

---

# **Tutorial pubblicazione Read the Docs su DocsItalia**

*Release version: latest*

**comune Palermo**

**28 giu 2018**



|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Che cosa è questo tutorial</b>                                | <b>1</b>  |
| <b>2</b> | <b>A chi si rivolge il tutorial</b>                              | <b>3</b>  |
| <b>3</b> | <b>Un tutorial a cura di</b>                                     | <b>5</b>  |
| <b>4</b> | <b>Perché questo tutorial</b>                                    | <b>7</b>  |
| <b>5</b> | <b>Un tutorial di sperimentazione</b>                            | <b>9</b>  |
| <b>6</b> | <b>Attività propedeutica alla pubblicazione su Read the Docs</b> | <b>11</b> |
| <b>7</b> | <b>Licenza</b>   | <b>13</b> |
| 7.1      | Contenuti del Tutorial . . . . .                                 | 13        |



---

Che cosa è questo tutorial

---



Fig. 1.1: immagine rilasciata con licenza CC0

Questo tutorial ha l'obiettivo di spiegare due modalità per la creazione di un Documento da visualizzare tramite la piattaforma [Read the Docs](#), secondo le specifiche della [Guida «Docs Italia»](#) del Team Trasformazione Digitale dell'AgID (Agenzia per l'Italia Digitale).

**Read The Docs** è un servizio online gratuito che semplifica la gestione della documentazione relativa ad una pubblicazione, e consente di automatizzare:

- la creazione dei documenti,
- il controllo delle versioni della pubblicazione che possono essere aggiornate nel tempo,
- l'hosting dei singoli documenti pubblicati.



Fig. 1.2: immagine dell'home page delle Linee guida open data del Comune di Palermo pubblicate su «Read the Docs»

### A chi si rivolge il tutorial

---

Il tutorial si rivolge principalmente ai dipendenti e dirigenti di ogni tipologia di Pubblica Amministrazione in Italia, soprattutto a chi è responsabile della pubblicazione di documenti pubblici online, regolamenti, manuali e a chi si occupa di comunicazione istituzionale.

Non è necessario essere sviluppatori per usare le procedure che spieghiamo in questo tutorial e infatti abbiamo strutturato questo documento in più parti:

---

**Importante:**

- una destinata a chi non è molto addentro ai linguaggi di programmazione e ai sistemi di versionamento (curata da **Ciro Spataro**) = **il capitolo 1**;
- un'altra per chi ha maggiori competenze digitali (a cura di **Pablo Persico**) = **il capitolo 3**;
- infine una in comune alle precedenti, abbastanza semplice, che consiste nella creazione del nome del progetto da pubblicare sulla piattaforma di Read the Docs, che si svolge solo dopo aver terminato la compilazione del progetto su Github = **il capitolo 2**.

---

Ciro Spataro e Pablo Persico hanno seguito due strade diverse (capitolo 1 e capitolo 3) per raggiungere lo stesso obiettivo di pubblicazione di documenti su Read the Docs. In tal maniera verrà coperta un'ampia rappresentanza (in fatto di competenze digitali) dei dipendenti e dirigenti pubblici.



## CAPITOLO 3

---

Un tutorial a cura di

---

- **Ciro Spataro** (Comune di Palermo, OpenDataSicilia), email: [c.spataro@comune.palermo.it](mailto:c.spataro@comune.palermo.it)
- **Pablo Persico** (Città Metropolitana di Napoli, OpenDataSicilia), email: [ppersico@cittametropolitana.na.it](mailto:ppersico@cittametropolitana.na.it)
- **Andrea Borruso** (OnData, OpenDataSicilia), supervisione e «spingitore di tutorial», email: [andrea.borruso@ondata.it](mailto:andrea.borruso@ondata.it)





---

### Perché questo tutorial

---

Il Team Trasformazione Digitale dell'AGID ha creato «Guida a Docs Italia» per consentire e agevolare le pubbliche amministrazioni ad usare il formato di «Read The Docs» al fine di pubblicare atti, documenti del tipo: Decreti, Norme, Regolamenti, Linee Guida, Piani, Deliberazioni, Determinazioni, ecc.

Dopo questo lavoro del Team Trasformazione Digitale, siamo partiti in 3 Pubbliche Amministrazioni a creare documenti su Read the Docs e siamo sicuri che questo formato sarà adottato da tanti altri soggetti che lavorano in altre PA. Vogliamo fare da apripista, stimolo e anche supporto. Siamo a disposizione per aiutarvi, con piacere, partendo proprio da questo tutorial.

Questo tutorial è un lavoro integrativo a quello svolto dal Team con la Guida «Docs Italia». Un esperimento svolto per attuare i contenuti della Guida e verificare, passo passo, la comprensione delle procedure indicate.

---

#### **OBIETTIVO**

**L'obiettivo prioritario** è far in modo da diminuire le pubblicazioni online in formato PDF, che oggi rappresenta il formato più usato nei siti web della PA, ma che non consente una leggibilità nei dispositivi mobili (viene molto difficile e non è affatto pratico) e che non permette un'agevole ricerca di parole nel testo. Il formato Read the Docs, invece, soddisfa i suddetti requisiti.

---

#### **DAL READ THE DOCS ALTRI FORMATI**

Dal formato Read the Docs è possibile convertire, in maniera istantanea, il documento nei formati **PDF**, **HTML** ed **EPUB** (che è *uno standard aperto specifico per la pubblicazione di libri digitali -eBook- basato su XML*).

Inoltre usare «Read the Docs» consente un facile riuso dei contenuti da parte della società e di quelle pubbliche amministrazioni che manifestano interesse a questo formato di pubblicazione.



---

### Un tutorial di sperimentazione

---

Questo lavoro vuole essere soprattutto sperimentazione sulle modalità con le quali si fanno cose negli uffici pubblici, pensando a migliorare la fruizione dei documenti online.

Un punto di partenza **aperto a contributi esterni**, un documento che possa svilupparsi anche grazie alle esperienze di soggetti di altre PA che vorranno adottare Read The Docs come strumento per diffondere cultura della documentazione della PA anche con finalità di riutilizzo e condivisione. Per questo motivo ad ogni pagina del tutorial abbiamo pensato di inserire una sezione per ricevere i commenti, grazie all'uso del servizio fornito da [Disqus](#).



---

### Attività propedeutica alla pubblicazione su Read the Docs

---

- avere un account [GitHub](#);
- avere un account [Read the Docs](#).

La [Guida di Docs italia](#) consiglia la creazione di una organizzazione su [GitHub](#).





Quest'opera è distribuita con Licenza CC BY SA 4.0 - Creative Commons, Attribuzione, Condividi allo stesso modo, Versione 4.0 Internazionale.

---

## 7.1 Contenuti del Tutorial

### 7.1.1 Capitolo 1 - Caricare il progetto editoriale su Github (lavorando online su Github)

(Modalità per lavorare su Github online, adatto per chi ha poca familiarità con gli strumenti di controllo versione).

#### Si inizia con lo «starter kit»

Questa <http://guida-docs-italia.readthedocs.io/it/latest/index/starter-kit.html> è la versione che dovremmo usare come modello di base, proposto dal Team Trasformazione Digitale, per riprodurre il nostro documento su Read the Docs.

Cliccando su [scarica lo Starter kit](https://github.com/italia/docs-italia-starter-kit) si viene rimandati al repository del progetto sulla piattaforma Github, cioè qui <https://github.com/italia/docs-italia-starter-kit>.

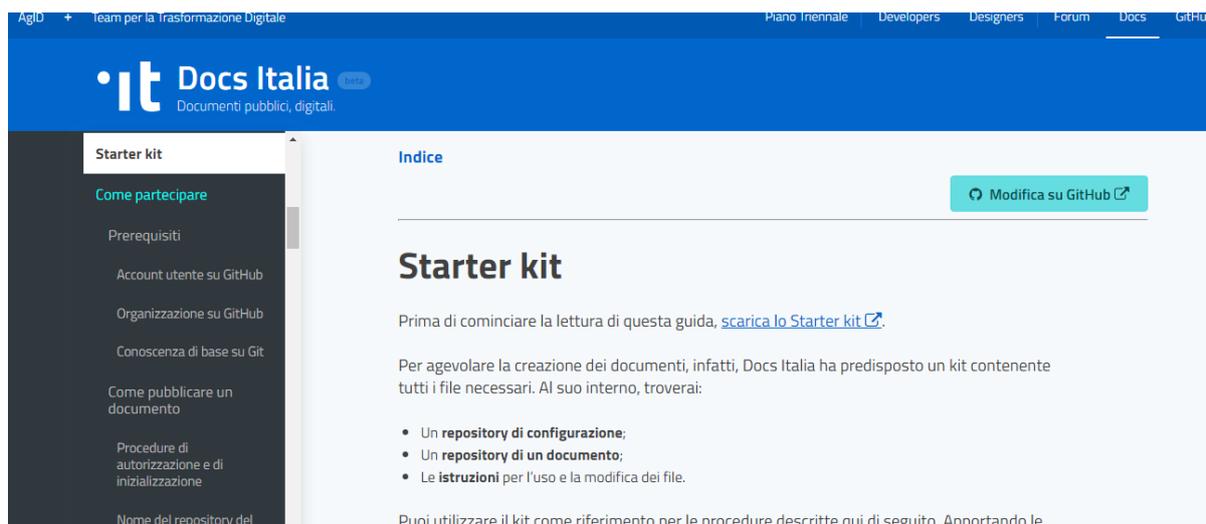
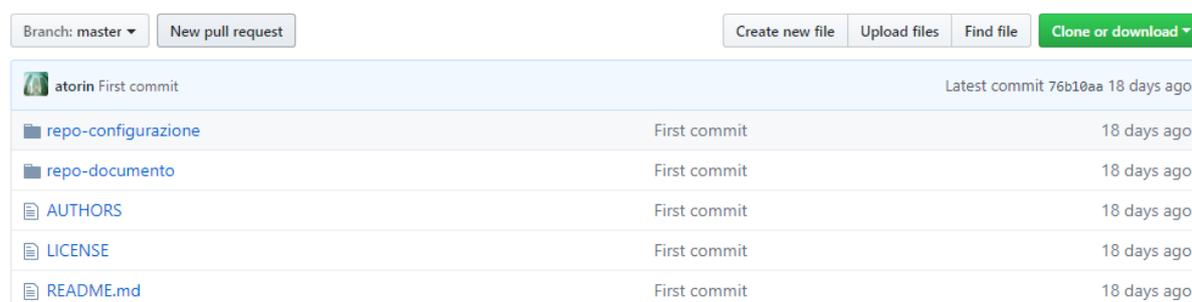


Fig. 7.1: la guida dello starter kit



## Importare il progetto dello «starter kit» nel proprio account Github

L'azione da avviare ora è l'importazione del progetto dello starter kit nel proprio account Github, dando un nome al progetto a cui si vuole lavorare. Quindi andare dentro le directory del progetto per personalizzare i contenuti e i metadati del nostro progetto. Ora vedremo come.

### Nel progetto Github abbiamo

- una directory **repo-configurazione** che contiene due file dove dobbiamo inserire un po di metadati sul progetto che andiamo a pubblicare, come: titolo, descrizione, tags, sito web ente che pubblica il documento;
- una directory **repo-documento** dove si trovano i file:
  - **conf.py** dove dobbiamo inserire il titolo del nostro documento che vogliamo visualizzare successivamente su Read the Docs. Importante: su `settings_project_name` usare un titolo non molto lungo e non usare parole accentate come à,è,ì,ò,ù, scegliendo il simbolo ' per accentare le vocali. Altra cosa importante `settings_basename = "titolo-documento"` cioè su `settings_basename` e `settings_file_name` usare parole delimitate dal simbolo - (trattino).
  - **document\_settings.yml** dove dobbiamo inserire un po di metadati sul progetto che andiamo a pubblicare: titolo, descrizione, tags, sito web ente che pubblica il documento;
  - **requirements.txt** che non dobbiamo toccare perché contiene l'istruzione che permette di visualizzare il progetto che stiamo andando a creare all'interno del template di Docs Italia, do-

ve si trovano anche tutti gli altri documenti pubblicati dal Team Trasformazione Digitale, cioè <https://docs.developers.italia.it/>,

- un file **AUTHORS** dove va scritto il nome dell'ente che pubblica il documento e della persona che ha lavorato;
- **README.md** un file dove descriviamo brevemente il nostro progetto di pubblicazione del documento;
- **LICENSE** un file dove andremo ad inserire il tipo di licenza adottata per pubblicare il nostro documento che verrà visualizzato su Read the Docs;
- **\_docs** una directory che contiene le pagine del nostro documento, cioè i contenuti veri e propri, che possono essere: testo, immagini, tabelle. Questa directory contiene file di tipo **RST** (reStructuredText). Nel paragrafo che segue approfondiremo questo tipo di file;
- **\_img**: una la directory dove andare a caricare le immagini che si vuole vengano visualizzate poi sulle pagine html di Read the Docs. E" importante nel testo dei file RST fissare le immagini (dentro il testo) usando la seguente sintassi:

```
.. figure:: _img/immagine1.png
```

Quando si vuole inserire una didascalia nell'immagine, è importante dopo aver scritto

```
.. figure:: _img/immagine1.png
```

andare due volte a capo

e lasciare tre spazi vuoti prima di inserire il testo

## La directory **\_docs** e i file RST

La directory **\_docs** contiene i file che sono le pagine del nostro documento (e i relativi contenuti) che visualizzeremo sulle pagine html di Read the Docs. Questi file sono di tipo **RST** (**reStructuredText**), un linguaggio di programmazione molto semplice per sintassi utilizzata.

Un ottima **guida alla sintassi usata dal linguaggio RST** è questa <http://docutils.sourceforge.net/docs/user/rst/quickref.html> dalla quale è possibile apprendere la sintassi corretta per le nostre necessità di pubblicazione del testo (e immagini, tabelle) nella pagina.

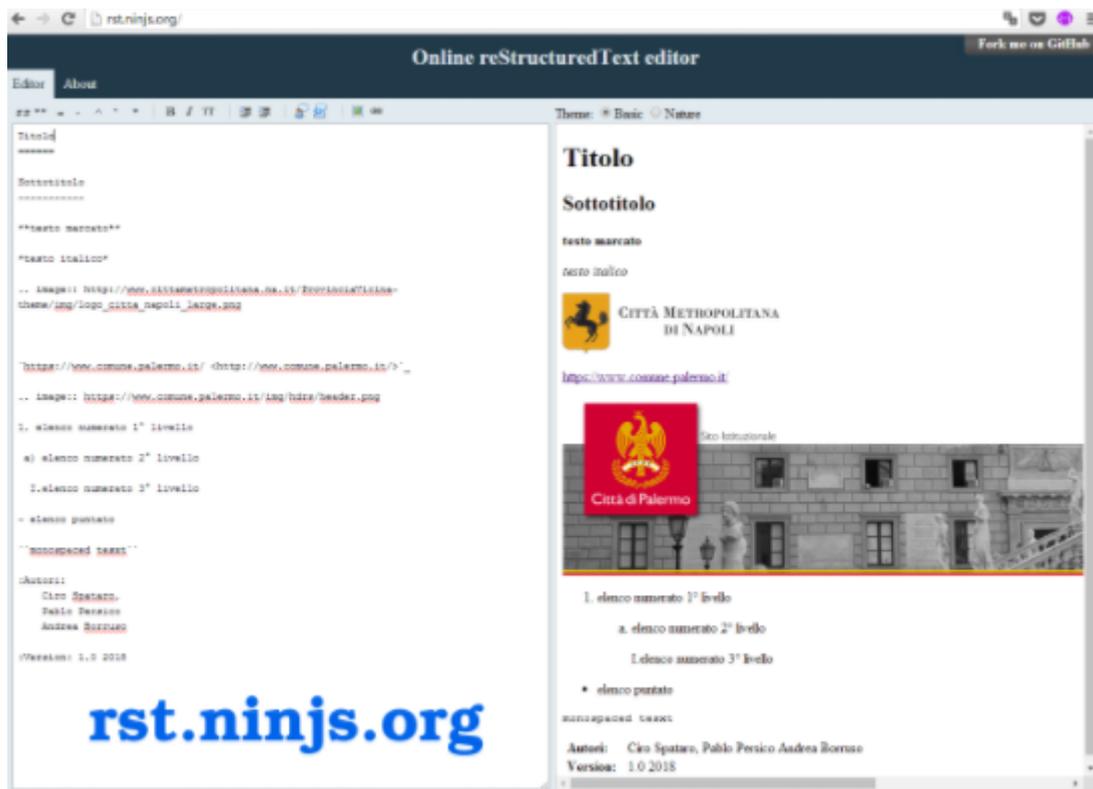
### Come Scrivere in formato ReStructuredText

Un buon editor RST online **WYSIWYG** (What You See Is What You Get: quello che vedi è quello che è) è <http://rst.ninjs.org/> dove è possibile scrivere e controllare lateralmente il risultato che si otterrà.

Alcune sintassi del linguaggio RST per gli usi più comuni di scrittura:

```
Title  
=====
```

per il titolo capitolo



```
Subtitle  
-----
```

per il titolo paragrafo

```
**testo marcato**
```

per il **testo marcato**

```
*testo in italico*
```

per il *testo in italico*

```
.. figure:: _image/immagine1.png
```

per inserire un'immagine

(il titolo del file immagine è ad esempio `immagine1.png`. Il file dovrà essere caricato in una directory denominata `img`, è consigliabile caricare questa directory contenente le immagini dentro la directory principale del progetto Github)

```
`link <https://www.link.it>`_
```

per inserire un collegamento ipertestuale. Per editare il carattere ` è necessario usare contemporaneamente i tasti `Alt+96` da Windows

```
.. important::
```

per inserire testo, se volete inserire una **nota** colorata all'interno del testo nella pagina, verrà visualizzato come di seguito:

### IMPORTANTE

Grazie di cuore se siete arrivati fin qui, siete molto coraggiosi/e e spero che i popcorn vi siano bastati :-)

[...la piccola ora ha 8 mesi, sono passati 5 mesi da quando ho iniziato a scrivere nel tempo libero (**le notti**) questo report]

quando si usa questa **nota** colorata è importante dopo aver scritto

```
.. important::
```

andare due volte a capo

    e lasciare tre spazi vuoti prima di scrivere il testo

### Nota sui file RST

---

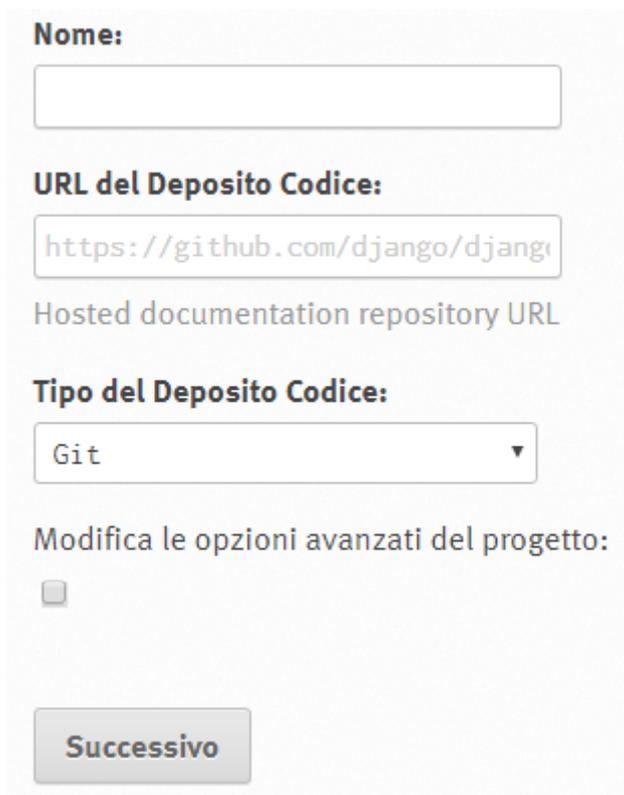
**Importante:** Guardando i file RST di qualsiasi pagina su Github (per esempio di questa pagina che state leggendo) capirete immediatamente come editare la sintassi corretta per le vostre necessità di rappresentazione di testo, immagini, tabelle, note, ecc.

---

Per tutte le necessità specifiche di editing nel documento (es.: tabelle) vedere [questo elenco di strumenti](#).

## 7.1.2 Capitolo 2 - Ora è il momento di lavorare su Read the Docs

Questa rappresenta la fase 2 ed è molto semplice come operazioni da effettuare. Una volta completato il lavoro di compilazione su Github, bisogna andare su <http://readthedocs.io> e (dopo aver creato il relativo account) importare il progetto, già creato, da Github



The screenshot shows a web form for importing a project. It has the following fields and elements:

- Nome:** An empty text input field.
- URL del Deposito Codice:** A text input field containing the URL `https://github.com/django/django`.
- Hosted documentation repository URL:** A label below the URL field.
- Tipo del Deposito Codice:** A dropdown menu with "Git" selected.
- Modifica le opzioni avanzati del progetto:** A checkbox that is currently unchecked.
- Successivo:** A button at the bottom of the form.

Nella finestra, su URL del Deposito Codice bisogna scrivere l'URL del progetto che avete creato su Github, e quindi scegliere il nome del progetto, esempio:

linee guida open data comune vattelapesca  
e lasciare Tipo del Deposito Codice selezionato su Git.

A quel punto verrà messo in collegamento il vostro progetto di Github con la piattaforma di Red the Docs.

Una primissima cosa da fare è andare su Amministrazione



e settare la lingua italiana.

Questo consentirà a Read the Docs di aggiungere al titolo del vostro progetto la desinenza `/it/latest`. Il documento è in italiano quindi prende la desinenza `/it`.

Esempio: `http://come-creare-guida.readthedocs.io/it/latest`

A questo punto il progetto di Github è compilato su Read the Docs.



Considerato che avevamo scelto come titolo del nostro progetto su Read the Docs:  
linee guida open data comune vattelapesca  
l'URL compilato da Read the Docs per vedere il nostro progetto sarà:  
`linee-guida-open-data-comune-vattelapesca.readthedocs.io`

---

#### AVVISO DI «PASSING»

##### Procedura andata a buon fine: «passing»

Se non ci sono errori commessi durante le procedure spiegate fino ad ora, tutto andrà a buon fine, e Read the Docs darà il messaggio in colore verde di «passing» al nostro progetto, significa che il nostro progetto è - quindi - online.

---

#### AVVISO DI «FAILING»

##### Procedura con Errore: «failing»

Diversamente Read the Docs alla sezione i miei progetti, darà un messaggio in colore rosso di «failed». In questo caso c'è qualche problema e bisogna ripercorrere tutti i passaggi fatti da quando si è iniziato a lavorare sul sito di Read the Docs.

---

Abbiamo completato tutte le procedure e ci possiamo godere il nostro documento nella nuova modalità di pubblicazione e visualizzazione su Read the Docs :-)

---

### 7.1.3 Capitolo 3 - Modalità Command Line (lavorando offline)

(Guida per chi usa Windows e non vuole fare a meno dei «piaceri» del terminale in stile bash).

Il lavoro da terminale consente maggiore flessibilità ed è adatto ad utenti mediamente esperti, questa modalità consente di lavorare avendo il contenuto della repository in locale sul proprio pc, modificare i file e ricaricarli modificati con soli tre comandi e avere così aggiornata la versione del documento su Read the Docs. Interessanti sono anche i tag per il “versioning” e i messaggi di commit, che vedremo come realizzare.

#### Cominciamo dal software (command line)

Per cominciare sarà necessario di installare qualche software qualora non fosse già installato. [Git for windows](#).

Nello specifico Git for Windows fornisce un'emulazione della BASH per l'utilizzo dell'applicazione da riga di comando, essa funzionerà in pratica come se si stesse lavorando in ambienti UNIX o UNIX like come Linux.

## Progetti

|   |             |         |
|---|-------------|---------|
| <b>carta servizi area cultura comune palermo</b>              | 43 builds   | passing |
| <b>libro firma</b>  | 333 builds  | passing |
| <b>linee guida open data comune palermo</b>                   | 165 builds  | passing |
| <b>regolamento commercio aree pubbliche comune palermo</b>    |             |         |
| <b>regolamento funzionamento urp comune palermo</b>           |             |         |
| <b>regolamento procedimenti amministrativi comune palermo</b> | 3 builds    | passing |
| <b>regolamento-sui-beni-confiscati-comune-palermo</b>         | 9 builds    | passing |
| <b>samba-digitale-pa</b>                                      | 1262 builds | passing |

In fase di installazione: abilitare il manager delle credenziali di Github.

### Quale editor di testo utilizzerai?

Sul terminale per fare piccole modifiche va bene Nano così come Vim entrambi già inclusi nella versione di git installata, mentre per lavorare sul testo fuori dal terminale conviene [Notepad++](#).

Per il terminale è possibile utilizzare il cmd di Windows ma consiglio caldamente: [ConEmu](#).

## Creare il progetto

Per creare il nostro progetto e implementare lo Starter Kit di Docs Italia bisognerà creare una nuova repository su GitHub:

### Create a new repository

A repository contains all the files for your project, including the revision history.

**Owner** **Repository name**

 pablopers ▾ /

Great repository names are short and memorable. Need inspiration? How about [furry-enigma](#).

**Description** (optional)

**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 let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

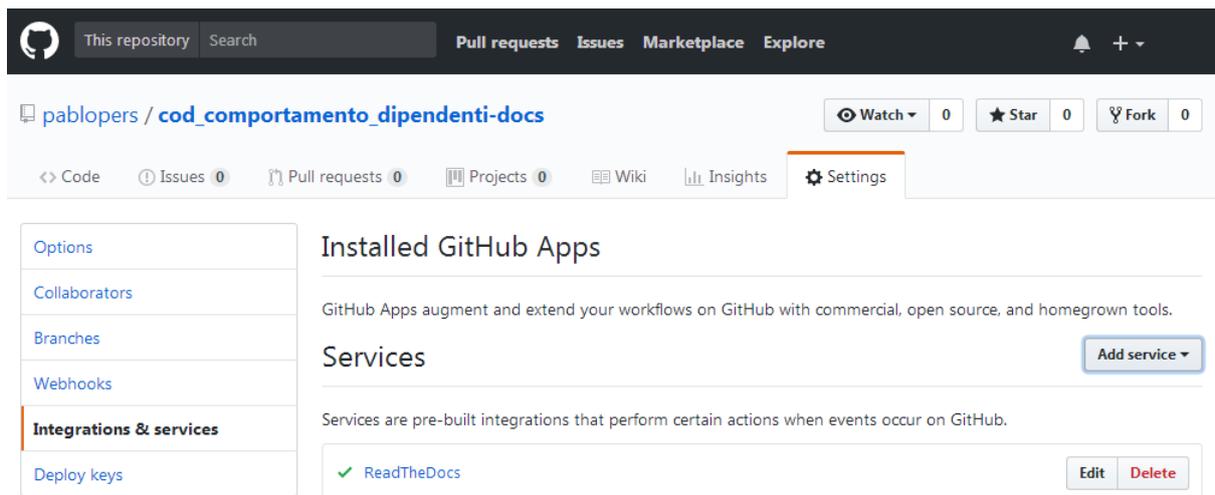
Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

**Create repository**

Nel settings della repository bisognerà aggiungere Read the Docs come *services* nella sezione *Integrations & services*.

Infine sarà necessario importare la repository in Read the Docs come indicato nel paragrafo [Ora andiamo a lavorare su Read the Docs](#).

Andiamo sul terminale e posizioniamoci in una cartella dove ci saranno i nostri repository e cloniamo lo starter kit.



Options

Collaborators

Branches

Webhooks

**Integrations & services**

Deploy keys

### Installed GitHub Apps

GitHub Apps augment and extend your workflows on GitHub with commercial, open source, and homegrown tools.

### Services

Services are pre-built integrations that perform certain actions when events occur on GitHub.

✓ ReadTheDocs

Add service

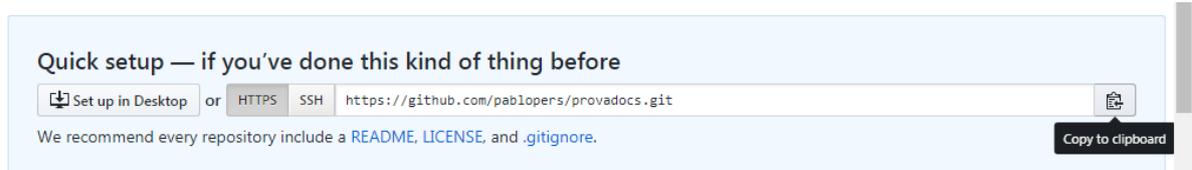
Edit Delete

```
$ git clone https://github.com/italia/docs-italia-starter-kit
```

nella stessa directory cloniamo il repository del nostro progetto

```
$ git clone https://github.com/pablopers/provadocs
```

Il link del repository lo prendete dal tasto a destra



Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH `https://github.com/pablopers/provadocs.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

Copy to clipboard

Se dovesse comparire un messaggio di errore non preoccupatevi

```
$ git clone https://github.com/pablopers/provadocs.git
Cloning into 'provadocs'...
warning: You appear to have cloned an empty repository.
```

ci dice solo che abbiamo clonato un repository vuoto (ancora per poco).

Ora dobbiamo implementare lo Starter kit nel nostro progetto copiando il contenuto completo, files e cartelle, delle cartella **repo-documento** e **repo-configurazione** presenti nello starter kit (usare come meglio preferite il terminale o explorer di Windows).

Procedere alla modifica dei file e alla creazione del documento con i relativi file rst così come già indicato nei paragrafi:

- «Nel progetto Github abbiamo»,
- «Nota sui file RST».

### Caricare il progetto su Github (command line)

Una volta realizzati i file del vostro progetto sarà necessario sincronizzare il vostro repository in locale con il repository su GitHub.

ricordare di posizionarsi nella cartella del repository clonato e digitare:

```
$ git add *
$ git commit
$ git push origin master
```

L'ultimo comando produrrà un risultato simile a questo

```
$ git push origin master
Counting objects: 15, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (15/15), 5.40 KiB | 306.00 KiB/s, done.
Total 15 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/pablopers/prova-docs.git
* [new branch]      master -> master
```

Infine su **Read the Docs** compilare una versione utilizzando il pulsante **Versione di Compilazione**, la dicitura «passing» indicherà che la compilazione è andata a buon fine.

Panoramica Download Cerca **Compilazioni** Versioni Amministrazione

### Compilazioni Recenti

Versione di Compilazione: latest ▼

|                                |                    |
|--------------------------------|--------------------|
| Passato versione latest (html) | 1 minuto addietro  |
| Fallito versione latest (html) | 16 minuti addietro |

per vedere il documento cliccare sul pulsante **Mostra Documenti**

**Mostra Documenti**

### Messaggio di Commit e Versioning

Per sapere cos'è un messaggio di commit, leggi: <http://guida-docs-italia.readthedocs.io/it/latest/index/appendice-3.html#messaggi-di-commit>

#### Perché è importante il commit?

Perché ad ogni commit corrisponde una «istantanea» del tuo repository dal quale è possibile tornare indietro azzerando l'ultima o le ultime modifiche fatte.

### Versioning per gestire gli aggiornamenti (command line)

Spesso ci si può trovare ad avere diverse versioni dello stesso documento un esempio possono essere le Linee guida degli Open Data del Comune di Palermo nella prima versione del 2013 e poi l'aggiornamento fatto nel 2017, oppure il Regolamento Foia della Città Metropolitana di Napoli in prima versione risalente al 1997.1 e oggi in via di approvazione nella versione 2018.1.

Nel caso citato può esserci di aiuto il «versioning» ossia la possibilità di definire una versione precedente con tanto di tag «1997.1» dalla più recente aggiornata versione «2018.1».

Il tag può essere inserito in fase di caricamento con i seguenti comandi:

```
$ git tag 1997.1
$ git push --tags
```

Completato il tutto con il comando:

```
$ git push origin master
```

il repository di GitHub sarà aggiornato e in automatico partirà la compilazione della versione predefinita cosiddetta (solitamente «latest»).

Per compilare la versione con il tag scelto, es.1997.1 basterà andare nella finestra compilazioni, scegliere il tag e rifare la compilazione scegliendo il tag.

**Compilazioni Recenti**

Versione di Compilazione: latest ▼

|                                |                           |
|--------------------------------|---------------------------|
| Passato versione latest (html) | 2 giorni, 11 ore addietro |
| Passato versione 1997.1 (html) | 2 giorni, 11 ore addietro |
| Passato versione 1997.1 (html) | 2 giorni, 12 ore addietro |
| Passato versione 1997.1 (html) | 2 giorni, 12 ore addietro |