу с решениями многих проблем. На форуме можно задать интересующие вас вопросы, а также принять участие в деятельности русскоязычного сообщества LibreOffice.

Также доступен IRC-канал #libreoffice-ru в сети FreeNode:

- https://webchat.freenode.net/?channels=#libreoffice-ru

11.3 Обучение

Документация и часто задаваемые вопросы по LibreOffice:

- https://wiki.documentfoundation.org/Documentation/ru

11.4 Независимые блоги

- Информатика в экономике и управлении: http://infineconomics.blogspot.ru
- Блог про LibreOffice: Советы, трюки, хитрости, инструкции, руководства: http://librerussia.blogspot.ru

11.5 Списки почтовой рассылки LibreOffice

Подписаться на официальную почтовую рассылку можно на странице официальной «Вики» LibreOffice: https://wiki.documentfoundation.org/Local_Mailing_Lists/ru

- genindex

Литература

[CIT2002] Это цитата (как часто используемая в журналах).

[CIT2003] Код вставки этой цитаты `.. [CIT2003]` размещен в самом конце *.rst* файла.

Алфавитный указатель

Символы

G

P