

Pubblicare un documento su dmnotes da linea di comando (script di pubblicazione)

Tag: CLI · dmnotes · Guide · SQLite

Ultimo aggiornamento: 2026-09-19

Pubblicare un documento su dmnotes da linea di comando

A cosa serve

Script per inserire (o aggiornare) un documento già pubblicato in dmnotes scrivendo un file Markdown, senza passare dall'interfaccia admin. Utile per:

- pubblicare guide scritte in locale (Markdown) direttamente sul sito;
- procedimenti automatici (script di pubblicazione da script di deploy);
- ripubblicare un documento corretto senza copia/incolla nell'editor.

Lo script: `/tmp/insert_doc_args.py` (sorgente nel container agente: `/workspace/dmnotes/insert_doc_args.py`).

1. Prerequisiti

- 1 Lo script gira su nas64 (192.168.1.10), perché il database vive sul filesystem del NAS: `/export/docker_config/dmnotes/data/dmnotes.db`
- 2 Serve root (sudo) per scrivere il DB: il file appartiene al container dmnotes e non è scrivibile dagli utenti normali.
- 3 Il container dmnotes legge il DB direttamente: una scrittura breve in transazione è sicura anche a servizio attivo (il documento appare subito, senza riavvii).
- 4 Il documento sorgente è un file Markdown (.md) completo di titolo (prima riga non usata come titolo: il titolo va passato come argomento) ed eventuale sintassi estesa supportata dall'editor di dmnotes.

2. Sintassi

```
python3 /tmp/insert_doc_args.py <file.md> "<titolo>" <slug> "<tag1,tag2,...>" [categoria]
```

Argomento	Obbligatorio	Descrizione
<code><file.md></code>	sì	percorso del file Markdown da pubblicare
<code>"<titolo>"</code>	sì	titolo del documento (meglio quotarlo: contiene spazi e simboli)

Argomento	Obbligatorio	Descrizione
<slug>	sì	identificativo URL; univoco; solo a-z 0-9 e trattini
"<tag1,tag2>"	sì	elenco tag separati da virgola (creati se non esistono)
[categoria]	no	slug della categoria, default public

Esempio reale

```
python3 /tmp/insert_doc_args.py /tmp/guida.md \
  "Tradurre meeting live EN+IT con AI locale: la pipeline v2" \
  traduzione-live-meeting-ai-locale-pipeline-v2 \
  "AI,Traduzioni,Whisper,GPU,Guide" \
  public
```

Output: inserito id 56 (l'id numerico assegnato).

Il documento è subito visibile su:

<https://busce.it/dmnotes/doc/<slug>>

e compare in homepage /dmnotes/.

3. Cosa fa lo script, passo passo

- 1 Legge il file Markdown specificato (UTF-8).
- 2 Controlla se lo slug esiste già: se sì, esce senza toccare nulla (slug già presente: id N) — comportamento idempotente: rilanciare lo stesso comando non crea doppioni.
- 3 Calcola l'istante UTC corrente per created_at/updated_at/published_at.
- 4 Inserisce la riga in documents con: status='published', visibility='public', owner_id=NULL (documento pubblico, non intestato a nessun utente); import_info='{ }', version=' '.
- 5 Crea/collega i tag: per ogni tag genera lo slug (normalizzazione NFKD → solo a-z0-9, spazi/simboli → -), inserisce in tags se mancante e collega in document_tags.
- 6 Collega la categoria (document_categories): se la categoria non esiste, viene semplicemente saltata (il documento resta senza categoria).
- 7 Aggiorna la ricerca full-text: cancella eventuali righe FTS del documento e inserisce title + body + tags in documents_fts, così il documento è immediatamente ricercabile dalla ricerca interna.

4. Aggiornare un documento già pubblicato

Lo script non aggiorna: se lo slug esiste, esce senza modifiche (mostrando slug già presente: id N).

Per aggiornare i contenuti di un documento esistente si modifica direttamente nel DB (dall'interfaccia admin di dmnotes oppure via SQL):

```
sudo sqlite3 /export/docker_config/dmnotes/data/dmnotes.db \
  "UPDATE documents SET content_md=readfile('/tmp/nuovo.md'), updated_at=strftime('%Y-%m-%dT%H:%M:%SZ', 'now')
  WHERE slug='<slug>';"
```

Attenzione: dopo un UPDATE manuale di content_md l'indice di ricerca FTS va riallineato a mano:

```
sudo sqlite3 /export/docker_config/dmnotes/data/dmnotes.db \  
"DELETE FROM documents_fts WHERE rowid=(SELECT id FROM documents WHERE slug='<slug>');  
INSERT INTO documents_fts(rowid,title,body,tags)  
SELECT id,title,content_md,'' FROM documents WHERE slug='<slug>';"
```

5. Ripulire il sorgente prima di pubblicare (consigli)

- Scrivere in Markdown puro: titoli #, elenchi, tabelle pipe, blocchi ``` — sono supportati dall'editor di dmnotes.
- Il titolo è separato dal contenuto: non duplicare un # titolo iniziale nel corpo se coincide con il titolo del documento (o volutamente, per gusto di layout).
- Immagini: usare percorsi assoluti raggiungibili dal sito (o URL esterni); lo script non gestisce upload di file binari.

6. Dove trovare tutto

Cosa	Dove
Script di pubblicazione (nel container agente)	/workspace/dmnotes/insert_doc_args.py
Esempio di documento pubblicato	https://busce.it/dmnotes/doc/traduzione-live-meeting-ai-locale-pipeline-v2
Vecchio script monouso (hardcoded)	/workspace/dmnotes/insert_doc.py (mantenuto per compatibilità)
DB dmnotes (su nas64)	/export/docker_config/dmnotes/data/dmnotes.db

Guida redatta il 2026-09-18.