
Guida a Docs Italia

Release bozza

italia

28 mag 2020

1	Che cos'è Docs Italia	3
1.1	Che cos'è	3
1.2	Obiettivi	3
1.3	Caratteristiche	4
1.4	Architettura dell'informazione	5
2	Starter kit	7
3	Come partecipare	9
3.1	Prerequisiti	9
3.2	Come pubblicare un documento	10
3.3	Amministratori di Docs Italia e assistenza	13
4	Scrivere un documento	15
4.1	Il formato reStructuredText (RST)	15
4.2	Creazione di documenti in formato RST	16
4.3	Migrazione su Docs Italia di documentazione esistente	17
5	Pubblicare un documento	19
5.1	Repository di configurazione	19
5.2	Repository del documento	21
5.3	Funzione commenti con Forum Italia	25
5.4	Caricare i file sul repository remoto	25
5.5	Backend di Docs Italia	26
6	Gestione dei documenti	29
6.1	Visibilità del documento	29
6.2	Operazioni di manutenzione	30
7	Appendice 1. Il versionamento	33
7.1	Introduzione	33
7.2	Glossario minimo	33
7.3	Tipi di versionamento	34
7.4	Estensione del processo di versionamento	34
8	Appendice 2. Sintassi del formato RST per Docs Italia	37
8.1	Struttura del documento	37

8.2	Formattazione e markup di base	38
8.3	Titoli	38
8.4	Sezioni	39
8.5	Indice	39
8.6	Riassunto	40
8.7	Figure	40
8.8	Tabelle	41
8.9	Glossario	41
8.10	Citazioni	42
8.11	Riferimenti	43
8.12	Note	43
8.13	Box	43
8.14	Componente “In consultazione”	45
8.15	Componente “Procedura”	45
8.16	Componente “Documenti utili”	46
8.17	Componente “Tono di voce”	46
8.18	Link utili	47
9	Appendice 3. Procedure e convenzioni su GitHub	49
9.1	Procedure di caricamento sul repository remoto	49
9.2	Messaggi di commit	51

Guida alla pubblicazione dei documenti della Pubblica Amministrazione su [Docs Italia](https://docs.italia.it)¹.

¹ <https://docs.italia.it>

Che cos'è Docs Italia

1.1 Che cos'è

Docs Italia è un servizio ideato dal [Team per la trasformazione digitale della Pubblica Amministrazione](https://teamdigitale.governo.it/)² in collaborazione con [AgID](http://www.agid.gov.it/)³ ed è gestito da un team interdisciplinare di sviluppatori, designer e tech writer esperti nel documentare un progetto attraverso guide, linee guida, FAQ e documentazione tecnica e amministrativa. Insieme a [Forum Italia](http://forum.italia.it)⁴ e [GitHub Italia](https://github.com), è uno degli strumenti operativi per la trasformazione digitale dei servizi pubblici.

Al momento, Docs Italia contiene alcuni dei più importanti progetti pubblici del Paese, come ANPR (anagrafe nazionale), SPID (identità pubblica), pagoPA (sistemi di pagamento) e molti altri, visibili nella home page del sito. Altri si aggiungeranno progressivamente, sulla base delle richieste delle Pubbliche Amministrazioni che possono richiedere di utilizzare il servizio tramite [Slack di Developers Italia](https://slack.developers.italia.it/)⁵ usando il canale `#docs-italia`⁶.

1.2 Obiettivi

Docs Italia nasce con l'obiettivo di cambiare il linguaggio con cui si comunicano e si gestiscono i progetti pubblici in Italia, creando strumenti di lavoro comuni a diverse culture professionali e in cui gli aspetti normativi e tecnici possano coesistere. In questo modo, si semplifica la vita di chi lavora ai progetti (in particolare i funzionari, i tecnici e i fornitori della Pubblica Amministrazione), aumentando l'efficacia dei progetti pubblici e perseguendo una impostazione di *open government* che favorisca l'apertura, lo scambio di idee e la collaborazione.

Docs Italia permette di gestire la documentazione tecnica di un progetto (ad es. la documentazione di pagoPA), ma anche documenti amministrativi come il [Piano Triennale per l'informatica nella Pubblica Amministrazione](#)⁷ e le linee guida (come le [Linee guida di design per i servizi web della PA](#)⁸).

La [Figura 1.1](#) mostra l'attuale home page di Docs Italia.

² <https://teamdigitale.governo.it/>

³ <http://www.agid.gov.it/>

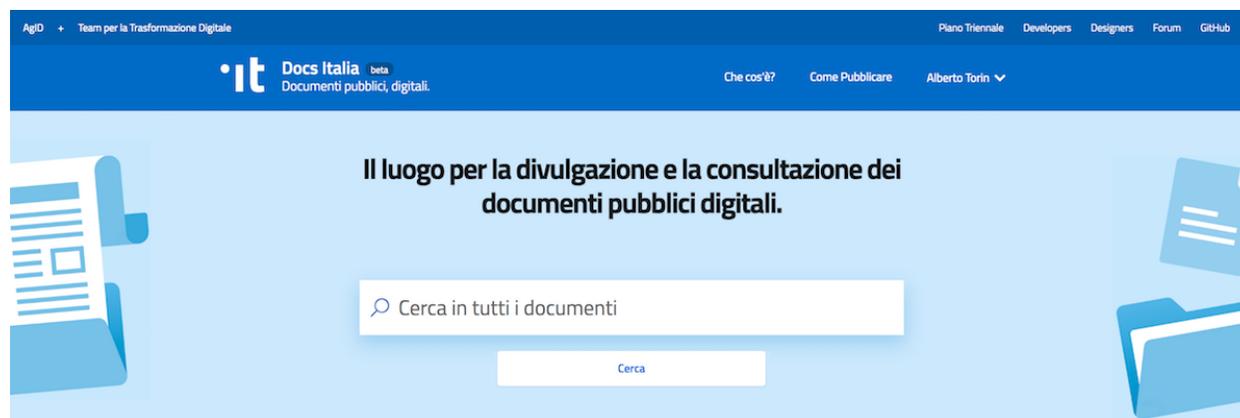
⁴ <http://forum.italia.it>

⁵ <https://slack.developers.italia.it/>

⁶ <https://developersitalia.slack.com/messages/C9T4ELD4G/>

⁷ <https://pianotriennale-ict.readthedocs.io>

⁸ <https://design-italia.readthedocs.io>



Documenti in evidenza

 Piano Triennale per l'informatica nella Pubblica Amministrazione. Codice dell'amministrazione digitale Lo Stato, le Regioni e le autonomie locali assicurano la disponibilità, la gestione, l'accesso, la trasmissione, la conservazione e la fruibilità dell'informazione in modalità digitale e si organizzano ed agiscono a tale fine utilizzando con le modalità più...	 Piano Triennale per l'informatica nella Pubblica Amministrazione. Censimento ICT Il Censimento del Patrimonio ICT della PA si propone, con cadenza annuale, di rilevare i dati necessari per delineare il quadro informativo/statistico sulle principali installazioni informatiche a livello nazionale, regionale e locale, determinando per ogni singola...	 Designers Italia Guida al linguaggio della Pubblica Amministrazione La guida al linguaggio della Pubblica Amministrazione è uno strumento per aiutarvi a creare e gestire i contenuti di siti o altri servizi digitali pubblici. La guida al linguaggio è parte del kit di Designers Italia dedicato alla progettazione, gestione e produzione...	 Sistema Pubblico di Identità Digitale Regole tecniche SPID, il Sistema Pubblico di Identità Digitale, è la soluzione che ti permette di accedere a tutti i servizi online della Pubblica Amministrazione con un'unica Identità Digitale (username e password) utilizzabile da computer, tablet e smartphone.
---	---	---	--

Fig. 1.1: L'attuale home page di Docs Italia.

1.3 Caratteristiche

Docs Italia si ispira a cinque principi, funzionali a costruire un linguaggio comune di gestione dei progetti pubblici:

1. il primo è che oggi, in tutto il mondo, i documenti non sono più il prodotto di un individuo e di un ufficio, ma sono sempre di più il frutto di una collaborazione che prevede il contributo di diverse persone e diversi profili professionali, tecnici, economici e amministrativi;
2. il secondo è che la Pubblica Amministrazione, seguendo una filosofia di [open government](#)⁹, tende a incoraggiare il contributo della comunità e di punti di vista esterni;
3. il terzo è che la Pubblica Amministrazione ha bisogno di modalità comuni per gestire le informazioni relative a un progetto, dalle linee guida di indirizzo alla documentazione tecnica;
4. il quarto è che i documenti non sono un oggetto statico, scolpito nella pietra o stampato nella carta, ma evolvono nel tempo: poter tenere traccia di questa evoluzione offre diversi vantaggi;
5. il quinto è che i documenti vengono fruiti sempre più spesso su Internet, e in particolare su smartphone: la semplicità di lettura fa la differenza.

1.3.1 Nuove funzionalità

- Docs Italia è organizzato per *publisher* (l'ente o il Ministero che pubblica i documenti) e per progetti (ciascun progetto può avere diversi documenti).

⁹ https://it.wikipedia.org/wiki/Open_government

- Docs Italia ha un motore di ricerca che permette di navigare tra i documenti. La funzione di ricerca diventa sempre più importante mano a mano che cresce la mole di testi pubblicati sulla piattaforma.
- Un widget consentirà di pubblicare sul proprio sito web l'elenco dei documenti appartenenti a uno specifico publisher (ad es. Ministero delle Finanze, MIUR) e a uno specifico progetto (ad es. ANPR). Sarà inoltre possibile pubblicare l'indice del documento.
- Attualmente è in corso di analisi lo sviluppo di una componente per [LibreOffice](#)¹⁰ che permetta l'esportazione e l'importazione di documenti in formato RST.
- Ogni documento su Docs Italia è associato a un URL predittivo del tipo: `docs.italia.it/<publisher>/<progetto>/<nome-documento>`.

La [roadmap dello sviluppo di Docs Italia](#)¹¹ è disponibile online, espressa attraverso una board di Trello.

1.4 Architettura dell'informazione

1.4.1 Publisher

Ogni documento su Docs Italia è associato a un *publisher*. Si tratta dell'amministrazione che pubblica il documento e che ne è la responsabile.

Ogni publisher (ad esempio un comune o un ministero) è collegato a un'organizzazione su GitHub, che a sua volta ospita tutti i repository dei documenti di quella Pubblica Amministrazione.

1.4.2 Progetto

Ogni documento su Docs Italia appartiene a un progetto, ovvero una specifica linea di lavoro o di interesse per la Pubblica Amministrazione. I documenti all'interno di uno stesso progetto hanno un tema comune, spesso relativo a una piattaforma, a una legge o a un servizio. Alcuni progetti attualmente presenti su Docs Italia sono:

- Anpr (Anagrafe nazionale della popolazione residente);
- SPID (Sistema di identità digitale);
- Piano triennale per l'informatica nella Pubblica Amministrazione.

Ogni progetto è a sua volta associato a un unico publisher.

1.4.3 Pagina documento

Ciascun documento ha le seguenti caratteristiche (vedi [Figura 1.2](#)):

1. L'utente può effettuare una ricerca specifica all'interno del documento;
2. È sempre possibile scegliere la versione del documento tramite un apposito pulsante;
3. È possibile consultare i metadati principali del documento, inclusi i tag;
4. È sempre possibile scaricare il documento in formato .epub e in formato .pdf;
5. Il pulsante "modifica su GitHub" permette di accedere al repository su GitHub dove è ospitato il documento e consente a chiunque di proporre delle modifiche al contenuto;
6. L'indice nella barra laterale consente di passare rapidamente da una sezione all'altra.

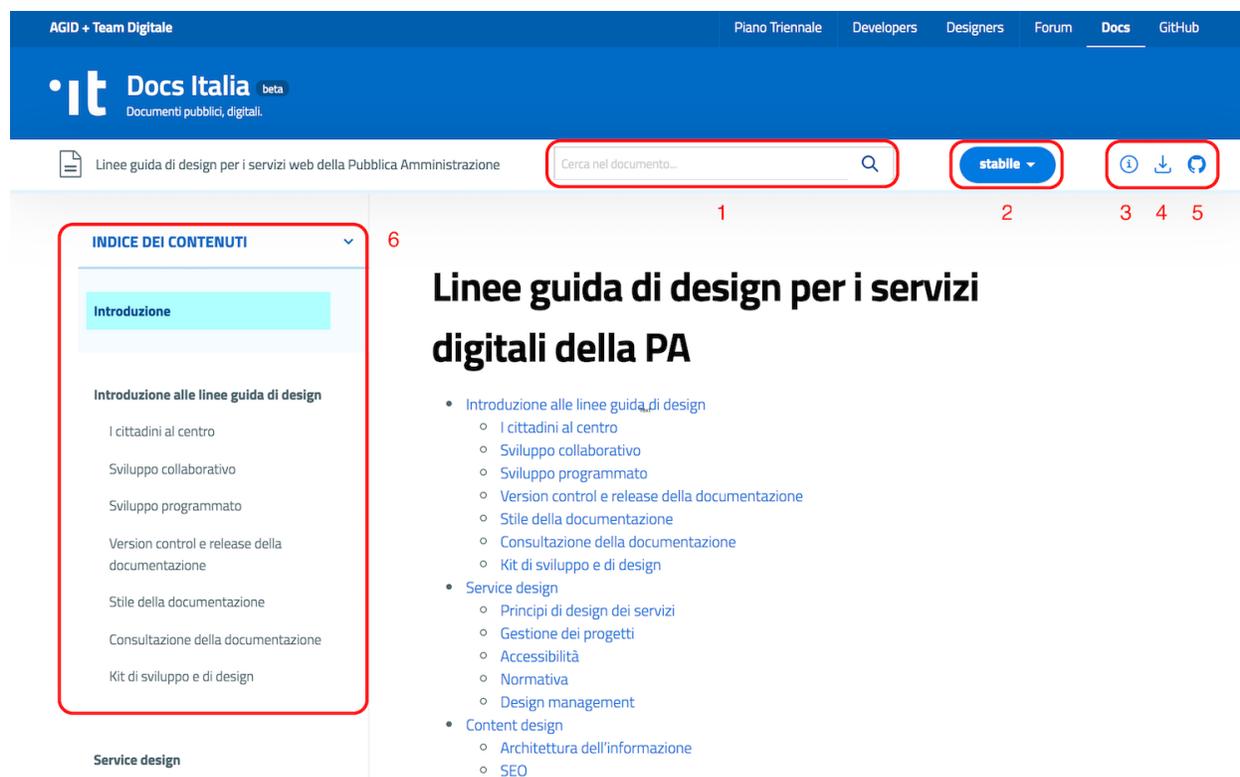


Fig. 1.2: Una pagina documento su Docs Italia.

Oltre a queste caratteristiche,

- L'ambiente di Docs Italia è *responsive*, ovvero consente una perfetta fruizione dei documenti su tutti i dispositivi, compreso lo smartphone;
- Le ancore in corrispondenza dei capitoli consentono di ottenere facilmente il link di un paragrafo. Per fare apparire l'ancora basta spostare il puntatore del mouse sopra al titolo corrispondente;
- Alcune funzionalità di paratesto specifiche permettono di segnalare informazioni importanti, esempi, procedure e documenti in consultazione (vedi l'Appendice 2).

1.4.4 La piattaforma

- La piattaforma di Docs Italia è basata sui progetti open source [Sphinx](https://www.libreoffice.org/)¹² e [Read the Docs](https://readthedocs.org/)¹³. I documenti pubblicati sono redatti in [formato reStructuredText](http://docutils.sourceforge.net/rst.html)¹⁴.
- Docs Italia si presta a una ottimizzazione SEO, facilitando la ricerca dei documenti pubblici anche attraverso i motori di ricerca.
- Docs Italia è integrato con [Forum Italia](https://forum.italia.it)¹⁵ (a sua volta basato sul progetto open source [Discourse](https://discourse.org/)¹⁶) per consentire di commentare i documenti.

¹⁰ <https://www.libreoffice.org/>

¹¹ <https://trello.com/b/jQUgRzRe/docs-italiasviluppo>

¹² <http://sphinx-doc.org/>

¹³ <https://readthedocs.org/>

¹⁴ <http://docutils.sourceforge.net/rst.html>

¹⁵ <http://forum.italia.it>

¹⁶ <https://discourse.org/>

Starter kit

Prima di cominciare la lettura di questa guida, scarica lo Starter kit¹⁷.

Per agevolare la creazione dei documenti, infatti, Docs Italia ha predisposto un kit contenente tutti i file necessari. Al suo interno, troverai:

- Un **repository di configurazione**;
- Un **repository di un documento**;
- Le **istruzioni** per l'uso e la modifica dei file.

Puoi utilizzare il kit come riferimento per le procedure descritte qui di seguito. Apportando le opportune modifiche ai file, sarai in grado di pubblicare rapidamente i tuoi documenti su Docs Italia.

¹⁷ <https://github.com/italia/docs-italia-starter-kit>

3.1 Prerequisiti

Docs Italia si basa su [GitHub](#)¹⁸ come strumento per pubblicare i documenti (e gestirne le versioni).

Per pubblicare un documento, è necessario avere:

1. Un **account utente su GitHub**;
2. Un'**organizzazione su GitHub**.

3.1.1 Account utente su GitHub

Se fai parte o lavori per un Ente pubblico e vuoi pubblicare i tuoi documenti su Docs Italia, hai bisogno prima di tutto di un account su GitHub. La registrazione è gratuita e può essere completata seguendo le [istruzioni nella pagina ufficiale](#)¹⁹.

L'account su GitHub consente l'identificazione dell'utente e permette di effettuare tutte le operazioni in maniera sicura e tracciabile.

3.1.2 Organizzazione su GitHub

L'Ente pubblico deve essere associato a un'[organizzazione su GitHub](#)²⁰.

Come creare un'[organizzazione su GitHub](#)²¹

Un'[organizzazione](#) è un insieme di repository gestiti da più utenti GitHub che collaborano a progetti comuni.

¹⁸ <https://github.com/>

¹⁹ <https://help.github.com/articles/signing-up-for-a-new-github-account/>

²⁰ <https://github.com/blog/674-introducing-organizations>

²¹ <https://github.com/organizations/new>

L'organizzazione rappresenta una modalità efficiente di raggruppare tutti i progetti digitali di un Ente pubblico. Offre, inoltre, alcuni vantaggi nella gestione dei permessi degli utenti e opzioni di sicurezza e amministrazione avanzate. Anche l'account organizzazione è gratuito e permette un numero illimitato di repository e collaboratori.

La [Figura 3.1](#) mostra la pagina dell'organizzazione Developers Italia su GitHub. È possibile notare il numero di repository aperti, le persone che collaborano con l'organizzazione e i team.

Fig. 3.1: L'organizzazione Developers Italia su GitHub

Nota: Se la tua amministrazione per qualche ragione non può creare un'organizzazione su GitHub, [contatta gli amministratori di Docs Italia](#) (pagina 13) per richiedere un repository nell'organizzazione di Developers Italia.

3.1.3 Conoscenza di base su Git

Docs Italia usa il [version control system](#)²² di [Git](#)²³, che permette di gestire le diverse versioni di un documento. La pubblicazione su Docs Italia richiede una conoscenza di base di Git e di alcuni termini collegati. Consulta il [Glossario minimo](#) (pagina 33) in Appendice.

3.2 Come pubblicare un documento

La pubblicazione di un documento su Docs Italia avviene secondo le seguenti fasi:

1. L'amministrazione crea un'organizzazione su GitHub;
2. L'amministrazione crea un repository di configurazione nella propria organizzazione;

²² https://it.wikipedia.org/wiki/Controllo_versione

²³ <https://git-scm.com/>

3. L'amministrazione richiede l'autorizzazione alla pubblicazione;
4. L'amministrazione crea un repository del documento dove caricare la propria documentazione.

I primi tre passaggi sono richiesti soltanto al primo accesso a Docs Italia: in questo caso, le principali operazioni sono a carico degli amministratori della piattaforma.

Il caricamento e la modifica dei file contenuti nei repository di configurazione e di documento rientrano, invece, in un processo iterativo, che può essere svolto anche in maniera collaborativa. Consulta il capitolo su [come pubblicare un documento](#) per maggiori informazioni.

3.2.1 Procedure di autorizzazione iniziali

Le procedure di autorizzazione iniziali servono a verificare le credenziali dell'utente e a consentire la pubblicazione su Docs Italia soltanto alle amministrazioni autorizzate.

L'Ente che vuole pubblicare su Docs Italia crea un'organizzazione su GitHub dove ospiterà i repository per la propria documentazione.

Nota: Requisiti per le organizzazioni

- Per poter importare i repository di un'organizzazione su Docs Italia, un utente deve rendere pubblica la propria appartenenza all'organizzazione. Per controllare l'appartenenza bisogna visitare questo indirizzo <https://github.com/orgs/<nome-org>/people>, avendo cura di sostituire <nome-org> con il nome della propria organizzazione. L'appartenenza a un'organizzazione è pubblica quando, nella scheda **People** della pagina dell'organizzazione, in corrispondenza del proprio nome compare la scritta **Public** e non **Private**.
- Per permettere a un utente membro di un'organizzazione di importare i repository dei documenti, è necessario rimuovere le restrizioni sulle applicazioni di terze parti dalle impostazioni di GitHub.

Dalla pagina dell'organizzazione, vai su **Settings, Third-party access**, quindi clicca sul pulsante **Remove restrictions** (vedi la [Figura 3.2](#)). Di fianco a *Policy* apparirà la dicitura **No restrictions**.

Dopo aver compiuto queste operazioni preliminari, è possibile seguire la procedura qui sotto.

Procedura

1. Un amministratore dell'account organizzazione crea un apposito repository di configurazione chiamato **italia-conf**, contenente alcuni file necessari a identificare l'amministrazione, i progetti e i documenti che si desidera pubblicare su Docs Italia. *Maggiori informazioni sul repository di configurazione* (pagina 19) sono disponibili nel capitolo dedicato alla pubblicazione. Un esempio di repository di configurazione si trova nello *Starter kit* (pagina 7) fornito.
2. Un amministratore dell'account organizzazione invia una richiesta di autorizzazione alla pubblicazione su Docs Italia tramite [Slack di Developers Italia](#)²⁴, usando il canale [#docs-italia](#)²⁵.
3. Gli amministratori di Docs Italia, effettuate le opportune verifiche, autorizzano l'organizzazione alla pubblicazione.
4. Un amministratore dell'account organizzazione crea un repository per il documento. Il nome del repository deve rispettare le *convenzioni sui nomi di Docs Italia* (pagina 12). A questo punto può aggiornare il file `document_settings.yml` e caricare i file del documento secondo le modalità indicate nella sezione *Repository del documento* (pagina ??). Un esempio completo di repository del documento è contenuto nello *Starter kit* (pagina ??).

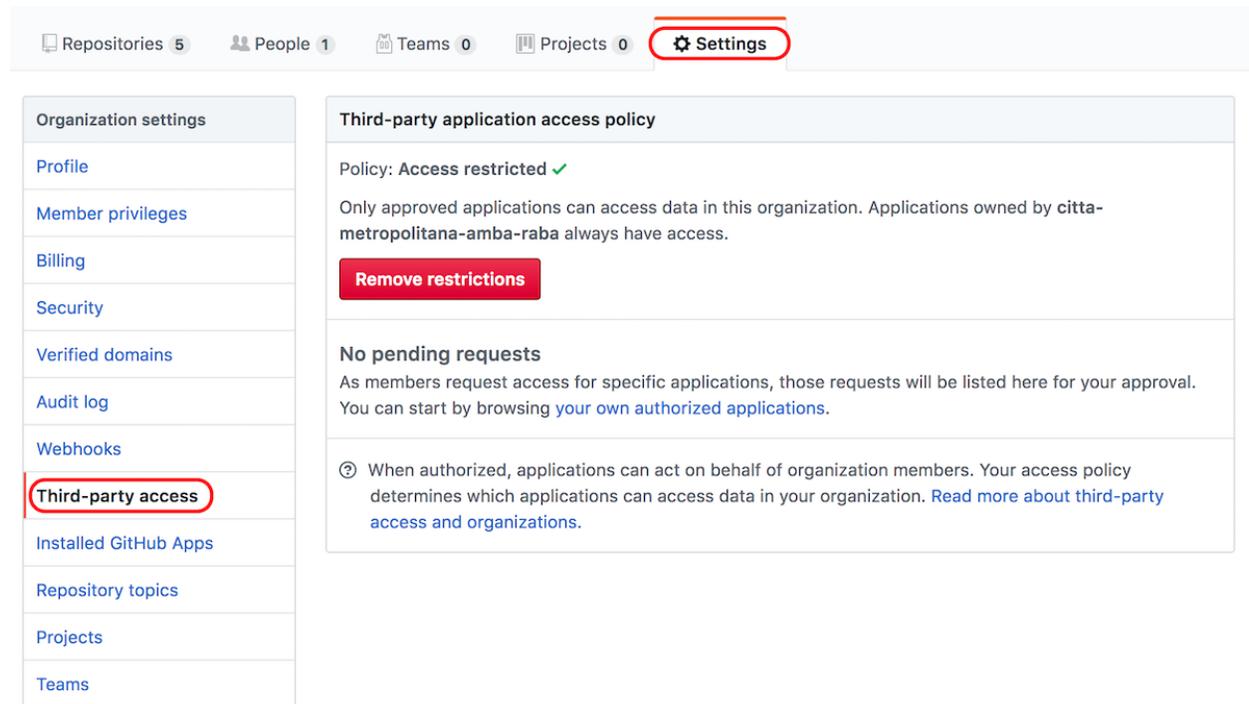


Fig. 3.2: Come rimuovere le restrizioni sulle applicazioni di terze parti per un'organizzazione su GitHub.

example

Processo di autorizzazione per organizzazioni GitHub

Il Ministero dell'Interno vuole pubblicare su Docs Italia un documento chiamato “Modalità di subentro”, relativo al progetto ANPR. Il Ministero dell'Interno ha un'organizzazione su GitHub, la cui amministratrice è Giulia Rossi.

- Giulia Rossi crea un repository di configurazione presso l'organizzazione GitHub del Ministero dell'Interno.
- Giulia Rossi invia la richiesta di autorizzazione alla pubblicazione su Docs Italia, e la sua richiesta viene approvata.
- Giulia Rossi crea, infine, un repository del documento presso l'organizzazione GitHub del Ministero dell'Interno, dove inserirà tutti i file relativi alla documentazione, aggiornando allo stesso tempo il file `document_settings.yml`.

3.2.2 Nome del repository del documento

Il nome del repository del documento deve seguire il formato: **nomeprogetto-nomedocumento-docs**.

Per esempio, un documento dal titolo “Istruzioni per il cambio di residenza” all'interno del progetto ANPR potrebbe essere ospitato nel repository **anpr-cambioresidenza-docs**.

Il nome deve sempre finire con **-docs** per segnalare che il repository contiene della documentazione.

²⁴ <https://slack.developers.italia.it/>

²⁵ <https://developersitalia.slack.com/messages/C9T4ELD4G/>

3.2.3 Passi successivi

Dopo aver creato i repository, è possibile caricare i file per generare la documentazione. Le procedure sono descritte nel capitolo *Pubblicare un documento* (pagina 19).

3.3 Amministratori di Docs Italia e assistenza

Docs Italia ha dei maintainer con dei privilegi di amministrazione che permettono loro qualsiasi intervento all'interno della piattaforma. Un maintainer di Docs Italia può, per esempio, gestire gli utenti e rimuovere dei documenti già pubblicati.

Tuttavia, nel caso in cui siano chiamati a supportare la creazione di un documento ospitato in un repository di un'organizzazione GitHub, i maintainer devono farsi autorizzare dagli amministratori del repository specifico.

Le attività di pubblicazione utilizzando GitHub e Docs Italia sono a carico dell'organizzazione e dell'utente. In caso di problemi, è possibile chiedere supporto al **servizio assistenza di Docs Italia** tramite [Slack di Developers Italia](#)²⁶ usando il canale [#docs-italia](#)²⁷.

²⁶ <https://slack.developers.italia.it/>

²⁷ <https://developersitalia.slack.com/messages/C9T4ELD4G/>

4.1 Il formato reStructuredText (RST)

I testi di partenza per la pubblicazione su Docs Italia devono essere in formato reStructuredText²⁸ (di seguito anche .rst o RST). Si tratta di file di testo redatti secondo *specifiche regole sintattiche*²⁹. La formattazione è ottenuta tramite speciali combinazioni di caratteri, che vengono interpretate da Docs Italia durante la creazione delle pagine.

L'esempio nella *Tabella 4.1* illustra come è possibile indicare il titolo del documento e una sezione, nonché come ottenere testo in grassetto e in corsivo. Consulta una *guida rapida alla sintassi RST*³⁰, oppure la *lista completa delle specifiche del linguaggio*³¹.

Tabella 4.1: Confronto fra testo RST e testo interpretato.

Testo in formato RST	Testo interpretato
<pre>***** Il titolo del documento ***** Lorem ipsum... Una sezione ===== Una frase **in grassetto**. Una frase <i>*in corsivo*</i>.</pre>	<pre>Il titolo del documento Lorem ipsum. . . Una sezione Una frase in grassetto. Una frase <i>in corsivo</i>.</pre>

²⁸ <https://it.wikipedia.org/wiki/ReStructuredText>

²⁹ <http://docutils.sourceforge.net/rst.html>

³⁰ <http://docutils.sourceforge.net/docs/user/rst/quickref.html>

³¹ <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html>

Per maggiore chiarezza, puoi consultare un [documento di esempio](#)³² contenente gli elementi di base della formattazione. Il documento è leggibile in formato RST e nei formati usati da Microsoft Office e LibreOffice, per un facile confronto.

4.1.1 Editor di testo

Per la creazione e la modifica dei file RST è sufficiente un editor di testo. In linea di principio, qualsiasi editor può essere usato, anche se alcuni programmi risultano più efficaci di altri. Ecco alcuni suggerimenti.

Atom

[Atom](#)³³ è un editor di testo avanzato, open source e sviluppato da GitHub, che permette la creazione e modifica, fra gli altri, di documenti in formato .rst. Atom consente di evidenziare adeguatamente il markup RST.

Questo editor è disponibile per varie piattaforme: Linux, Windows, MacOS.

Notepad++

[Notepad++](#)³⁴ è un editor di testo open source disponibile per il sistema operativo Windows. Come Atom, permette la creazione e modifica di documenti in formato .rst con visualizzazione del linguaggio di markup RST.

Online Editor

Oltre agli editor stand alone, è disponibile anche il seguente editor online [rst.ninjs.org](#)³⁵ che permette di creare, modificare e visualizzare istantaneamente documenti in formato .rst secondo la logica WYSIWYG (What You See Is What You Get).

Tabelle .rst

Per realizzare le tabelle in formato .rst è possibile utilizzare un [editor di tabelle online](#)³⁶.

4.2 Creazione di documenti in formato RST

Il primo aspetto rilevante per scrivere una documentazione efficace è [adottare il punto di vista degli utenti che la useranno](#)³⁷, in questo caso in particolare funzionari e tecnici della Pubblica Amministrazione e dei suoi fornitori. Scrivi il tuo testo seguendo i suggerimenti sulla struttura e sul linguaggio illustrati nella guida di stile in appendice. Puoi includere nel tuo documento titoli, tabelle, immagini e link esterni, utilizzando la sintassi opportuna.

Il contenuto del tuo testo può essere diviso in vari file .rst per facilitare l'organizzazione e la lettura. Tale divisione può avvenire tipicamente a livello di capitolo o di sezione (vedi anche [Struttura del repository](#) (pagina 22)). Docs Italia combinerà insieme i file per creare l'intero documento, rispettando i link interni e la struttura.

Nota: File separati diventeranno pagine HTML separate, facilitando la lettura da parte dell'utente.

Procedura.

1. Scrivi il testo utilizzando il tuo editor preferito o *uno di quelli suggeriti da noi* (pagina 16). Utilizza la sintassi RST per titoli, sottotitoli, liste e link.

³² <https://github.com/italia/docs-italia-esempiformattazione-docs>

³³ <https://atom.io/>

³⁴ <https://notepad-plus-plus.org/>

³⁵ <http://rst.ninjs.org/>

³⁶ <http://truben.no/table/>

³⁷ <https://gdstechnology.blog.gov.uk/2016/10/28/writing-documentation-for-developers/>

2. Salva il tuo file in formato .rst. In alcuni editor, specie su Windows, potrebbe essere necessario selezionare “Tutti i file” e aggiungere l’estensione manualmente.

4.3 Migrazione su Docs Italia di documentazione esistente

Nel caso in cui si abbiano già dei documenti di partenza (per esempio, in formato DOCX, ODT o PDF), questi devono essere convertiti in RST per poter essere pubblicati su Docs Italia. La conversione è in parte automatica ma necessita di una revisione manuale.

4.3.1 Conversione col convertitore web

Lo strumento di conversione principale verso RST è il [convertitore web](#)³⁸.

Nota:

- Il convertitore accetta documenti in formato DOCX e ODT, ma non in formato DOC.
- Il convertitore non accetta documenti in formato PDF.

Procedura. Conversione di un documento

1. Converti col convertitore web
2. Controlla la conversione automatica ed esegui una revisione manuale del testo

Attenzione: Se il documento di partenza è un PDF, è necessaria una prima conversione verso DOCX.

4.3.2 Revisione dei contenuti e correzione degli errori

La revisione del testo è necessaria perché la conversione automatica può presentare degli errori di sintassi. Talvolta, le tabelle costituiscono un elemento problematico, specie quelle con struttura non regolare (ad es., presenza di celle multiple o tabelle annidate).

Per correggere gli errori, è necessaria una **revisione manuale del file** utilizzando uno degli editor di documenti RST già presentati. Alcuni editor consentono di visualizzare un’anteprima automatica delle modifiche al testo (consulta la sezione *Editor di testo* (pagina 16)). Correggi uno a uno gli errori di formattazione che si presentano, assicurandoti di rispettare la *sintassi dei documenti .rst*³⁹.

Eliminati gli errori di sintassi, è necessario uniformare il documento allo *stile di Docs Italia*⁴⁰. Consulta la Guida di stile contenuta nell’Appendice 2 per maggiori informazioni.

³⁸ <https://docs.italia.it/converti/>

³⁹ <http://docutils.sourceforge.net/docs/user/rst/quickref.html>

⁴⁰ <https://design-italia.readthedocs.io/it/stable/>

4.3.3 Comandi di conversione

Se preferisci utilizzare degli strumenti da riga di comando che non richiedono una connessione a Internet, puoi convertire i documenti utilizzando direttamente i [comandi di conversione](#)⁴¹, che forniscono le stesse funzionalità del convertitore web attraverso un'interfaccia testuale.

Questo approccio è consigliato per gli utenti che abbiano familiarità con la riga di comando, che vogliano convertire molti file, o che intendano usare la conversione come un passaggio intermedio all'interno di script personalizzati.

⁴¹ <https://github.com/italia/docs-italia-comandi-conversione>

Publicare un documento

Dopo aver scritto la documentazione in formato RST, è possibile avviare la fase di pubblicazione.

In questo capitolo vengono forniti alcuni chiarimenti sul repository di configurazione, sul repository del documento e sul backend di Docs Italia.

5.1 Repository di configurazione

Il repository di configurazione deve essere chiamato **italia-conf** e contenere le informazioni (metadati) relative al publisher, ai progetti e ai documenti correlati che appaiono in vari punti all'interno delle pagine su Docs Italia.

Puoi modificare i file nel repository di configurazione presente nello Starter kit come descritto qui sotto. Successivamente, dovrai **caricare i file sul repository remoto** creato in precedenza (vedi *Come pubblicare un documento* (pagina 10)) usando *una delle due procedure descritte in Appendice* (pagina 49).

example

Repository di configurazione di prova

Lo *Starter kit* (pagina 7) contiene un esempio di repository di configurazione. Consulta le istruzioni contenute nel kit per conoscere come modificare i file.

5.1.1 Contenuto del repository

Il repository di configurazione deve contenere due file principali:

1. **publisher_settings.yml**, per i metadati relativi al publisher;
2. **projects_settings.yml**, per i metadati relativi ai progetti.

In questi file vengono specificati, per esempio, quali progetti appartengono al publisher e quali repository del documento appartengono a un determinato progetto. In aggiunta, nel repository di configurazione possono essere inclusi i loghi del publisher o dei progetti.

Oltre ai metadati liberamente modificabili, esiste un **set di tag tratti da un vocabolario controllato**, condiviso fra Docs Italia e Forum Italia. Tramite questi tag (almeno 5 per ciascun publisher o progetto) si stabilisce un collegamento fra i contenuti di Docs Italia e gli argomenti nel Forum, permettendo agli utenti di trovare più rapidamente quello che cercano.

Le Tabelle 2 e 3 illustrano alcuni dei possibili metadati che è possibile specificare nei file di configurazione. Gli esempi associati mostrano come formattare i file di configurazione. Per maggiori informazioni è possibile consultare le istruzioni allegate allo *Starter kit* (pagina 7).

5.1.2 Metadati del publisher

Importante: I campi **name** e **description** sono obbligatori.

Tabella 5.1: Alcuni dei metadati associati al publisher.

Parametro	Descrizione
name	Il nome per esteso dell'Ente associato al publisher
description	Una descrizione estesa delle funzioni e degli scopi dell'Ente
short_name	Il nome abbreviato dell'Ente associato al publisher o l'acronimo (opzionale)
motto	Il motto o una breve frase che contraddistingue l'Ente (opzionale)
logo	L'URL del logo (può essere contenuto nel repository di configurazione) (opzionale)
website	L'URL del sito dell'Ente (opzionale)
address	L'indirizzo della sede dell'Ente (opzionale)
tags	La lista dei tag che descrivono il publisher (opzionale)

example

File `publisher_settings.yml` tratto dallo Starter kit

```
1 publisher:
2   name: Ministero della Documentazione Pubblica
3   description:
4     Lorem ipsum dolor sit amet, consectetur
5     adipiscing elit, sed do eiusmod tempor
6     incididunt ut labore et dolore magna aliqua.
7     Ut enim ad minim veniam, quis nostrud
8     exercitation ullamco laboris nisi ut
9     aliquip ex ea commodo consequat.
10    Duis aute irure dolor in reprehenderit in
11    voluptate velit esse cillum dolore eu
12    fugiat nulla pariatur. Excepteur sint
13    occaecat cupidatat non proident, sunt in
14    culpa qui officia deserunt mollit anim id
15    est laborum.
16  short_name: Min. Doc. Pub.
17  website: https://www.ministerodocumentazione.gov.it
18  tags:
19    - documents
20    - public
21    - amazing publisher
22  logo: assets/images/logo.svg
```

5.1.3 Metadati dei progetti

Importante: I campi **name**, **description** e **documents** sono obbligatori.

Tabella 5.2: Alcuni dei metadati associati a ciascun progetto.

Parametro	Descrizione
name	Il nome per esteso del progetto
description	Una descrizione estesa delle funzioni e degli scopi del progetto
documents	La lista dei documenti afferenti al progetto, identificati tramite i nomi dei loro repository
short_name	Il nome abbreviato del progetto o l'acronimo (opzionale)
logo	L'URL del logo (può essere contenuto nel repository di configurazione) (opzionale)
website	L'URL del sito del progetto (opzionale)
start_date	La data di inizio del progetto (opzionale)
end_date	La data di fine del progetto (opzionale)
tags	La lista dei tag che descrivono il progetto (opzionale)

example

File `projects_settings.yml` tratto dallo Starter kit

```

1 projects:
2   - name: Progetto Documentato Pubblicamente
3     description:
4       Lorem ipsum dolor sit amet, consectetur
5       adipiscing elit, sed do eiusmod tempor
6       incididunt ut labore et dolore magna aliqua.
7       Ut enim ad minim veniam, quis nostrud
8       exercitation ullamco laboris nisi ut
9       aliquip ex ea commodo consequat.
10      Duis aute irure dolor in reprehenderit in
11      voluptate velit esse cillum dolore eu
12      fugiat nulla pariatur. Excepteur sint
13      occaecat cupidatat non proident, sunt in
14      culpa qui officia deserunt mollit anim id
15      est laborum.
16     documents:
17       - project-document-doc
18       - another-project-document-doc
19     short_name: PDP
20     website: https://progetto.ministerodocumentazione.gov.it
21     tags:
22       - digital
23       - citizenship
24       - amazing project

```

5.2 Repository del documento

Dopo aver inserito il repository del documento fra quelli presenti nel file `projects_settings.yml`, è possibile importare il documento dal backend di Docs Italia. Per una corretta compilazione, il repository del documento deve contenere come minimo alcuni file specifici.

example

Repository del documento di prova

Lo *Starter kit* (pagina 7) contiene un esempio di repository del documento. Consulta le istruzioni contenute nel kit per conoscere come modificare i file.

Prima di pubblicare il documento su Docs Italia, l'utente deve:

1. Modificare il file `README.md`, descrivendo il contenuto del repository;
2. Modificare il file `index.rst` e inserire i file della documentazione;
3. Modificare il file `LICENSE`;
4. Modificare il file `document_settings.yml`.

Consulta le sezioni seguenti per maggiori informazioni su come modificare questi file.

5.2.1 Struttura del repository

Lo *Starter kit* (pagina 7) suggerisce l'uso di una struttura del repository del documento di questo tipo:

example

Struttura del repository

```
1 +-- README.md
2 +-- index.rst
3 +-- titolo-capitolo-1.rst
4 +-- titolo-capitolo-2.rst
5 +-- LICENSE
6 +-- AUTHORS
7 +-- document_settings.yml
```

Il repository dovrà comprendere, come requisito minimo:

1. Un file **README.md**, che serve da homepage del repository su GitHub e in cui viene descritto il contenuto del repository stesso.
2. Un file **index.rst**, che corrisponderà alla pagina principale del sito della documentazione.
3. I file **titolo-capitolo-1.rst** e **titolo-capitolo-2.rst**, come esempio di capitoli del documento.
4. I file **LICENSE** e **AUTHORS**, che specificano la licenza d'uso associata al repository e l'attribuzione dei diritti d'autore.
5. Un file **document_settings.yml**, che specifica i metadati associati al documento.

5.2.2 File README.md

Il file `README.md` rappresenta la prima pagina che gli utenti vedono quando accedono a un repository su GitHub. Deve fornire le informazioni sul contenuto del repository stesso e viene redatto usando la *sintassi Markdown*⁴², che differisce dal formato RST discusso in precedenza. Puoi consultare l'esempio contenuto nello *Starter kit* (pagina 7).

⁴² <https://guides.github.com/features/mastering-markdown/>

Il file README.md comincia con un titolo con il seguente formato:

```
# Nome_Progetto, Nome_Documento
```

Il segno # indica un titolo e Nome_Documento (obbligatorio) serve a distinguere diversi documenti relativi, per esempio, ad aspetti diversi di uno stesso progetto.

All'inizio del README, indica il **link alla documentazione** su Docs Italia, assieme al/ai **link ai testi di partenza**, se presenti.

5.2.3 File index.rst

Il file index.rst corrisponde alla **home page del documento** e serve ad almeno tre scopi:

1. Visualizzare il titolo del documento;
2. Fornire un'introduzione al testo;
3. Visualizzare un indice di tutte le pagine contenute.

Puoi trovare [maggiori informazioni sulla struttura dell'indice del documento](#) in Appendice.

5.2.4 Struttura del documento

I vari capitoli del documento sono contenuti in file separati allo stesso livello del file `index.rst`. Per documenti più complessi, con sezioni su più livelli, la [struttura consigliata](#) è descritta in Appendice.

5.2.5 File LICENSE e AUTHORS

Il file LICENSE specifica il tipo di licenza associata alla documentazione. Le licenze sono identificate attraverso il loro [codice SPDX](#)⁴³.

Per i documenti contenuti in Docs Italia, è obbligatorio indicare una licenza aperta. Il suggerimento è di usare una delle seguenti opzioni:

- CC-BY-4.0 per la documentazione;
- CC0-1.0 per le leggi, gli schemi e i documenti normativi.

Nel caso di licenza CC-BY-4.0 è obbligatorio creare un file AUTHORS che contiene l'attribuzione della proprietà dei diritti d'autore. Nel caso di licenza CC0-1.0 è possibile farlo (*MAY*) ma non è obbligatorio.

Per maggiori informazioni, è possibile consultare il [README per le licenze](#)⁴⁴ del Team per la Trasformazione Digitale.

5.2.6 File document_settings.yml

Il file document_settings.yml specifica i **metadati associati al documento**, in modo simile a quanto fatto in precedenza per *publisher* (pagina 20) e *progetto* (pagina 21). Il documento eredita i metadati relativi al publisher e al progetto da quelli presenti nei file del *repository di configurazione* (pagina 19), se presenti.

I metadati facilitano la ricerca delle informazioni da parte degli utenti. In particolare, anche per i documenti è previsto l'uso di **tag tratti da un vocabolario controllato** per permettere l'integrazione fra Docs Italia e il Forum (vedi anche *Repository di configurazione* (pagina 19)).

⁴³ <https://spdx.org/licenses/>

⁴⁴ <https://teamdigitale.github.io/licenses/README.it.html>

La [Tabella 5.3](#) mostra un elenco dei possibili metadati e una loro descrizione. L'esempio successivo mostra il file `document_settings.yml` contenuto nello *Starter kit* (pagina 7).

Importante: I campi `name`, `description` e `tags` sono obbligatori per la corretta compilazione del documento.

Tabella 5.3: Alcuni dei metadati associati a ciascun documento. I campi contrassegnati con l'asterisco sono obbligatori.

Parametro	Descrizione
<code>name (*)</code>	Il nome per esteso del documento
<code>description (*)</code>	Una descrizione estesa delle funzioni e degli scopi del documento
<code>tags (*)</code>	La lista dei tag che descrivono il documento
<code>short_name</code>	Il nome abbreviato del documento (opzionale)
<code>author</code>	Chi ha creato il documento, username GitHub (opzionale)
<code>contributors</code>	Chi ha collaborato alla stesura del testo: nomi e cognomi (opzionale)
<code>published</code>	La data in cui il documento è stato pubblicato per la prima volta (opzionale)
<code>expiration</code>	La data in cui il documento diventa obsoleto, utilizzabile per nascondere documenti non più validi (opzionale)
<code>id</code>	Un identificativo univoco della documentazione (opzionale)
<code>license</code>	Il tipo di licenza associato al documento (opzionale)
<code>origin</code>	L'URL del documento di partenza (opzionale)
<code>software_website</code>	L'URL del software a cui la documentazione si riferisce (opzionale)
<code>audience</code>	A chi è rivolto il documento (ad es. cittadini, comuni, software house, ecc) (opzionale)
<code>type</code>	Il tipo di documento (ad es. linee guida, documentazione tecnica, leggi, procedure, ecc) (opzionale)

example

File `document_settings.yml` tratto dallo Starter kit

```
1 document:
2   name: Titolo del documento
3   description: |
4     Lorem ipsum dolor sit amet, consectetur
5     adipiscing elit, sed do eiusmod tempor
6     incididunt ut labore et dolore magna aliqua.
7     Ut enim ad minim veniam, quis nostrud
8     exercitation ullamco laboris nisi ut
9     aliquip ex ea commodo consequat.
10    Duis aute irure dolor in reprehenderit in
11    voluptate velit esse cillum dolore eu
12    fugiat nulla pariatur. Excepteur sint
13    occaecat cupidatat non proident, sunt in
14    culpa qui officia deserunt mollit anim id
15    est laborum.
16   tags:
17     - topic
18     - related topic
19     - amazing project
```

5.3 Funzione commenti con Forum Italia

Docs Italia è completamente integrato con [Forum Italia](#)⁴⁵, la piattaforma di discussione sui progetti digitali della Pubblica Amministrazione.

È possibile aggiungere dei commenti ai propri documenti. Ciascun commento inserito su Docs Italia è automaticamente visibile anche su uno specifico topic (argomento) in Forum Italia. Viceversa, quando vengono inseriti dei commenti nel topic sul Forum, questi sono subito visibili anche nel documento su Docs Italia.

5.3.1 Aggiungere i commenti nel documento

Al momento, è possibile aggiungere uno o più thread di commenti per ciascuna pagina. Ogni thread corrisponde a uno specifico topic nel Forum.

Quando richiedi l'autorizzazione a pubblicare un documento su Docs Italia, gli amministratori creeranno per te una Categoria nel Forum dove verranno visualizzati i commenti al tuo documento.

Per ciascun argomento nel Forum, ti verrà assegnato un `<topic-id>` dagli amministratori. Per abilitare i commenti, copia lo script seguente nel punto in cui vuoi visualizzarli:

```
.. forum_italia::
   :topic_id: <topic-id>
```

sostituendo `<topic-id>` con il codice opportuno.

example

Il codice da inserire per il topic con ID 1234 è:

```
1 .. forum_italia::
2   :topic_id: 1234
```

Ripetendo questa procedura, è possibile collegare sezioni specifiche del documento con il corrispondente argomento sul Forum.

Se vuoi attivare le funzionalità di commento per l'intera pagina, dovrai aggiungere l'opzione `:scope: document` e inserire il codice seguente alla fine della pagina.

```
.. forum_italia::
   :topic_id: <topic-id>
   :scope: document
```

In caso di problemi, è possibile *contattare gli amministratori di Docs Italia* (pagina 13).

5.4 Caricare i file sul repository remoto

Tutti i file necessari alla creazione del documento su Docs Italia devono essere caricati nel repository del documento remoto, in modo che Docs Italia possa aggiornare la pagina del documento, attualmente vuota. Ogni documento su Docs Italia viene aggiornato automaticamente ogni qual volta viene effettuata una modifica al repository di configurazione o del documento.

Le *procedure da utilizzare* (pagina 49) per caricare i file sono descritte in Appendice.

⁴⁵ <https://forum.italia.it/>

5.5 Backend di Docs Italia

Docs Italia possiede una modalità di backend, ovvero una piattaforma accessibile dagli utenti autorizzati dove è possibile eseguire alcune attività di amministrazione avanzata (vedi Figura 5.1).

L'utente può accedere al backend di Docs Italia semplicemente con il proprio account GitHub.

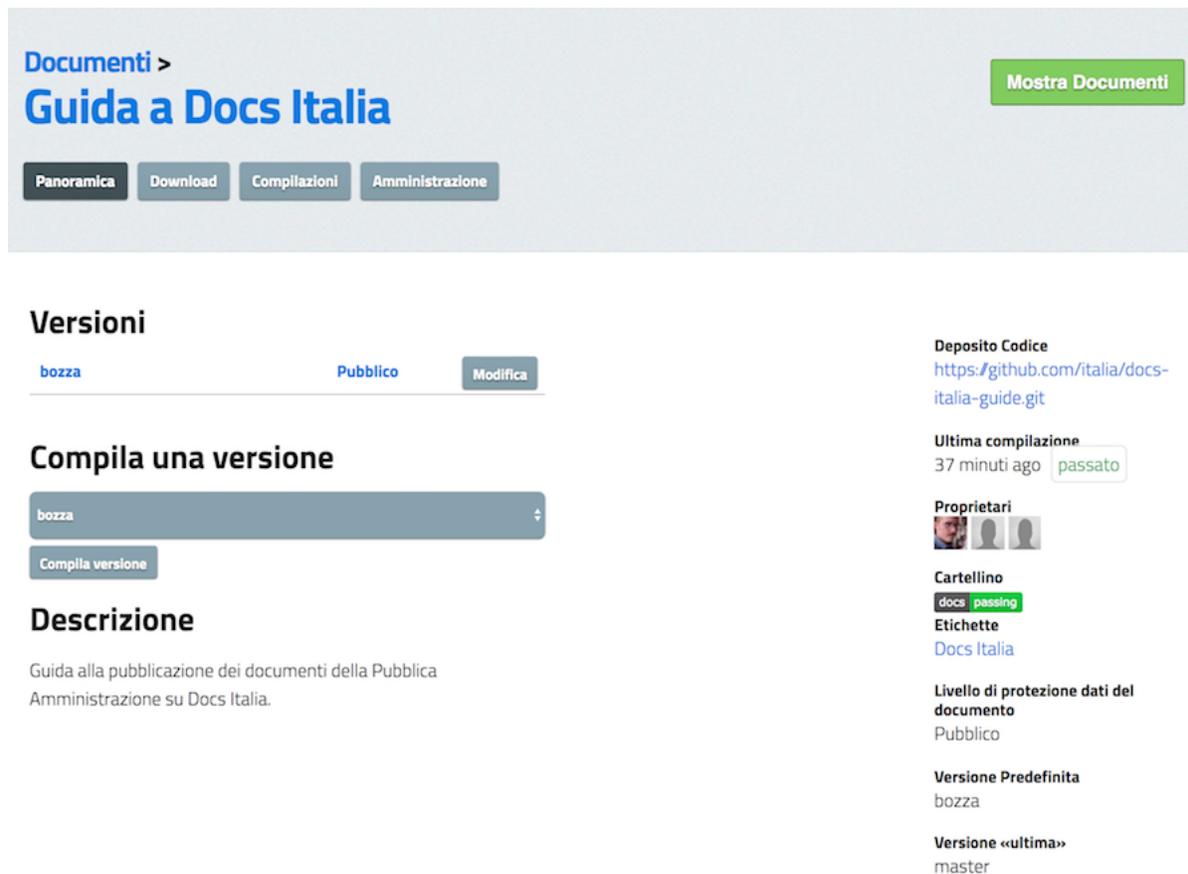


Fig. 5.1: Il backend di Docs Italia per gestire un documento.

Nota: Al primo accesso, l'utente deve autorizzare a sua volta la piattaforma a interagire con la propria organizzazione GitHub: tale operazione è necessaria per consentire l'aggiornamento del documento a ogni modifica del repository.

Il backend permette di modificare le impostazioni avanzate, come l'attivazione o meno di determinate versioni di un documento o l'aggiornamento manuale delle pagine. Da qui, l'utente può gestire tutti i documenti corrispondenti ai repository di una determinata organizzazione per i quali ha i permessi di scrittura.

Dal backend sarà possibile accedere ad alcuni widget relativi al publisher, ai progetti o ai documenti. Per esempio, la lista dei progetti, la lista dei documenti e l'indice di un documento.

5.5.1 Anteprima del documento

Docs Italia prevede la possibilità di creare un'anteprima dei documenti privata, non raggiungibile tramite i collegamenti all'interno della piattaforma, in modo da poterla controllare prima di renderla pubblica.

I documenti privati sono raggiungibili dal publisher cliccando sul pulsante **Mostra Documenti** nel backend della piattaforma. Tali documenti possono essere resi pubblici in un secondo tempo tramite un'apposita impostazione nel backend di Docs Italia.

La [Figura 5.2](#) mostra com'è possibile impostare un documento privato su Docs Italia.

Scelta Versioni Attive

Le versioni attive sotto visualizzate sul sito.

master

Attivo (Non compilato)

Privato

bozza

Attivo (Compilato)

Pubblico

Protetto

Privato

Salva

Fig. 5.2: Il backend di Docs Italia permette di impostare documenti privati.

5.5.2 Documenti in lingue multiple

Docs Italia permetterà di gestire i documenti e le loro traduzioni. L'utente potrà leggere il documento in un'altra lingua in qualsiasi pagina si trovi, semplicemente selezionando il pulsante corrispondente. Questa funzionalità verrà presto aggiunta alla piattaforma.

6.1 Visibilità del documento

Dopo aver pubblicato il documento su Docs Italia, è utile eseguire le seguenti operazioni per aumentarne la visibilità.

6.1.1 Presentazione sul proprio sito web

È opportuno inserire i link ai documenti nel proprio sito web, creando una sezione *ad hoc*, se necessario.

Docs Italia sta lavorando alla creazione di alcuni widget che permetteranno di pubblicare in modo semplice sul proprio sito la lista dei link ai propri documenti e, per ciascun testo, l'indice completo.

6.1.2 Pubblicazione sulla home page di Docs Italia

I documenti più recenti vengono visualizzati automaticamente nella home page di Docs Italia.

Aggiungere i metadati e indicare l'eventuale progetto di riferimento può aiutare gli utenti a trovare più facilmente il tuo documento.

6.1.3 Divulgazione su Forum Italia

È possibile dare visibilità ai nuovi documenti utilizzando [Forum Italia](https://forum.italia.it/)⁴⁶, il forum di discussione sui progetti digitali della Pubblica Amministrazione. Per farlo, basta andare su Forum Italia e aprire un topic (è richiesta la registrazione) in corrispondenza del Progetto cui fa riferimento il documento.

La [Figura 6.1](#) mostra l'annuncio su Forum Italia relativo alla pubblicazione del Codice dell'amministrazione digitale.

Se il documento non è associato ad alcun Progetto, è possibile usare la categoria [Community & Altro](https://forum.italia.it/c/community-feedback)⁴⁷.

⁴⁶ <https://forum.italia.it/>

⁴⁷ <https://forum.italia.it/c/community-feedback>

Codice dell'Amministrazione Digitale su Docs Italia

● Community & Altro



gpalestra Giuseppe Palestra

2 10gg

Il [Codice dell'Amministrazione Digitale](#) ²⁶ è pubblicato su [Docs Italia](#) ¹⁰ dove è possibile consultare le precedenti edizioni del documento ed anche le differenze fra le varie versioni.

Ad esempio, è possibile vedere le [differenze fra le ultime due versioni](#) ²⁷.

[Docs Italia](#) ¹⁰ è lo strumento, basato sul progetto open source Read the Docs, che usiamo per pubblicare documenti pubblici utili ai progetti della Pubblica Amministrazione.

6 "Mi piace" Rispondi

Fig. 6.1: L'annuncio su Forum Italia relativo alla pubblicazione del CAD.

6.1.4 Promozione sui canali social

Un'ulteriore promozione del documento può avvenire attraverso i **canali social**. [Twitter](#)⁴⁸ è un ottimo modo per condividere il link e aumentare il traffico verso il proprio sito.

Alcuni dei documenti verranno promossi sulla **newsletter** di [Developers Italia](#)⁴⁹. Anche in questo caso, è necessario contattare il Team per stabilire la strategia più appropriata.

6.2 Operazioni di manutenzione

Dopo la pubblicazione, sono necessarie alcune operazioni per **mantenere i documenti aggiornati**. Le modifiche possono essere necessarie a seguito di cambi nella legislazione, di segnalazioni degli utenti tramite issue su GitHub e Forum, o semplicemente per migliorare la documentazione stessa.

6.2.1 Modifiche alla documentazione

Qualsiasi modifica alla documentazione avviene modificando i file all'interno del repository GitHub. La [procedura](#) (pagina 49) è descritta in appendice. Una modifica alla documentazione effettuata tramite un commit sul repository genererà automaticamente la documentazione aggiornata su Docs Italia.

È buona pratica rivedere il contenuto dei testi con scadenza periodica, in modo da evitare che le informazioni diventino obsolete o addirittura errate.

Le modifiche sostanziali alla documentazione, a seguito per esempio di un cambiamento nella legislazione o nel software di riferimento, devono essere associate a una versione differente (vedi [l'appendice](#) (pagina 33) sul versionamento).

⁴⁸ <https://twitter.com>

⁴⁹ <http://developers.italia.it>

6.2.2 Issues e pull requests

Lo sviluppo collaborativo della documentazione tramite GitHub si avvale di due funzionalità: *issues*⁵⁰ e *pull requests*⁵¹.

Le *issues* (letteralmente “problemi”) servono a indicare ai responsabili del progetto dei possibili problemi con la documentazione, dai semplici refusi nel testo agli errori fattuali nei contenuti.

Le *pull requests* servono invece per proporre direttamente delle modifiche che, dopo una revisione da parte del publisher, possono essere integrate direttamente nel progetto.

Questi strumenti permettono agli utenti di segnalare eventuali errori e possibili correzioni, garantendo quindi un alto standard di qualità per la documentazione.

⁵⁰ <https://help.github.com/articles/about-issues/>

⁵¹ <https://help.github.com/articles/about-pull-requests/>

Appendice 1. Il versionamento

7.1 Introduzione

Il *version control*⁵² o versionamento consente di tracciare i cambiamenti occorsi a un file o a un insieme di file. Permette, tra le altre cose, di riportare i file o l'intero progetto a uno stadio precedente, visualizzare le modifiche nel corso del tempo, sviluppare più linee di lavoro in parallelo e identificare gli autori delle modifiche.

I sistemi di controllo versione sono usati abitualmente nei progetti di sviluppo software. Questi sistemi possono essere applicati anche alla documentazione: con un approccio di tipo “docs as code”⁵³ (documentazione come codice), è possibile tracciare i cambiamenti puntuali dei vari file e definire delle versioni.

7.2 Glossario minimo

repository

Il repository è una cartella in cui vengono conservati tutti i file di un progetto. Questa cartella può essere salvata localmente o ospitata su una piattaforma online come *GitHub*⁵⁴ (repository remoto).

commit

Un commit è una *fotografia* del progetto e di tutti i file in un determinato istante. Eseguire un commit significa essenzialmente congelare lo stato del progetto per poterlo recuperare in futuro.

tag

Il tag è un'etichetta che punta a uno specifico commit. Può essere usato per identificare degli stadi particolari nell'evoluzione del progetto (ad es. le *release*, ovvero i rilasci del software o della documentazione).

⁵² https://it.wikipedia.org/wiki/Controllo_versione

⁵³ <https://gdstechnology.blog.gov.uk/2017/08/25/why-we-use-a-docs-as-code-approach-for-technical-documentation/>

⁵⁴ <https://github.com/>

7.3 Tipi di versionamento

Docs Italia utilizza il sistema di controllo versione di GitHub: per ogni documento, esiste una traccia pubblica di tutte le modifiche effettuate e dei relativi autori.

Il versionamento adotta un **sistema di release basato sui tag**, che varia in base al tipo di documento.

Documentazione di un progetto software

Il versionamento del codice e della relativa documentazione vanno di pari passo.

Dal momento che non è possibile imporre una singola strategia di versionamento, le versioni della documentazione avranno formati diversi a seconda del tipo di versionamento usato per il software.

Linee guida

La versione viene indicata dall'anno e da un numero progressivo, nel formato **AAAA.N**. Ad esempio, la versione numero 1 dell'anno 2018 sarà indicata con 2018.1. È possibile *estendere il processo di versionamento* (pagina 34) alle fasi di consultazione e approvazione.

Testo legislativo

La versione del documento è determinata dalle modifiche introdotte nel corso dell'iter legislativo.

La versione viene indicata dalla data in cui è approvata una modifica al testo, nel formato **vAAAA-MM-GG**. Ad esempio, un testo modificato tramite Decreto Legislativo del 13 dicembre 2017 sarà associato alla versione v2017-12-13.

Vedi il [Codice dell'amministrazione digitale](#)⁵⁵ per un esempio.

7.4 Estensione del processo di versionamento

Per tenere conto di alcune esigenze relative al procedimento amministrativo, è possibile estendere il versionamento del documento descritto sopra. In particolare, per ciascun documento sono previste diverse fasi di sviluppo.

Documento in fase di creazione

Il documento non è ancora pubblico. Su Docs Italia esiste solo il titolo, con l'etichetta "Documento in fase di creazione". Il documento può essere contenuto in un repository pubblico o privato di GitHub. La stesura si potrebbe iniziare già su GitHub, che supporta perfettamente un approccio collaborativo.

Documento in bozza (versione alfa, con tag)

Il documento è pubblicato come bozza, in consultazione. I nuovi contenuti e le modifiche ai contenuti esistenti, dopo essere approvati, vengono pubblicati nella versione alfa del documento. Vengono resi disponibili per una discussione pubblica e una revisione da parte della community, anche se questa è priva di valore ufficiale.

Documento firmato, in attesa di parere (versione beta, con tag)

Il documento è firmato, in attesa di parere positivo. In Docs Italia sarà presente, oltre al PDF del documento, anche il PDF firmato relativo allo stesso documento in pubblicazione.

Documento approvato (release, con tag)

Il documento ha ricevuto parere positivo e viene pubblicato in via ufficiale. Su Docs Italia appare anche come **versione stable**.

Successive modificazioni (versione latest)

⁵⁵ <http://cad.readthedocs.io>

Le modifiche intermedie apportate al documento, quando non esplicitamente associate a una versione come descritto sopra, vengono indicate con la generica versione **latest**. In questo caso vengono visualizzate le modifiche più recenti, senza però che il documento abbia valore ufficiale.

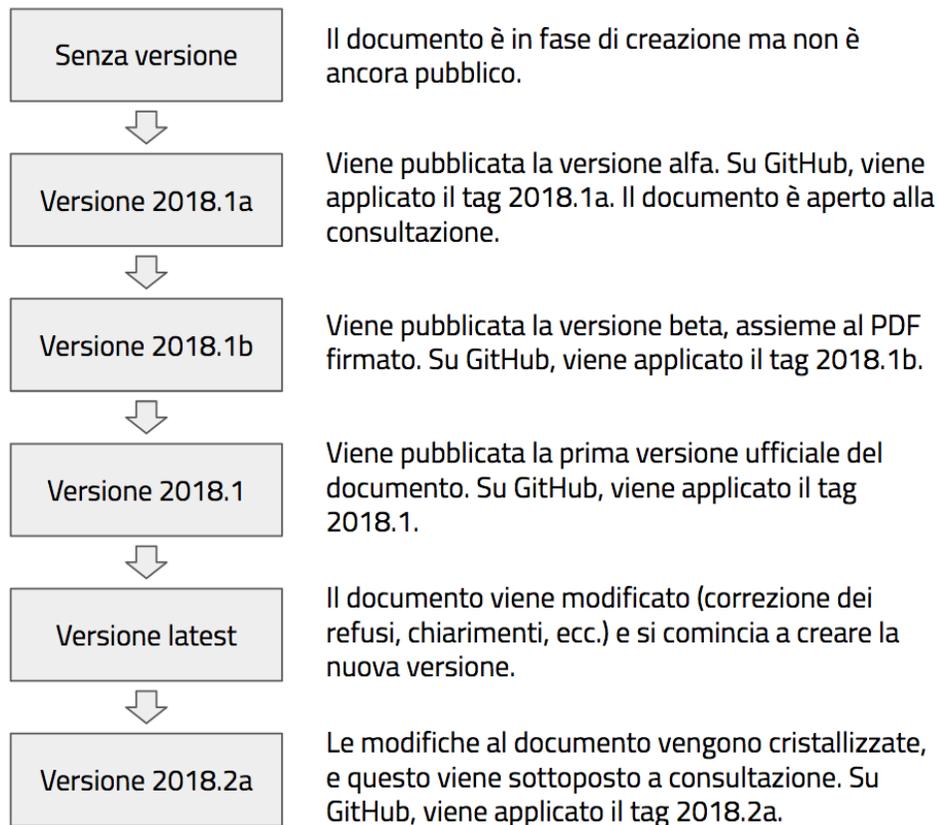


Fig. 7.1: Esempio di flusso di versionamento.

Appendice 2. Sintassi del formato RST per Docs Italia

Guida alla sintassi RST per i documenti su Docs Italia

Questa appendice spiega in modo dettagliato come sfruttare le componenti di formattazione avanzata per i documenti su Docs Italia.

Per una guida allo stile e al linguaggio da usare, è possibile consultare la [Guida al linguaggio della Pubblica Amministrazione](#)⁵⁶.

8.1 Struttura del documento

I documenti pubblicati su Docs Italia devono essere suddivisi in file differenti. Il file `index.rst` contiene il titolo del documento, un breve riassunto e l'indice dei contenuti. Il resto del documento è diviso in capitoli (sezioni di primo livello) che corrispondono ad altrettanti file.

example

Esempio di file `index.rst`

```
Titolo del documento
=====

.. highlights:

   Breve testo che riassume il contenuto del documento. Lorem ipsum dolor
   sit amet, consectetur adipiscing elit.

.. toctree::
   :numbered:

   titolo-primo-capitolo
   titolo-secondo-capitolo
```

⁵⁶ <http://guida-linguaggio-pubblica-amministrazione.readthedocs.io/it/latest/index.html>

I file che contengono le diverse sezioni di primo livello devono essere nominati con il titolo del capitolo, sostituendo tutti gli spazi con il trattino e omettendo le particelle grammaticali.

Poiché la strutturazione del repository impatta sugli URL generati, tutti i file devono essere collocati nella radice del repository.

Per documenti particolarmente complessi, che rendono necessaria una ulteriore suddivisione in file, è possibile creare una directory per ogni sezione (usando il titolo della sezione) e replicare la struttura di un documento semplice all'interno di ogni directory.

example

Esempio di struttura di un documento complesso

```
├── index.rst
├── titolo-primo-capitolo
│   ├── index.rst
│   ├── primo-paragrafo.rst
│   ├── titolo-interessante.rst
│   └── titolo-sezione.rst
├── titolo-secondo-capitolo
│   ├── index.rst
│   ├── nuova-sezione.rst
│   └── titolo-molto-molto-lungo.rst
├── titolo-terzo-capitolo
├── index.rst
├── paragrafo-rivoluzionario.rst
├── paragrafo-tecnologico.rst
└── paragrafo-tradizionale.rst
```

8.2 Formattazione e markup di base

Per le formattazioni di base del formato RST, è possibile consultare la sezione [Link utili](#) (pagina 47).

8.3 Titoli

I titoli che identificano le sezioni sono indicati sottolineando il testo con una riga di caratteri speciali, ordinati secondo una gerarchia. Su Docs Italia, la convenzione da seguire è indicata nella tabella seguente.

Tabella 8.1: Gerarchia dei simboli per i titoli

Titolo	Carattere sottolineatura	Esempio
Documento (solo index.rst)	= (uguale)	Titolo del documento =====
Titolo di capitolo (sezione di primo livello)	- (trattino)	Titolo del capitolo -----
Titolo del paragrafo (sezione di secondo livello)	~ (tilde)	Titolo del paragrafo ~~~~~
Titolo del sotto-paragrafo (sezione di terzo livello)	^ (cappelletto, caret)	Titolo del sotto-paragrafo ^^^^^^
Titolo della sezione di quarto livello	' (apice singolo)	Titolo quarto livello ''''''

8.4 Sezioni

Ciascuna sezione di primo livello (capitolo) corrispondente ad un diverso file deve contenere un breve [riassunto](#) subito dopo il titolo.

Il documento non dovrebbe contenere sezioni oltre il quarto livello.

8.5 Indice

L'indice del documento va inserito nel file `index.rst` per mezzo della direttiva di Sphinx `toctree`⁵⁷. Per generare la numerazione delle [sezioni](#), delle [tabelle](#) e delle [figure](#), è necessario usare l'opzione `:numbered:`. Il contenuto della direttiva è costituito da un elenco di tutti i file da includere. I file corrispondono alle sezioni di primo livello.

Il glossario, se presente, non va inserito nella `toctree` insieme ai file delle sezioni, ma in una `toctree` dedicata che non deve avere l'opzione `:numbered:` (perché il glossario non va numerato) e deve invece riportare le opzioni `:hidden:` e `:name: glossary_toc`. Queste opzioni sono necessarie per fare in modo che Sphinx identifichi e gestisca correttamente la sezione glossario.

example

Esempio di `toctree` con glossario

```
.. toctree::
   :numbered:

   titolo-primo-capitolo
   titolo-secondo-capitolo
```

(continues on next page)

⁵⁷ <http://www.sphinx-doc.org/en/master/usage/restructuredtext/directives.html#directive-toctree>

(continua dalla pagina precedente)

```
.. toctree::
   :name: glossary_toc
   :hidden:

   glossario
```

8.6 Riassunto

Il riassunto (*summary*) del documento e di ogni sezione va inserito usando la direttiva di RST `highlights`⁵⁸.

Esempio:

example

```
.. highlights:

   Breve testo che riassume il contenuto del documento. Lorem ipsum
   dolor sit amet, consectetur adipiscing elit.
```

8.7 Figure

Le immagini dovrebbero essere inserite all'interno di una figura (includendo quindi almeno una didascalia). Le figure vanno inserite usando la direttiva RST `figure`⁵⁹.

Per avere la possibilità di creare riferimenti alla figura nel corpo del testo, è necessario inserire l'opzione `:name:` con un nome che può essere richiamato altrove con un semplice `:ref:`testo del riferimento <name-della-figura>``.

Per ragioni di accessibilità, è obbligatorio inserire nella direttiva l'opzione `:alt:` contenente il testo alternativo della figura.

Come contenuto della direttiva `figure` è opportuno inserire una didascalia. Essa è visualizzata anche nell'indice delle figure: se non viene specificata una didascalia, nell'indice delle figure apparirà la dicitura "Nessuna descrizione".

Il nome del file contenente l'immagine dovrebbe essere facilmente comprensibile quindi si dovrebbe evitare di usare nomi come `image_01.png`. Una buona idea potrebbe essere quella di usare lo stesso testo dell'opzione `name` facendo attenzione ad usare i trattini al posto degli spazi e omettere le particelle grammaticali.

example

```
.. figure:: mappa-modello-strategico.png
   :name: mappa-modello-strategico
   :alt: La figura mostra la mappa del modello strategico. Essa è formata
         da una pila di componenti corrispondenti ai diversi capitoli del
         piano triennale.
```

(continues on next page)

⁵⁸ <http://docutils.sourceforge.net/docs/ref/rst/directives.html#highlights>

⁵⁹ <http://docutils.sourceforge.net/docs/ref/rst/directives.html#figure>

(continua dalla pagina precedente)

```
Mappa del modello strategico di evoluzione del sistema
informativo della PA
```

Si può creare un riferimento all'immagine dell'esempio con Come previsto dalla `:ref:`strategia <mappa-modello-strategico>`` del piano triennale.

8.8 Tabelle

Le tabelle vanno inserite mediante la direttiva RST `table`⁶⁰.

È obbligatorio inserire l'opzione `:name:` per avere la possibilità di creare riferimenti alla tabella nel corpo del testo.

È obbligatorio inserire una didascalia. La didascalia è visualizzata anche nell'indice delle tabelle.

Sono disponibili alcune classi (da inserire nell'opzione `class`) per stilizzare la tabella.

example

```
.. table:: Didascalia testo tabella. Enim ad minim veniam neste.
   :name: attivita-previste
```

```
+-----+-----+-----+
| Lorem  | Vivamus | Phasellus |
| ipsum sit | elementum | viverra |
| dolor est | semper | nulla ut |
| quantu | nisi    | metus    |
| ieres  | aenean  | arius    |
| numer  | vusere  | laoreet  |
|        |         | quisque  |
|        |         | rutrum   |
+-----+-----+-----+
| Maecenas | Cursus | Fusce |
| nec odio | nunc,  | vulputate |
| et ante | quis   | eleifend |
| tincidunt | gravida | sapie ves |
| tempus  | magna mi | tibulum |
|         | a libero | purus |
|         |         | quam |
+-----+-----+-----+
```

8.9 Glossario

Il glossario va inserito in unico file che poi è incluso in una `toctree` dedicata. Per generare correttamente i riferimenti ai termini di glossario è obbligatorio usare la direttiva `glossary`⁶¹ di Sphinx e la seguente struttura.

example

⁶⁰ <http://docutils.sourceforge.net/docs/ref/rst/directives.html#table>

⁶¹ <http://www.sphinx-doc.org/en/master/usage/restructuredtext/directives.html#directive-glossary>

Glossario

A

-

```
.. glossary::
```

```
  atermine
```

```
    Definizione del termine. Neque porro quisquam est, qui dolorem ipsum quia dolor sit consetetur, adipisci velit, sed quia non incidunt ut labore et dolore magnam aliquam quaerat.
```

```
    Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
```

```
  antiopam
```

```
    Neque porro quisquam est, qui dolorem ipsum quia dolor sit consetetur, adipisci velit, sed quia non numquam eius modi incidunt ut labore et dolore magnam aliquam quaerat.
```

B

-

```
.. glossary::
```

```
  btimeam
```

```
    Neque porro quisquam est, qui dolorem ipsum quia dolor sit consetetur, adipisci velit, sed quia non numquam eius modi incidunt ut labore et dolore magnam aliquam quaerat.
```

Per inserire nel testo un riferimento ad un termine di glossario è necessario usare la sintassi: `:term:`antiopam``. È possibile anche fare riferimento ad un termine di glossario senza usare esattamente la stessa parola presente nel glossario. In questo caso la sintassi è: `:term:`laudem <antiopam>``.

8.10 Citazioni

Per inserire nel testo delle citazioni da mettere in evidenza è possibile usare la direttiva di RST `epigraph`⁶². La direttiva va usata usando la struttura nell'esempio.

example

```
.. epigraph::
```

```
  Quote nel testo. Lorem ipsum dolor sit amet, adipisici elit, sed aiusmod tempo soiu incidunt labore et aliqua sumen fortes.
```

```
  -- Autore della citazione
```

⁶² <http://docutils.sourceforge.net/docs/ref/rst/directives.html#epigraph>

8.11 Riferimenti

Per creare riferimenti a figure e tabelle è possibile usare la sintassi: `:ref:`testo che riferisce <attività-previste>``.

Per creare nel corpo del testo delle ancore referenziabili è possibile usare la sintassi:

```
.. _`qui inani vivendo`:
```

Queste ancora possono essere referenziate con

```
:ref:`corrumpit mel <qui inani vivendo>`
```

In generale, è possibile usare la sintassi `ref63` per creare riferimenti ipertestuali a tutte le direttive che includono un'opzione `:name:`.

I riferimenti sono generati correttamente anche se puntano a file diversi da quello corrente.

8.12 Note

Le note nel testo possono essere inserite usando la [sintassi prevista⁶⁴](#) dallo standard RST. Anziché usare la convenzione, derivata dalla carta stampata, di inserire il testo delle note alla fine della pagina, in Docs Italia il testo delle note deve essere inserito alla fine del paragrafo che contiene la nota.

I numeri delle note vanno indicati esplicitamente per evitare che possano variare a seguito di una rigenerazione della documentazione.

example

```
Civibus facilisis vulputate ex mea, summo dicunt sed et.
Iriure graecis ei vis. Facilis petentium laboramus ad eam, id alii
qui ex dolores vulputate scribentur [1]_. Nam no esse eleifend, vis
dignissim. Vim ad augue vidisse, adhuc everti eos te, sea blandit
adversarium ne.
```

```
.. [1] Cfr. art. 101 del Codice degli appalti d.lgs. 18 aprile 2016
n. 50.
```

8.13 Box

Nel corpo dei documenti è possibile inserire i seguenti tipi di box.

⁶³ <http://www.sphinx-doc.org/en/master/usage/restructuredtext/roles.html#role-ref>

⁶⁴ <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html#footnotes>

Tabella 8.2: Tipi di box in Docs Italia

Box	Sintassi
Nota	<code>.. note::</code>
Errore	<code>.. error::</code>
Suggerimento	<code>.. hint::</code>
Attenzione	<code>.. attention::</code>
Importante	<code>.. important::</code>
<i>Approfondimento</i> (pagina 45) (il box, che può essere espanso, mostra solo una parte del testo)	<code>.. admonition:: deepening : class: admonition-deepening display- ↔page</code>
Esempio	<code>.. admonition:: example : class: admonition-example display- ↔page .. role:: admonition-internal-title : class: admonition-internal-title `Titolo interno al box`:admonition- ↔internal-title:</code>
RFC 2119: la semantica di questi box è quella espressa nel Request for Comments 2119 ⁶⁵ .	<code>.. admonition:: must .. admonition:: should .. admonition:: may .. admonition:: must-not .. admonition:: should-not</code>
Usa e non usare	<code>.. admonition:: use .. admonition:: use-not</code>
Generico con titolo arbitrario	<code>.. admonition:: titolo del box</code>

⁶⁵ <https://www.ietf.org/rfc/rfc2119.txt>

8.13.1 Il box Approfondimento

Nel box Approfondimento è possibile decidere quale parte del contenuto è sempre visibile e quale deve essere mostrata soltanto con il pulsante «Mostra tutto». È possibile indicare il contenuto nascosto con la direttiva `.. container:: more`, come nell'esempio seguente.

Nel caso in cui non venga usato questo parametro, verranno mostrati automaticamente soltanto i primi quattro elementi del contenuto (paragrafi `<p>`, elenchi `` o altro tipo di tag).

```
.. admonition:: deepening
   :class: admonition-deepening display-page

   Paragrafo di testo visibile.

   Paragrafo di testo visibile.

   .. container:: more

      Paragrafo di testo nascosto. Può essere mostrato cliccando su "Mostra tutto".

      Paragrafo di testo nascosto. Può essere mostrato cliccando su "Mostra tutto".
```

Questo codice genera il seguente risultato.

deepening

Paragrafo di testo visibile.

Paragrafo di testo visibile.

Paragrafo di testo nascosto. Può essere mostrato cliccando su «Mostra tutto».

Paragrafo di testo nascosto. Può essere mostrato cliccando su «Mostra tutto».

8.14 Componente “In consultazione”

Codice:

```
.. admonition:: consultation

   Questo documento è in consultazione pubblica dal 3 aprile al 4
   maggio 2018, come previsto dall'art. 71 comma 1 del decreto
   legislativo 7 marzo 2005, n. 82. Ogni commento verrà preso in
   considerazione per la stesura finale.
```

8.15 Componente “Procedura”

Codice:

```
.. topic:: Procedura
   :class: procedure

   1. Assicurati di avere tutti i file necessari elencati nella
```

(continues on next page)

(continua dalla pagina precedente)

```

sezione precedente e visita la pagina del repository su
GitHub;

2. Clicca sul pulsante *Clone or download*;
3. Clicca sul pulsante *Copy to clipboard* accanto all'URL del
   repo;

.. image:: images/github_example.png

.. role:: procedure-internal-title
   :class: procedure-internal-title

:procedure-internal-title:`Da linea di comando, esegui`

1. :code:`cd` alla cartella con i file della documentazione
2. :code:`git clone <URL>`, dove <URL> è l'URL del repo. Puoi
   ottenerlo facendo semplicemente incolla (CTRL+V oppure CMD+V)
3. :code:`git add *`
4. :code:`git commit`
5. All'apertura dell'editor di testo, scrivi il messaggio di
   commit, secondo le modalità descritte nella sezione `Messaggi
   di commit`_
6. :code:`git push origin master`

```

8.16 Componente “Documenti utili”

Codice:

```

.. topic:: Documenti utili
   :class: useful-docs

- :mimetype:`application/pdf` `Predisposizione e invio del file
  di pre-subentro [204kb] <https://www.example.com>`_

- :mimetype:`application/pdf` `Guida rapida per il censimento
  degli utenti e delle postazioni per i Comuni [82kb]
  <https://www.example.com>`_

- :mimetype:`text/html` `Piano dei test di integrazione (Apertura
  nuova finestra) <https://www.example.com>`_

```

8.17 Componente “Tono di voce”

Codice:

```

.. topic:: Titolo del Q&A fra utente e amministrazione
   :class: question-and-answers

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit
sed quia consequuntur magni dolores eos qui ratione voluptatem
nesciunt.

```

(continues on next page)

(continua dalla pagina precedente)

```

.. pull-quote:: utente

    Quote di domanda dell'utente. Lorem ipsum dolor sit amet,
    sed aiusmod tempo soiu incidunt labore et aliqua sumen fortes.

    - :term:`Keyword 1`
    - :term:`Keyword 2`
    - :term:`Keyword 3`
    - :term:`Keyword 4`

.. pull-quote:: amministrazione

    Quote di risposta dall'Amministrazione. Quod sale ius, id
    conclusionemque mea, vis no ipsum quaeque minimum tritani.

    - :term:`Keyword 1`
    - :term:`Keyword 2`
    - :term:`Keyword 3`

.. glossary::

    Keyword 1
    Neque porro quisquam est, qui dolorem ipsum quia dolor sit
    consectetur, adipisci velit, sed quia non numquam eius modi
    incidunt ut labore et dolore magnam aliquam quaerat.

    Keyword 2
    Neque porro quisquam est, qui dolorem ipsum quia dolor sit
    consectetur, adipisci velit, sed quia non numquam eius modi
    incidunt ut labore et dolore magnam aliquam quaerat.

    Keyword 3
    Neque porro quisquam est, qui dolorem ipsum quia dolor sit
    consectetur, adipisci velit, sed quia non numquam eius modi
    incidunt ut labore et dolore magnam aliquam quaerat.

    Keyword 4
    Neque porro quisquam est, qui dolorem ipsum quia dolor sit
    consectetur, adipisci velit, sed quia non numquam eius modi
    incidunt ut labore et dolore magnam aliquam quaerat.

```

8.18 Link utili

- Homepage di reStructuredText⁶⁶
- Quickreference di RST⁶⁷
- Markup specification di RST⁶⁸
- Direttive di RST⁶⁹

⁶⁶ <http://docutils.sourceforge.net/rst.html>

⁶⁷ <http://docutils.sourceforge.net/docs/user/rst/quickref.html>

⁶⁸ <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html>

⁶⁹ <http://docutils.sourceforge.net/docs/ref/rst/directives.html>

- [Direttive di Sphinx](#)⁷⁰

⁷⁰ <http://www.sphinx-doc.org/en/master/usage/restructuredtext/directives.html>

Appendice 3. Procedure e convenzioni su GitHub

9.1 Procedure di caricamento sul repository remoto

Per caricare i file presenti in un repository locale in un repository remoto, sono disponibili due strategie:

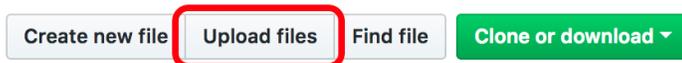
1. Upload tramite interfaccia grafica sul sito github.com⁷¹;
2. Upload da repository Git locale tramite i comandi *clone* e *push*.

Il primo metodo è adatto per chi ha poca familiarità con gli strumenti di controllo versione, mentre il secondo consente maggiore flessibilità ed è adatta a utenti mediamente esperti.

9.1.1 Upload tramite interfaccia grafica

Procedura

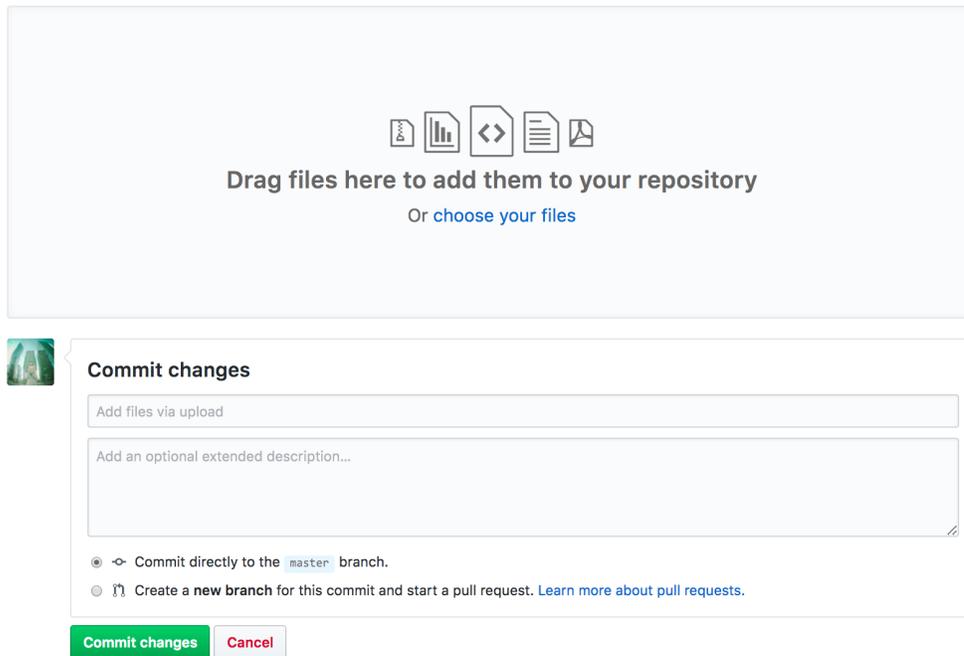
1. Assicurati di avere tutti i file necessari elencati nella sezione precedente
2. Visita la pagina del repository su GitHub
3. Clicca sul pulsante **Upload files**



4. Clicca su **choose your files** e seleziona tutti i file che intendi caricare
5. Nel riquadro “Commit changes”, specifica un oggetto del commit nel primo box, e opzionalmente un testo di spiegazione, secondo le modalità descritte nella sezione *Messaggi di commit* (pagina 51) |

⁷¹ <https://github.com/>

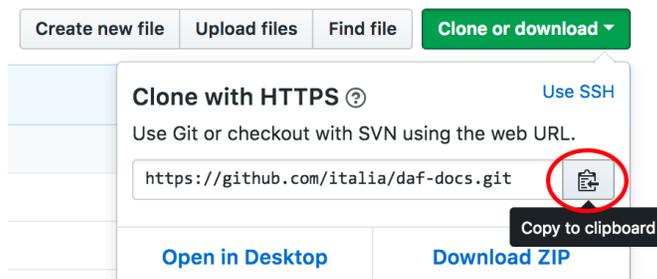
6. Clicca sul pulsante **Commit changes**



9.1.2 Upload da un repository Git locale

Procedura

1. Assicurati di avere tutti i file necessari elencati nella sezione precedente
2. Visita la pagina del repository su GitHub
3. Clicca sul pulsante **Clone or download**
4. Clicca sul pulsante **Copy to clipboard** accanto all'URL del repo



Da linea di comando, esegui

1. `cd` alla cartella con i file della documentazione
2. `git clone <URL>`, dove `<URL>` è l'URL del repo. Puoi ottenerlo facendo semplicemente incolla (CTRL + V oppure CMD + V)
3. `git add *`

4. `git commit`
5. All'apertura dell'editor di testo, scrivi il messaggio di commit, secondo le modalità descritte nella sezione *Messaggi di commit* (pagina 51)
6. `git push origin master`

9.2 Messaggi di commit

Ogni volta che si effettua una modifica nel repository, è necessario utilizzare un commit. Questo viene accompagnato da un messaggio che descrive le modifiche apportate.

Il messaggio di commit si compone di due parti:

1. oggetto del commit (obbligatorio)
2. testo di spiegazione del commit (opzionale)

L'**oggetto del commit** è sempre obbligatorio e indica in maniera succinta le modifiche apportate al testo o al codice.

- Indica *cosa* hai fatto, non *come* o *perché*.
- Usa uno stile diretto e conciso, spiegando con un'unica frase il commit.
- Elimina gli articoli e le preposizioni, se necessario (se la frase è troppo lunga).
- Un buon oggetto di commit dovrebbe completare la frase: "Con questo commit, ho...".

example

Con questo commit, ho...

- modificato la funzione,
- corretto il bug, migliorato lo stile,
- rimosso variabili inutilizzate,
- aggiunto paragrafo dopo introduzione

Nell'oggetto del commit si dovrebbe indicare il tipo di commit fra i seguenti:

- Docs: modifiche alla documentazione
- Stile: formattazione, riformulazione di frasi, ecc
- Struttura: modifiche alla struttura del testo
- Refusi: correzione di piccoli refusi

example

Oggetto del commit

- Stile: diviso frase troppo lunga
- Docs: creato documentazione
- Struttura: aggiunto abstract prima dell'introduzione

Il **testo di spiegazione** del commit è opzionale, e può essere usato per fornire ulteriori dettagli riguardo alle modifiche effettuate. Dev'essere separato dall'oggetto del commit da una linea vuota.

Se il commit risolve una o più issue, è obbligatorio indicarne il numero all'interno del testo di spiegazione.