
Python

28 mag 2020

Indice dei contenuti

1	Che cos'è Docs Italia	3
1.1	Che cos'è	3
1.2	Obiettivi	3
1.3	Caratteristiche	3
1.4	Architettura dell'informazione	5
2	Starter kit	9
3	Come partecipare	11
3.1	Prerequisiti	11
3.2	Come pubblicare un documento	12
3.3	Amministratori di Docs Italia e assistenza	14
4	Scrivere un documento	17
4.1	Il formato reStructuredText (RST)	17
4.2	Creazione di documenti in formato RST	18
4.3	Migrazione su Docs Italia di documentazione esistente	18
5	Pubblicare un documento	21
5.1	Repository di configurazione	21
5.2	Repository del documento	24
5.3	Funzione commenti con Forum Italia	28
5.4	Caricare i file sul repository remoto	29
5.5	Backend di Docs Italia	29
6	Gestione dei documenti	33
6.1	Visibilità del documento	33
6.2	Operazioni di manutenzione	34
7	Appendice 1. Il versionamento	37
7.1	Introduzione	37
7.2	Glossario minimo	37
7.3	Tipi di versionamento	37
7.4	Estensione del processo di versionamento	38
8	Appendice 2. Guida di stile	41
8.1	Organizzazione dei contenuti	41

8.2	Stile e linguaggio	42
9	Appendice 3. Procedure e convenzioni su GitHub	45
9.1	Procedure di caricamento sul repository remoto	45
9.2	Messaggi di commit	47

Guida alla pubblicazione dei documenti della Pubblica Amministrazione su [Docs Italia](#).

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](#) in collaborazione con [AgID](#) 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](#) e [GitHub Italia](#), 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](#) usando il canale [#docs-italia](#).

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](#) e le linee guida (come le [Linee guida di design per i servizi web della PA](#)).

La [Figura 1.1](#) mostra 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:

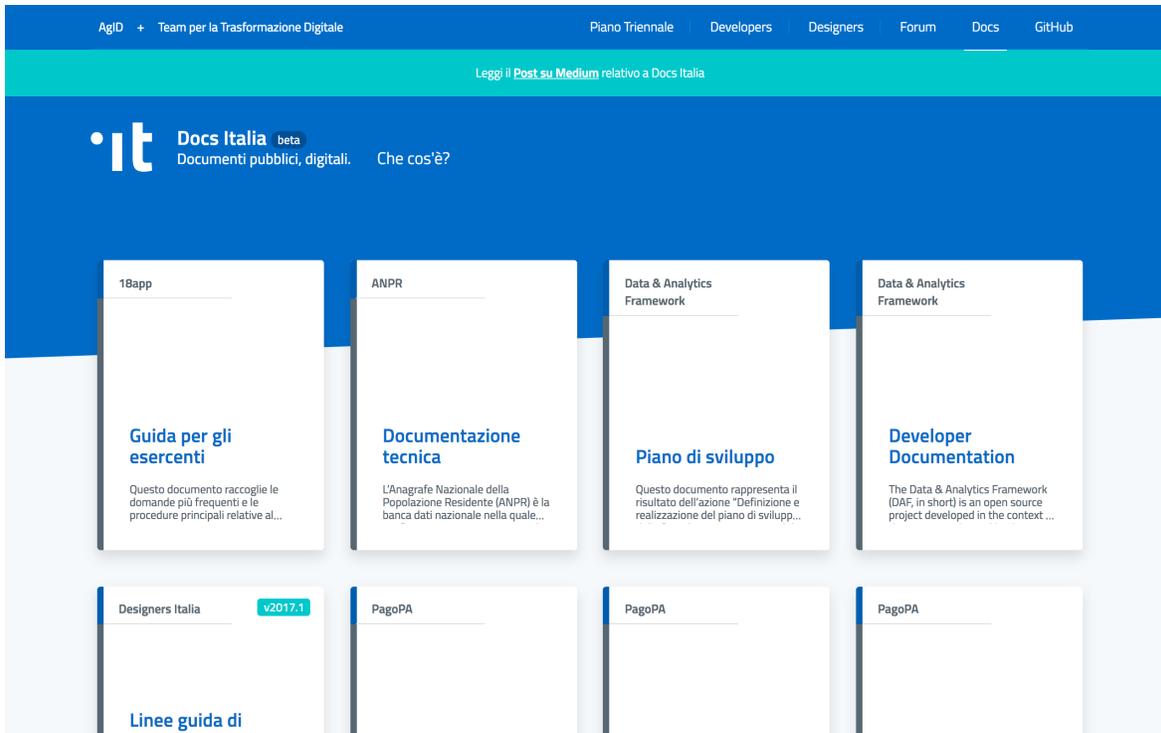


Fig. 1: L'attuale home page di Docs Italia.

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**, 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 sarà organizzato per publisher (l'ente o il Ministero che pubblica i documenti) e per progetti (ciascun progetto potrà avere diversi documenti).
- Docs Italia avrà 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)
 - l'elenco dei documenti appartenenti a uno specifico progetto (ad es. ANPR)
 - l'indice del documento

- Attualmente è in corso di analisi lo sviluppo di una componente per [LibreOffice](#) che permetta l'esportazione e l'importazione di documenti in formato RST.
- Docs Italia sarà ospitata all'URL [docs.italia.it](#). Ciascun documento sarà raggiungibile all'indirizzo: `docs.italia.it/<publisher>/<progetto>/<nome-documento>`.

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

1.4 Architettura dell'informazione

1.4.1 Pagina documento

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

1. È sempre possibile scegliere la versione del documento tramite un apposito pulsante;
2. L'utente può effettuare una ricerca specifica all'interno del documento;
3. L'indice nella barra laterale consente di passare rapidamente da una sezione all'altra;
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.

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 (vedi [Figura %s](#));
- Le *àncore* in corrispondenza dei capitoli consentono di ottenere facilmente il link di un paragrafo (vedi [Figura %s](#)). Per fare apparire l'*àncora* basta spostare il puntatore del mouse sopra al titolo corrispondente;
- Le funzionalità dei box colorati e delle note permettono di segnalare informazioni importanti, esempi, casi di studio e interpretazioni del testo (vedi [Figura %s](#)).

1.4.2 La piattaforma

- La piattaforma di Docs Italia è basata sui progetti open source [Sphinx](#) e [Read the Docs](#). I documenti pubblicati sono redatti in formato `reStructuredText`.
- 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](#) (a sua volta basato sul progetto open source [Discourse](#)) per consentire di commentare i documenti.

The screenshot shows a document page on the Docs Italia website. The page title is "Linee guida di design per i servizi web della PA". The page is annotated with five red boxes and numbers:

- 1**: A dropdown menu for "Versione" (Version) set to "stable".
- 2**: A search bar labeled "Cerca nel documento" (Search in document).
- 3**: A sidebar menu with several items, including "Menu Item level 02 (active section)" and "Menu Item level 02 (active item)".
- 4**: A "Download" section with buttons for "HTML", "PDF", and "Epub".
- 5**: A "modifica su GitHub" (edit on GitHub) button.

The main content area contains the following text:

Indice » Linee guida di design per i servizi web della PA

H1 Titolo del documento

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Magna pars studiorum, prodita quaerimus. Pellentesque habitant morbi tristique senectus et netus. Cum ceteris in veneratione tui montes, nascetur mus. Nihilne te nocturnum praesidium Palati, nihil urbis vigiliae. Integer legentibus erat a ante historiarum dapibus. Ut enim ad minim veniam, quis nostrud exercitation. Inmensae subtilitatis, obscuris et malesuada fames.

H2 Paragrafo del documento

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Magna pars studiorum, prodita quaerimus. Pellentesque habitant morbi tristique senectus et netus.

H3 Titolo del documento

Data	servizio/ oggetto	documenti aggiornati	Tipo modifica
2017-01-17 00:00:00	6001	6001-RichiestaCertificati.xls	eliminazione nota sull'obbligatorietà del doc di riconoscimento
2017-01-17 00:00:00	docRiconoscimento	docRiconoscimento.xls	allineamento file xls al tracciato xsd: campo docRiconoscimento (max 40)
2017-01-17 00:00:00			allineamento file xls al tracciato

Fig. 2: Una pagina documento su Docs Italia.

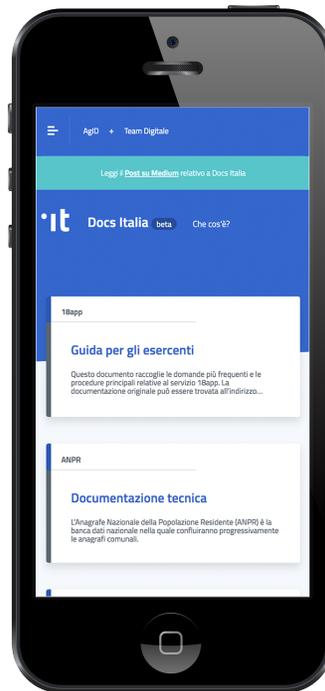


Fig. 3: Ambiente *responsive* della piattaforma Docs Italia.

Condizioni d'uso

[Link a questa intestazione](#)

1. Oggetto del servizio

Il presente servizio è volto a consentire agli esercenti commerciali, di incassare i buoni connessi all'utilizzo del bonus del valore di 500€ a coloro che abbiano compiuto diciott'anni nel 2016 e nel 2017 e di vedersene liquidare il controvalore dal Ministero dei beni e delle attività culturali e del turismo che è fornitore del presente servizio.

Fig. 4: Ancora in corrispondenza di un titolo.

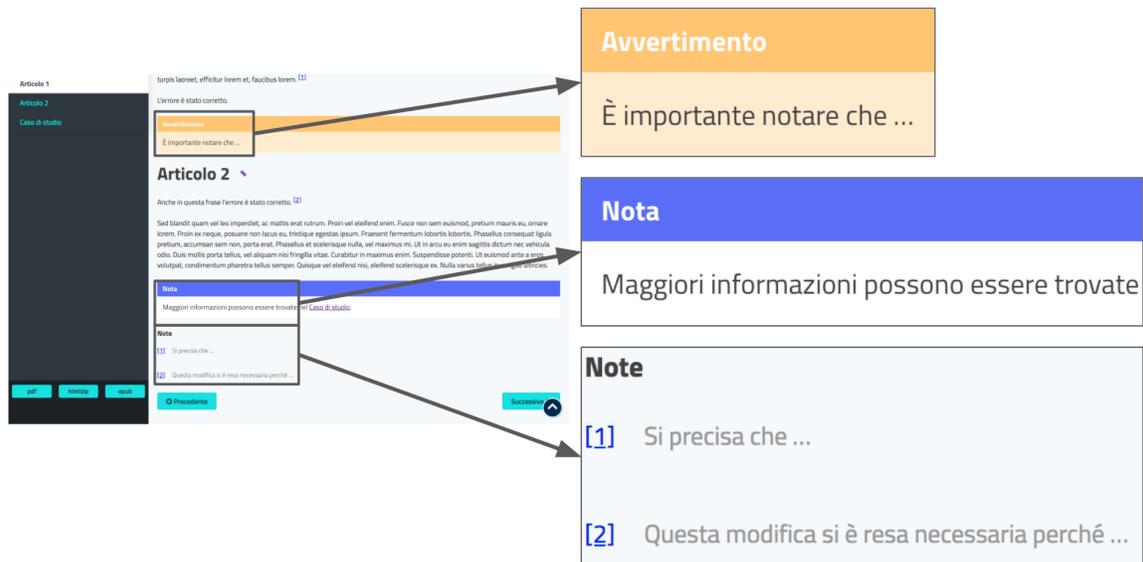


Fig. 5: Box e note in una pagina del documento.

Starter kit

Prima di cominciare la lettura di questa guida, [scarica lo Starter kit](#).

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.

3.1 Prerequisiti

Docs Italia si basa su [GitHub](#) 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](#).

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

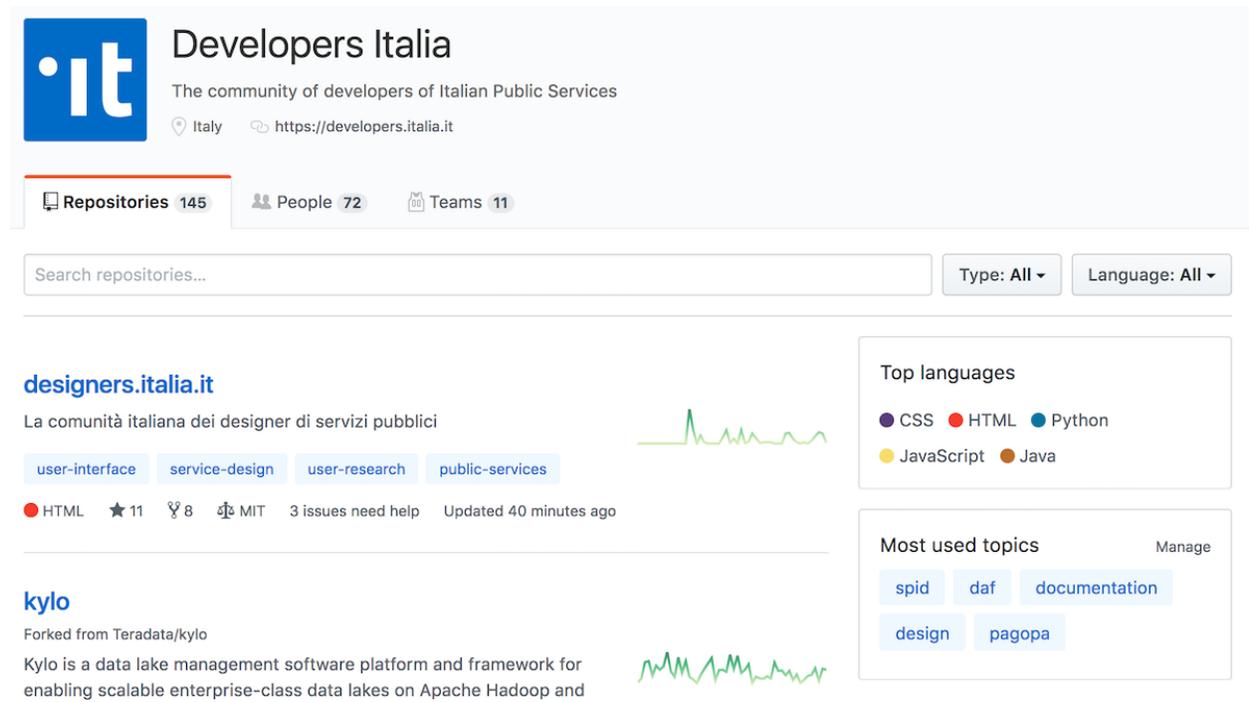
L'Ente pubblico deve essere associato a un'[organizzazione su GitHub](#).

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

Un'organizzazione è un insieme di repository gestiti da più utenti GitHub che collaborano a progetti comuni.

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.



Developers Italia
The community of developers of Italian Public Services
Italy <https://developers.italia.it>

Repositories 145 People 72 Teams 11

Search repositories... Type: All Language: All

designers.italia.it
La comunità italiana dei designer di servizi pubblici

user-interface service-design user-research public-services

HTML ★ 11 8 MIT 3 issues need help Updated 40 minutes ago

kylo
Forked from Teradata/kylo
Kylo is a data lake management software platform and framework for enabling scalable enterprise-class data lakes on Apache Hadoop and

Top languages
CSS HTML Python JavaScript Java

Most used topics Manage
spid daf documentation design pagopa

Fig. 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* per richiedere un repository nell'organizzazione di Developers Italia.

3.1.3 Conoscenza di base su Git

Docs Italia usa il [version control system](#) di Git, 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](#) 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;
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 2). Di fianco a **Policy** apparirà la dicitura **No restrictions**.

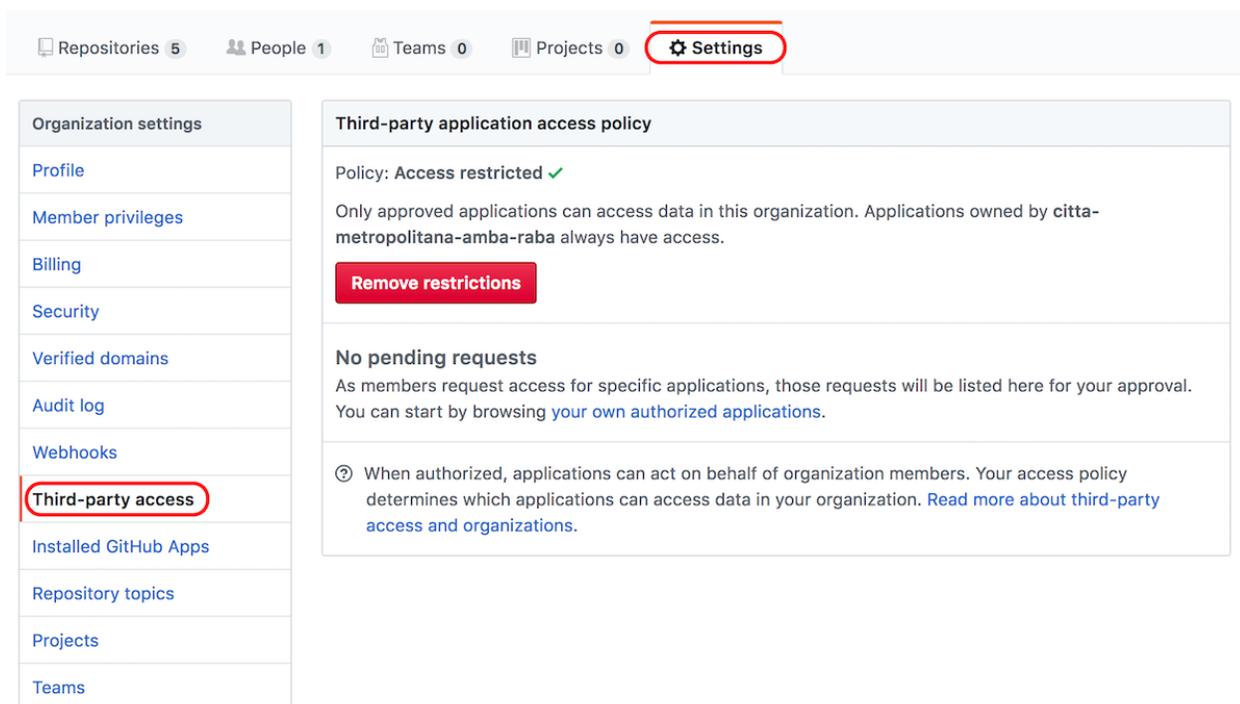


Fig. 2: Come rimuovere le restrizioni sulle applicazioni di terze parti per un'organizzazione su GitHub. Questo passaggio è necessario per permettere agli utenti di importare i repository dei documenti su Docs Italia.

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* sono disponibili nel capitolo dedicato alla pubblicazione. Un esempio di repository di configurazione si trova nello *Starter kit* fornito.
2. Un amministratore dell'account organizzazione invia una richiesta di autorizzazione alla pubblicazione su Docs Italia tramite **Slack di Developers Italia**, usando il canale **#docs-italia**.
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*. 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*. Un esempio completo di repository del documento è contenuto nello *Starter kit*.

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.

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*.

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](#) usando il canale [#docs-italia](#).

4.1 Il formato reStructuredText (RST)

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

L'esempio nella [Tabella 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](#), oppure la [lista completa delle specifiche del linguaggio](#).

Tabella 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>

Per maggiore chiarezza, puoi consultare un [documento di esempio](#) 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](#) 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++](#) 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 che permette di creare, modificare e visualizzare istantaneamente documenti in formato `.rst` secondo la logica WYSIWYG (What You See Is What You Get).

Tabella `.rst`

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

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](#), 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](#)). 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*. Utilizza la sintassi RST per titoli, sottotitoli, liste e link.
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](#).

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*). Correggi uno a uno gli errori di formattazione che si presentano, assicurandoti di rispettare la sintassi dei documenti .rst.

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

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**, 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.

Pubblicare 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

Nota: Questa sezione è riservata ai publisher che pubblicano documenti tramite la propria organizzazione. Se utilizzi un account utente, invece, le operazioni descritte qui verranno effettuate per tuo conto da parte degli amministratori di Docs Italia.

Il repository di configurazione dovrà avere un nome specifico 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*) usando *una delle due procedure descritte in Appendice*.

example

Repository di configurazione di prova

Lo *Starter kit* 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 essere chiamato **italia-conf** e 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*.

5.1.2 Metadati del publisher

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

Tabella 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.
```

(continues on next page)

(continua dalla pagina precedente)

```

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 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_setting.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

```

(continues on next page)

```
23 - citizenship
24 - amazing project
```

5.2 Repository del documento

Docs Italia genera automaticamente la pagina del documento e il relativo URL alla creazione del repository. **Se il repository del documento è vuoto, tuttavia, la pagina associata risulterà vuota** e verrà visualizzato il messaggio “Documento in fase di creazione”. La pagina del documento verrà creata automaticamente non appena l’utente caricherà nel repository tutti i file necessari alla compilazione, come descritto qui di seguito.

example

Repository del documento di prova

Lo *Starter kit* 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 la cartella _docs, inserendo i file della documentazione;
3. Modificare il file conf.py con le informazioni sul documento;
4. Modificare il file LICENSE;
5. 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* suggerisce l’uso di una struttura del repository del documento di questo tipo:

example

Struttura del repository

```
1 +-- README.md
2 +-- index.rst
3 +-- _docs/
4 | +-- _img/
5 | | +-- cap1_img1.jpg
6 | | +-- cap2_img1.jpg
7 | | ...
8 | +-- cap1.rst
9 | +-- cap2.rst
10 | ...
11 +-- conf.py
12 +-- LICENSE
13 +-- AUTHORS
```

(continues on next page)

(continua dalla pagina precedente)

```
14 +-- requirements.txt
15 +-- 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. Una cartella **_docs**, che contiene tutti i file .rst ed eventuali immagini relativi alla documentazione.
4. Un file **conf.py**, che specifica i metadati necessari alla compilazione della documentazione.
5. Un file **requirements.txt**, necessario alla compilazione e che non necessita di essere modificato.
6. I file **LICENSE** e **AUTHORS**, che specificano la licenza d'uso associata al repository e l'attribuzione dei diritti d'autore.
7. 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](#), che differisce dal formato RST discusso in precedenza. Puoi consultare l'esempio contenuto nello [Starter kit](#).

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.

Per i primi due punti, la stesura deve seguire le considerazioni fatte in precedenza sulla [sintassi RST](#).

Per quanto riguarda il terzo punto, invece, l'inclusione delle pagine nel documento avviene tramite la [direttiva toctree](#). Prendendo ad esempio la [struttura del repository illustrata in precedenza](#), un possibile index.rst viene costruito come di seguito.

example

index.rst con toctree

```

1  Il titolo del documento
2  =====
3
4  Una breve introduzione al testo.
5
6  .. toctree::
7
8     _docs/cap1.rst
9
10    _docs/cap2.rst
11
12    ...

```

5.2.4 Cartella `_docs`

Questa cartella contiene i file della documentazione creati come indicato nel capitolo *Scrivere un documento*. Ciascun file contenuto nella cartella verrà convertito in una pagina a sé stante, e collegato alla pagina principale tramite la direttiva `toctree` illustrata nella sezione precedente.

5.2.5 File `conf.py`

Questo file è presente in ogni repository del documento e costituisce il principale script da modificare per personalizzare il documento. I parametri che tipicamente l'utente deve modificare sono contenuti nella seguente tabella.

Tabella 3: Parametri del file `conf.py`.

Parametro	Descrizione
<code>project</code>	Il nome del documento
<code>copyright</code>	Il tipo di copyright
<code>version</code>	La versione ridotta del documento
<code>release</code>	La versione completa del documento (incluso <i>alfa</i> , <i>beta</i> , ecc)
<code>language</code>	La lingua del documento

Un esempio completo di file `conf.py` è contenuto nel repository del documento all'interno dello *Starter kit*.

5.2.6 File `LICENSE` e `AUTHORS`

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

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](#) del Team per la Trasformazione Digitale.

5.2.7 File requirements.txt

Il file requirements.txt contiene le informazioni riguardo al tema di stile delle pagine. Nel file **non deve** essere specificata la release del tema, in modo che il documento benefici sempre dell'ultima versione disponibile. In questo modo, il versionamento del documento (in particolare del suo contenuto) rimane indipendente dal versionamento del tema.

Tipicamente, l'utente non ha bisogno di modificare questo file, e può usare semplicemente quello contenuto nello Starter kit.

5.2.8 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* e *progetto*. Il documento eredita i metadati relativi al publisher e al progetto da quelli presenti nei file del *repository di configurazione*, 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*).

La Tabella 4s mostra un elenco dei possibili metadati e una loro descrizione. L'esempio successivo mostra il file *document_settings.yml* contenuto nello *Starter kit*.

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

Tabella 4: Alcuni dei metadati associati a ciascun documento. I campi contrassegnati con l'asterisco sono obbligatori.

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

example

File projects_setting.yml tratto dallo Starter kit

```

1 document:
2   name: Titolo del documento

```

(continues on next page)

(continua dalla pagina precedente)

```
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

Per attivare la funzione commenti, inserisci questo script (e gli id necessari, che ti verranno forniti dagli amministratori di Docs Italia) in ciascuna delle sezioni che vuoi rendere commentabili. I commenti saranno visibili anche su Forum Italia.

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

Tramite le funzionalità di [Discourse](#), è possibile aggiungere dei commenti ai propri documenti. Ciascun commento inserito su Docs Italia è automaticamente visibile anche su uno specifico topic 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 un thread di commenti per ciascuna pagina. Questo corrisponde a un singolo 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 creare i commenti in una pagina, copia lo script seguente alla fine del file RST corrispondente:

```
.. discourse::
:topic_identifier: <topic-id>
```

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

example

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

```
1 .. discourse::  
2  
3 :topic_identifier: 1234
```

Ripetendo questa procedura, è possibile collegare ciascuna pagina del documento con il corrispondente argomento sul Forum. In caso di problemi, è possibile *contattare gli amministratori di Docs Italia*.

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* per caricare i file sono descritte in Appendice.

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 %s).

L'utente potrà accedere al backend di Docs Italia semplicemente con il proprio account GitHub. Durante l'iniziale fase beta, l'accesso avviene tramite un [account su Read the Docs](#). Le funzioni descritte di seguito, però, non variano.

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 potrà 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 Nome del documento

Nota: Le istruzioni contenute in questa sezione sono valide finché non entrerà in vigore il nuovo formato per gli URL presentato nella sezione *Nuove funzionalità*.

Nel backend di Docs Italia, è possibile impostare il nome che appare nell'URL del documento. Scegli lo stesso nome utilizzato per il nome del repository del documento, senza la parte finale **-docs**.

Così, per il documento "Istruzioni per il cambio di residenza" ospitato nel repository **anpr-cambioresidenza-docs**, il nome nell'URL diventa **anpr-cambioresidenza**.

Projects >
CAD View Docs

Overview Downloads Search Builds Versions Admin

Versions

v2017-12-13	Public	Edit
v2016-08-26	Public	Edit
v2012-10-18	Public	Edit
v2010-12-30	Public	Edit
v2009-06-18	Public	Edit
v2006-04-04	Public	Edit
v2005-03-07	Public	Edit

Build a version

v2017-12-13

Build

Description

Repository
<https://github.com/italia/cad-docs>

Last Built
3 weeks, 1 day ago failed

Owners


Badge
docs passing

Tags
Project has no tags. Add some in your [project settings](#).

Project Privacy Level
Public

Short URLs
cad.readthedocs.io
cad.rtdf.io

Default Version
v2017-12-13

Fig. 1: Il backend di Docs Italia per gestire un documento.

5.5.2 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 usando URL predittivi (ad es. docs.italia.it/<publisher>/test/<nome_repo>). Tali documenti possono essere resi pubblici in un secondo tempo tramite un'apposita impostazione nel backend di Docs Italia.

La Figura %s mostra com'è possibile impostare un documento privato su Docs Italia.

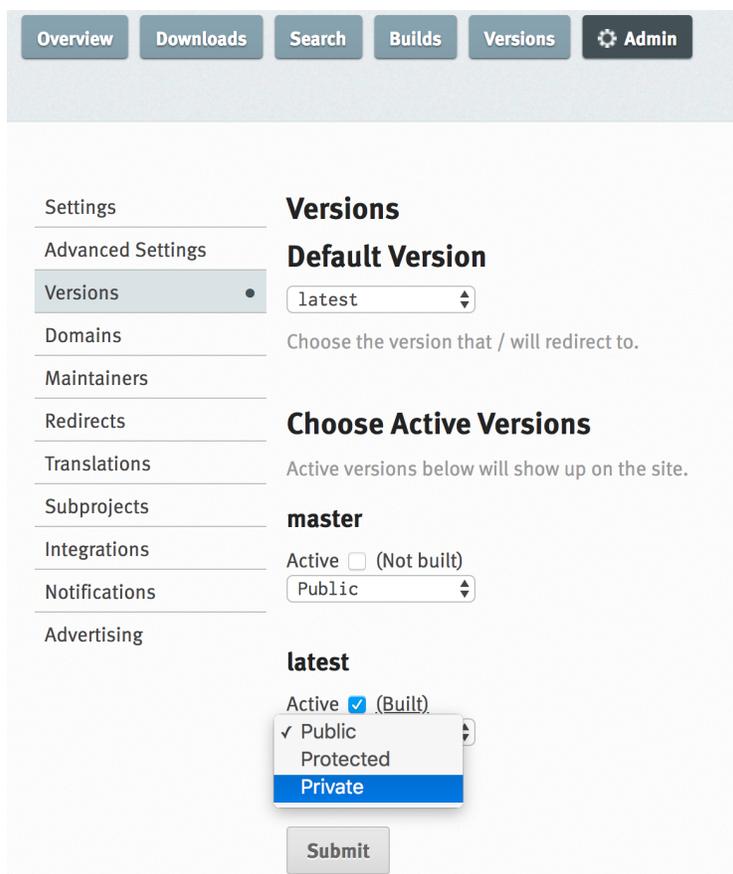


Fig. 2: Il backend di Docs Italia permette di impostare documenti privati.

5.5.3 Documenti in lingue multiple

Docs Italia permette di gestire i documenti e le loro traduzioni. L'utente può leggere il documento in un'altra lingua in qualsiasi pagina si trovi, semplicemente selezionando il pulsante corrispondente.

Per creare un documento multilingue è necessario creare i documenti nelle diverse lingue seguendo le procedure descritte in precedenza e, successivamente, impostare un documento come “principale” e indicare gli altri come traduzioni di questo.

Per esempio, un documento in italiano chiamato *anpr-modalitasubentro* potrebbe essere il documento principale. Per indicarne il documento *anpr-modalitasubentro-eng* come la traduzione inglese, segui la seguente procedura.

Procedura

Impostare la traduzione di un documento

1. Accedi al backend di Docs Italia corrispondente al documento principale, *anpr-modalitasubentro*.
2. Dal pannello di amministrazione, seleziona la voce **Translations**.
3. Nel riquadro **Project**, indica il nome del documento corrispondente alla traduzione. In questo caso, *anpr-modalitasubentro-eng*.
4. Clicca sul pulsante **Submit**.

La Figura 3 mostra il backend di Docs Italia relativo alla selezione dei documenti di traduzione.

Per problemi con la creazione di documenti multilingue è possibile contattare gli amministratori di Docs Italia tramite Slack di Developers Italia usando il canale #docs-italia.

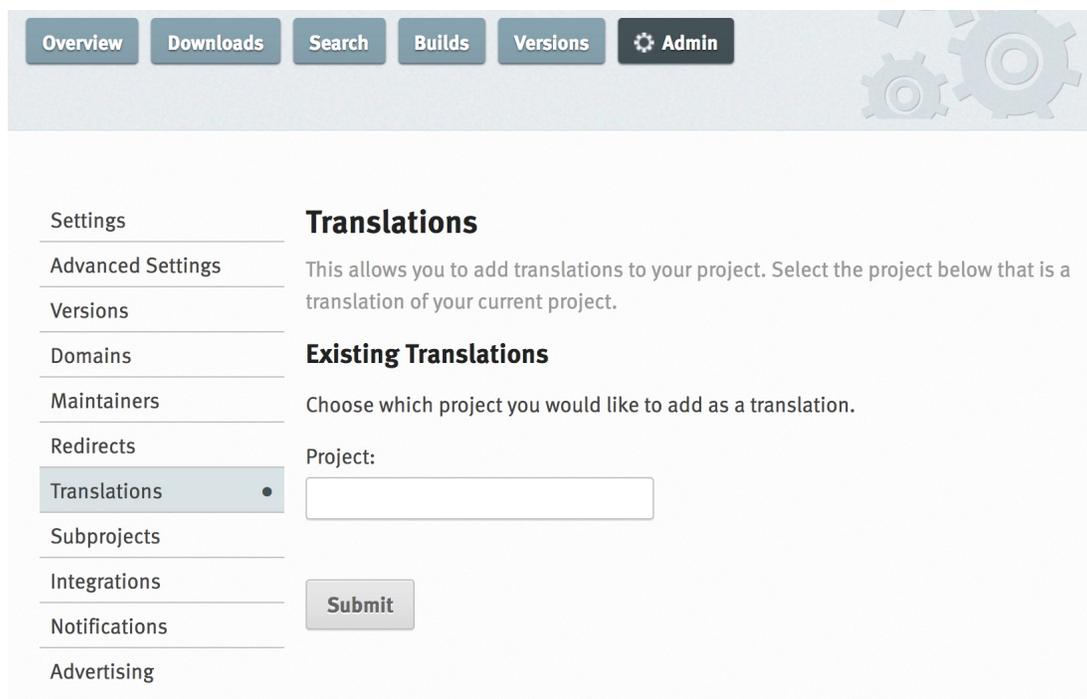


Fig. 3: Il backend di Docs Italia permette di impostare documenti multilingue.

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](#), 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 5](#) 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](#).

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

6.1.4 Promozione sui canali social

Un'ulteriore promozione del documento può avvenire attraverso i **canali social**. [Twitter](#) 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](#). 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* 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* sul versionamento).

6.2.2 Issues e pull requests

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

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.

Appendice 1. Il versionamento

7.1 Introduzione

Il **version control** 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*” (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](#) (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).

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* 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](#) 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)

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.

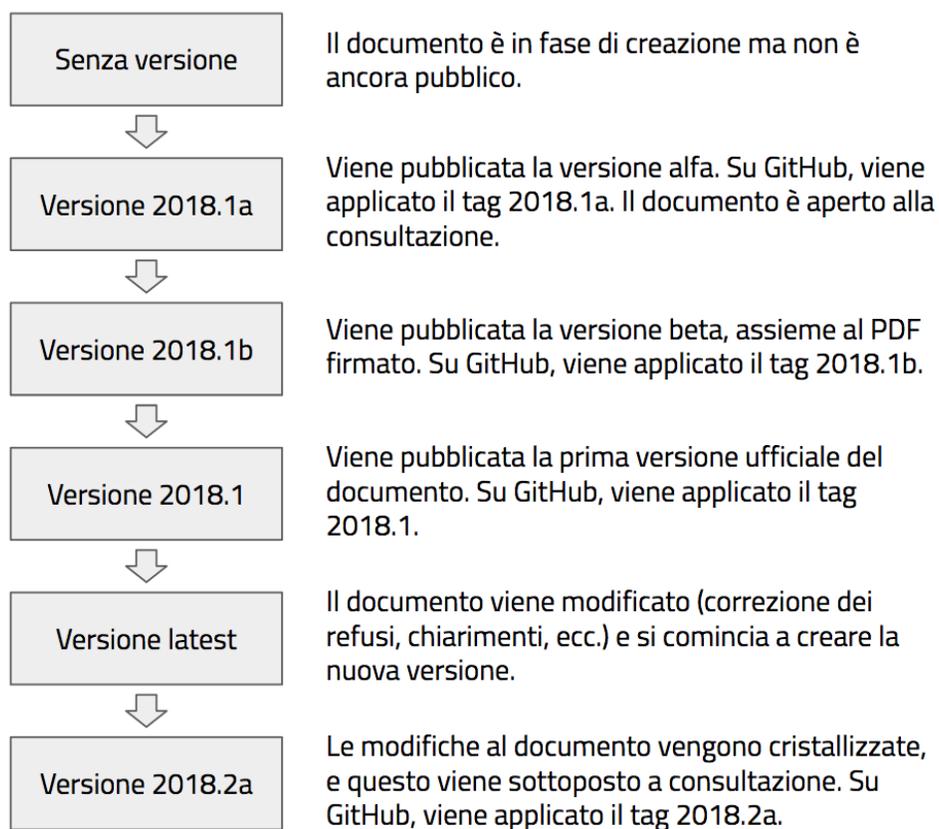


Fig. 1: Esempio di flusso di versionamento.

Appendice 2. Guida di stile

Questa appendice fornisce alcune indicazioni sull'organizzazione dei contenuti e lo stile dei documenti su Docs Italia e integra le indicazioni fornite nelle [Linee guida di design per i servizi web della PA](#).

8.1 Organizzazione dei contenuti

Il tuo testo verrà letto principalmente online.

Inserisci le informazioni più importanti all'inizio del testo, in modo da fornire subito a chi legge le informazioni che cerca. Utilizza i paragrafi successivi per ulteriori spiegazioni.

Usa il grassetto o il corsivo per mettere in evidenza i punti più importanti e attirare l'attenzione.

Usa una struttura modulare.

Una struttura ben definita aiuta a comprendere meglio il testo. Separa le varie sezioni in paragrafi con un titolo rilevante. A volte, un'immagine o una tabella possono fornire più informazioni di un testo.

Nell'organizzare i contenuti, utilizza la seguente struttura modulare:

- Acronimi
- Contenuti
- Domande frequenti (FAQ)

Gli acronimi, se presenti, dovrebbero essere inseriti in testa alla documentazione. Si sconsiglia, nei documenti web, l'utilizzo di lunghe introduzioni: i lettori sono interessati direttamente ai contenuti.

Prima di creare una sezione di FAQ, chiediti se serve davvero¹: stai semplicemente duplicando le informazioni? Puoi inserire gli stessi contenuti all'interno della documentazione? Nel caso in cui sia comunque necessario creare una sezione di FAQ, le domande devono essere efficaci, sintetiche e dirette verso il fruitore della documentazione.

Tipi di contenuti

¹ GOV.UK, nelle proprie linee guida, sconsiglia l'uso delle FAQ: <https://www.gov.uk/guidance/content-design/writing-for-gov-uk#dont-use-faqs>

Per aiutarti a strutturare meglio il tuo testo, puoi **pensare alla funzione di ciascuna parte** secondo questa tabella (tipica della [documentazione DITA](#)):

Concetti	Descrizioni e spiegazioni relative a un argomento. Formato: paragrafi di testo. Esempio: Introduzione a pagoPA
Procedure	Lista di operazioni o di passi per ottenere un risultato. Formato: elenco puntato o numerato. Esempio: 18app, Istruzioni per la compilazione di una fattura
Liste di consultazione	Copie di indice-valore (per esempio un glossario, lista di funzioni API, ecc). Formato: tabella a due colonne. Esempio: ANPR, Tabella 14. Lingue
Frammenti	Brevi frasi che fungono da connessione fra le varie parti della guida. Esempio: “Maggiori informazioni possono essere trovate su questo sito”.

8.2 Stile e linguaggio

Relativamente allo stile da utilizzare, è possibile far riferimento alle [Linee Guida di Design](#), applicando qualche ulteriore accorgimento dettagliato di seguito.

8.2.1 Caratteri speciali

Alcuni caratteri speciali possono essere interpretati in maniera errata in fase di compilazione del documento. Per esempio, gli apici obliqui sono utilizzati per definire alcune direttive nel linguaggio RST.

Per questo motivo, utilizza doppi apici dritti invece che doppi apici obliqui, sia in apertura che in chiusura. L'apice obliquo va sostituito con l'apice dritto.

Da evitare	Consigliato
Doppi apici obliqui: ` ``	Doppi apici dritti: ""
Apice obliquo singolo: ` `	Apice dritto singolo: ''

8.2.2 Codice

Il codice deve essere visualizzato con carattere *monospace*. Per poter visualizzare il codice con tale formattazione, indipendentemente dal template, è necessario usare il ruolo `:code:` inserendo il codice tra due apici obliqui.

example

Formattazione del codice

```
:code:`<aux digit (1n)>[<application code> (2n)]<codice IUV>`
```

Codice reso nel testo

```
<aux digit (1n)>[<application code> (2n)]<codice IUV>
```

8.2.3 Note

L'utilizzo delle note nei documenti web è sconsigliato: a volte, le note possono essere sostituite da *riferimenti* nel testo (vedi sotto). Se indispensabili, le note devono essere tutte in fondo alla pagina e non, ad esempio, dopo una tabella o alla fine di un paragrafo.

Le note hanno una numerazione indipendente in ciascuna pagina. Per un esempio, consulta il [Piano Triennale per l'Informatica nella PA 2017-2019](#).

8.2.4 Riferimenti

All'interno del documento, i riferimenti ad altre parti di testo devono essere corredati da opportuni link.

Quando crei un collegamento a un sito esterno, evita di associare il link a frasi come “clicca qui” oppure “a questo link”. Usa, invece, frasi che descrivono il contenuto del collegamento, come “visita la pagina di aiuto” oppure “consulta la sintassi RST”.

8.2.5 Tabelle

Nelle tabelle, il testo di ogni cella inizia sempre con la lettera maiuscola, salvo il caso in cui si faccia riferimento a identificatori che inizino con la lettera minuscola. A fine riga in ciascuna tabella non serve il punto. L'intestazione delle colonne e delle righe di una tabella è in grassetto.

La tabella deve essere preceduta da una didascalia centrata che ne descriva il contenuto.

Le tabelle devono essere numerate in maniera progressiva. Per la numerazione, si può utilizzare l'auto numerazione già presente nel template di Docs Italia. La numerazione manuale è fortemente sconsigliata in quanto introduce un elemento di possibile incoerenza nel documento che si sta scrivendo o che si modificherà in futuro.

Tabella 1: Esempio di tabella in Docs Italia.

	Intestazione col. 1	Intestazione col. 2	Intestazione col. 3
Intestazione riga 1	Testo	Testo	Testo
Intestazione riga 2	Testo	Testo	Testo

8.2.6 Figure e immagini

Tutte le figure e le immagini devono essere seguite da opportune didascalie, numerate in maniera progressiva. Per la numerazione delle figure si può utilizzare l'auto numerazione già presente nel template di Docs Italia. La numerazione manuale è fortemente sconsigliata in quanto introduce un elemento di possibile incoerenza nel documento che si sta scrivendo o che si modificherà in futuro.

Un esempio è mostrato in Figura %s.



TEAM PER LA
TRASFORMAZIONE
DIGITALE
Presidenza del Consiglio dei Ministri

Fig. 1: Esempio di figura in Docs Italia: il logo del Team per la trasformazione digitale.

8.2.7 Titoli

Come già indicato nelle [Linee Guida di Design](#), nei titoli non dovrebbero essere presenti trattini o slash (-, /, |). Nel caso sia necessario utilizzare un segno divisorio, questo dovrebbe essere il punto.

example

Formato del titolo

Appendice 1. Codici dei regimi fiscali

8.2.8 Ottimizzazione SEO

L'ottimizzazione per i motori di ricerca (in inglese, *search engine optimization* o SEO) è un aspetto importante da tenere presente nella scrittura di un documento. L'obiettivo dell'ottimizzazione SEO dei contenuti è di migliorare il posizionamento del documento fra i risultati delle ricerche degli utenti in corrispondenza di alcune parole chiave. I primi risultati, infatti, sono quelli che ricevono più click e quindi più visite.

Per maggiori informazioni sull'ottimizzazione SEO, puoi consultare il [capitolo dedicato nelle Linee guida di design per i servizi web della PA](#), dove vengono illustrate una serie di tecniche e strategie per migliorare i contenuti dei documenti.

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:

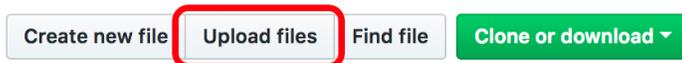
1. Upload tramite interfaccia grafica sul sito github.com;
2. Upload da repository Git locale tramite i comandi *clone* e *push*.

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.

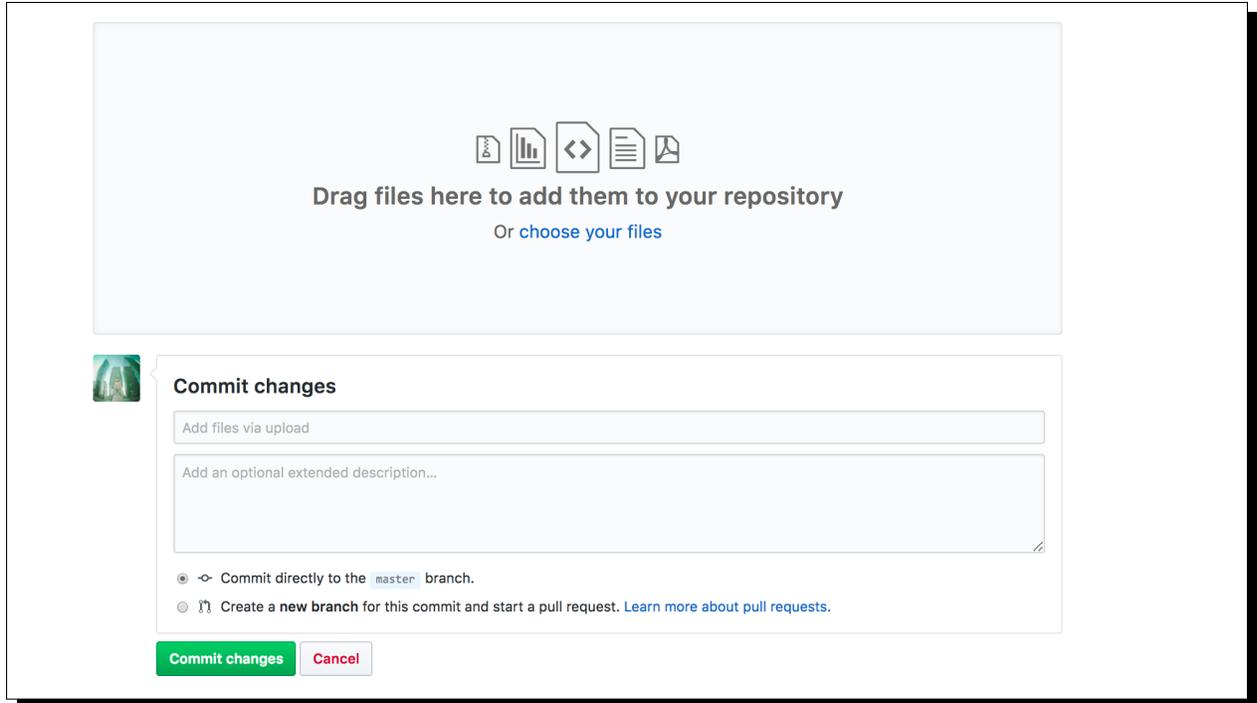
9.1.1 Upload tramite interfaccia grafica

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**



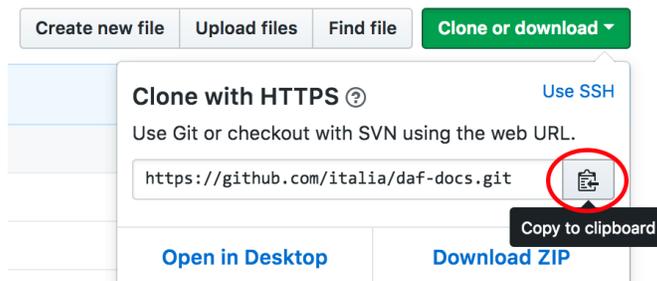
4. Clicca su **choose your files** e seleziona tutti i file che intendi caricare
5. Nel riquadro “Commit changes”, specifica un oggetto del commit nel primo box, e opzionalmente un testo di spiegazione, secondo le modalità descritte nella sezione *Messaggi di commit* |
6. Clicca sul pulsante **Commit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