



BugBusters

Come Usare il POC Nexum

Guida all'installazione e utilizzo

Stato	In redazione
Responsabile	
Verificatore	
Redattori	Alberto Autiero
Destinatari	BugBusters, Eggon, Prof. Tullio Vardanega, Prof. Riccardo Cardin

Registro delle Modifiche

Versione	Data	Descrizione	Redatto	Verificato	Approvato
0.0.1	29/12/2025	Prima stesura della guida all'utilizzo del POC del documento	Alberto Autiero	-	-

Indice

1	Introduzione	4
2	Prerequisiti	4
3	Configurazione Iniziale	4
3.1	Configurare PostgreSQL	4
3.1.1	Modificare la password nel file di configurazione	4
3.1.2	Verificare che PostgreSQL sia in esecuzione	4
3.2	Configurare le Credenziali AWS	5
3.2.1	Creare il file .env	5
4	Avvio del POC	5
4.1	Script PowerShell avvia-poc.ps1	5
5	URL Disponibili	6
5.1	Applicazioni Principali	6
5.2	Pagine di Test HTML	6
6	Come Testare il POC	6
6.1	Test 1: Analisi Documenti	6
6.2	Test 2: Generazione Testo con Conversazioni	6
7	Script PowerShell Disponibili	7
8	Troubleshooting	7
8.1	Errore: "company_id mancante"	7
8.2	Errore: "Database non inizializzato"	7
8.3	Errore: "Password PostgreSQL errata"	7
8.4	Errore: "AWS credentials expired"	7
8.5	Errore: "Could not find gem"	8
8.6	Errore: "Port already in use"	8
8.7	Il server non si avvia	8
8.8	Warning VIPS	8
9	Note Importanti	8
10	Riavvio Dopo Modifiche	8
11	Risorse Aggiuntive	9

1 Introduzione

Questa guida spiega in dettaglio come configurare, avviare e utilizzare il Proof of Concept (POC) Nexum. Seguire attentamente le istruzioni per garantire il corretto funzionamento di tutti i componenti.

2 Prerequisiti

Prima di iniziare, assicurarsi di avere installato:

- **Ruby** (versione 3.0 o superiore)
- **PostgreSQL** (versione 12 o superiore)
- **Node.js** (versione 16 o superiore, per il frontend Angular)
- **Credenziali AWS** con accesso a Bedrock
- **PowerShell** (già incluso in Windows)

3 Configurazione Iniziale

3.1 Configurare PostgreSQL

Il POC richiede PostgreSQL in esecuzione con le credenziali corrette.

3.1.1 Modificare la password nel file di configurazione

Aprire il file `backend/config/database.yml` e modificare la password PostgreSQL in tutte le sezioni (development, test, production):

Listing 1: Configurazione database.yml

```
1 development:
2   <<: *default
3   database: nexum_poc_development
4   username: postgres
5   password: "TUA_PASSWORD_QUI" # <-- Modifica questa riga
6   host: 127.0.0.1
7   port: 5432
```

Nota: Sostituire "TUA_PASSWORD_QUI" con la password dell'utente PostgreSQL.

3.1.2 Verificare che PostgreSQL sia in esecuzione

Verificare che PostgreSQL sia attivo eseguendo:

Listing 2: Verifica PostgreSQL

```
1 Test-NetConnection -ComputerName localhost -Port 5432
```

Se la porta non risponde, avviare il servizio PostgreSQL da "Servizi" di Windows.

3.2 Configurare le Credenziali AWS

Il POC necessita delle credenziali AWS per accedere a Bedrock. Creare un file `.env` nella cartella `backend`.

3.2.1 Creare il file `.env`

Creare un file chiamato `.env` nella cartella `backend/` con il seguente contenuto:

Listing 3: File `.env` di esempio

```
1 AWS_ACCESS_KEY_ID=la_tua_access_key
2 AWS_SECRET_ACCESS_KEY=la_tua_secret_key
3 AWS_SESSION_TOKEN=la_tua_session_token
4 AWS_REGION=us-east-1
```

Dove trovare le credenziali AWS:

1. Accedere al portale AWS IAM Identity Center (solitamente <https://eggon.awsapps.com/start/>)
2. Selezionare il ruolo "Bedrock-Bugbuster"
3. Copiare le credenziali temporanee (Access Key ID, Secret Access Key, Session Token)
4. Incollare i valori nel file `.env`

Importante:

- Il file `.env` è già ignorato da git (non verrà committato)
- Le credenziali AWS scadono dopo alcune ore
- Quando le credenziali scadono, crearne di nuove dal portale AWS

4 Avvio del POC

4.1 Script PowerShell `avvia-poc.ps1`

Il modo più semplice per avviare tutto:

Listing 4: Script combinato

```
1 .\avvia-poc.ps1
```

Questo script:

- Apre una nuova finestra PowerShell per il Backend
- Apre una nuova finestra PowerShell per il Frontend

- Mostra un messaggio con tutti gli URL disponibili

Per fermare i servizi: Premere CTRL+C

5 URL Disponibili

Una volta avviati i servizi, avrai accesso a:

5.1 Applicazioni Principali

- **Frontend Angular:** <http://localhost:4200>
- **Backend Rails API:** <http://localhost:3000>

5.2 Pagine di Test HTML

- **Generazione Testo:** <http://localhost:3000/tester.html>
- **Analisi Documenti:** <http://localhost:3000/documentTester.html>

6 Come Testare il POC

6.1 Test 1: Analisi Documenti

1. Aprire <http://localhost:3000/documentTester.html>
2. Assicurarsi che "Company ID" sia impostato a 1
3. Cliccare su "Scegli file" e selezionare `documento-test.pdf` presente nella cartella backend
4. Cliccare su "Carica e Analizza"
5. Attendere che l'AI analizzi il documento
6. Visualizzare i risultati estratti

Creare un PDF di test: Se non hai il file `documento-test.pdf`, puoi crearne uno eseguendo:

```
1 ruby crea-pdf-test.rb
```

6.2 Test 2: Generazione Testo con Conversazioni

1. Aprire <http://localhost:3000/tester.html>
2. Impostare "Company ID" a 1

3. Cliccare su "Carica toni" per vedere i toni disponibili
4. Selezionare un tono dal menu a tendina
5. Scrivere un prompt (es: "Scrivi un'email di presentazione")
6. Cliccare su "Genera"
7. Visualizzare la risposta dell'AI

7 Script PowerShell Disponibili

Il progetto include diversi script PowerShell utili:

Script	Descrizione
<code>avvia-poc.ps1</code>	Avvia sia Backend che Frontend (apre due finestre)
<code>avvia-backend.ps1</code>	Avvia solo il Backend Rails
<code>avvia-frontend.ps1</code>	Avvia solo il Frontend Angular
<code>verifica-postgres.ps1</code>	Verifica la connessione a PostgreSQL (chiede password)
<code>crea-pdf-test.rb</code>	Crea un PDF di test per l'API di analisi

Tabella 1: Script disponibili nel progetto

8 Troubleshooting

8.1 Errore: "company_id mancante"

- Assicurati di includere `company_id=1` nella query string o nel form
- Le pagine HTML di test ora includono un campo per il Company ID

8.2 Errore: "Database non inizializzato"

- Esegui manualmente: `cd backend && bundle exec rails db:create db:migrate db:seed`

8.3 Errore: "Password PostgreSQL errata"

- Verifica che la password in `backend/config/database.yml` corrisponda alla password del tuo utente PostgreSQL
- Usa lo script `verifica-postgres.ps1` per testare la connessione

8.4 Errore: "AWS credentials expired"

- Le credenziali AWS scadono dopo alcune ore

- Ottieni nuove credenziali dal portale AWS IAM Identity Center
- Aggiorna il file `backend/.env` con le nuove credenziali

8.5 Errore: "Could not find gem"

- Esegui: `cd backend && bundle install`

8.6 Errore: "Port already in use"

- La porta 3000 (backend) o 4200 (frontend) è già in uso
- Chiudi l'applicazione che usa quella porta o modifica la porta nel file di configurazione

8.7 Il server non si avvia

- Verifica che PostgreSQL sia in esecuzione
- Controlla che le variabili AWS siano configurate nel file `.env`
- Controlla i log in `backend/log/development.log`

8.8 Warning VIPS

- I warning VIPS (riguardanti moduli opzionali per immagini) possono essere ignorati - non bloccano il funzionamento dell'applicazione.

9 Note Importanti

- **Database:** Il seed iniziale crea una Company con ID=1, alcuni Tone di esempio e una conversazione di esempio
- **Company ID:** La maggior parte delle API richiede `company_id=1` (la company di default)
- **Storage:** I file caricati vengono salvati localmente in `backend/storage/`
- **Log:** I log del backend sono disponibili in `backend/log/development.log`
- **Hot Reload:**
 - Rails: Si ricarica automaticamente quando modifichi i file Ruby
 - Angular: Si ricompila automaticamente quando modifichi i file TypeScript/HTML

10 Riavvio Dopo Modifiche

Dopo aver modificato:

- **File Ruby (modelli, controller, ecc.):** Rails si ricarica automaticamente
- **File di configurazione Rails:** Riavviare il POC (Ctrl+C e rieseguire `.\avvia-poc.ps1`)
- **File Angular:** Angular si ricompila automaticamente
- **File .env:** Riavviare il POC (Ctrl+C e rieseguire `.\avvia-poc.ps1`)

11 Risorse Aggiuntive

- **Documentazione Backend:** Vedere `backend/docs/SETUP_E_DOCUMENTAZIONE.txt`
- **Gemfile:** Lista delle dipendenze Ruby in `backend/Gemfile`
- **package.json:** Lista delle dipendenze Node.js in `frontend/package.json`