

connetti_database.php

Intestazione:

Connessione al database MySQL tramite PDO.

Questo file si occupa di creare una connessione al database MySQL utilizzando PDO, con configurazione adatta a un ambiente Docker.

La connessione viene stabilita al database "progetto_IGS" sul servizio "db" (nome host Docker), con utente "root" e password "root".

In caso di errore nella connessione, viene generata un'eccezione PDOException.

Variabile globale esportata:

- \$pdo: istanza PDO per l'accesso al database.

Funzioni:

Nessuna funzione

creaTassonomia.php

Intestazione:

Riceve tramite POST i dati:

- 'nome': nome della tassonomia
- 'desc': descrizione della tassonomia

Effettua l'inserimento nella tabella `tassonomia` e restituisce:

- l'ID della tassonomia appena inserita in caso di successo,
- '-1' in caso di errore.

```
@param string $_POST['nome'] Nome della tassonomia.  
@param string $_POST['desc'] Descrizione della tassonomia.  
  
@return string ID inserito oppure '-1' in caso di errore.
```

Funzioni:

Nessuna funzione

elimina_tassonomia.php

Intestazione:

Riceve tramite POST:

- 'idTassonomia' (int): ID della tassonomia da controllare ed eventualmente eliminare.

Funzionalità:

- Controlla se la tassonomia ha elementi.
- Se ha elementi, restituisce l'id della radice e se è padre (ha figli).
- Se non ha elementi, elimina la tassonomia.

Restituisce un JSON con un campo 'esito' che può assumere valori:

- 1 : eliminazione avvenuta con successo
- 0 : tassonomia popolata, con dati su radice e figli
- -1 : errore nel controllo elementi
- -2 : errore nel recupero della radice
- -3 : errore nel controllo dei figli
- -4 : errore nell'eliminazione della tassonomia

```
@param int $_POST['idTassonomia'] ID della tassonomia da gestire.  
@return void Stampa un JSON con l'esito dell'operazione e informazioni  
aggiuntive.
```

Funzioni:

getRadiceTassonomia:

```
@param mixed $pdo  
@param mixed $idTassonomia  
@return int ritorno l'id della radice o "-1" in caso di problemi
```

checkFigli:

```
@param \PDO $pdo connessione al DB  
@param int $idTassonomia id della tassonomia di cui ottenere la radice  
@return int ritorno l'id della radice o "-1" in caso di problemi
```

elimina_tassonomia:

```
@param \PDO $pdo connessione al DB
@param int $idTassonomia id della tassonomia da elimina
@return int ritorno 1 se va tutto bene o -4 se da problemi
```

elimina_voceCategoria.php

Intestazione

Riceve tramite POST:

- 'idVoce' (int): ID della voce da cancellare.

Funzionalità:

- Controlla se la voce è in uso (presente nella tabella `valore` con `tipo = 2`).
- Se la voce è in uso, impedisce la cancellazione e restituisce un codice di errore.
- Se la voce non è in uso, procede con l'eliminazione.

Restituisce un JSON con un campo 'esito' che può assumere valori:

- 1 : eliminazione avvenuta con successo
- -1 : idVoce non impostato correttamente
- -2 : voce in uso, impossibile eliminare
- -3 : errore nell'esecuzione della query di eliminazione

```
@param int $_POST['idVoce'] ID della voce da cancellare.
@return void Stampa un JSON con l'esito dell'operazione.
```

Funzioni

controllaUso:

```
@param \PDO connessione al DB
@param int $idVoce id della voce da controllare.
@return bool restituisce se la voce è in uso o no
```

eliminaVoce:

```
@param \PDO $pdo connessione al DB
@param int $idVoce id della voce de eliminare
```

```
@return bool ritorno il risultato della query (true se è tutto ok, false in caso contrario)
```

get_categoria.php

Intestazione

Script per ottenere informazioni o elementi di una categoria.

Parametri GET:

- idCategoria (int): ID della categoria da interrogare.
- scelta (string): indica cosa restituire:
 - '1' -> informazioni generali della categoria (id tassonomia, nome)
 - '2' -> elenco delle voci associate alla categoria

Risposte JSON:

- Per '1' o '2': array con i dati richiesti o array con chiave "risultato" e codice errore.
- Codici di errore possibili:
 - -1: parametro scelta mancante
 - -2: valore scelta non valido
 - -3: idCategoria non valido (non numerico)
 - -4: errore esecuzione query getInfoCategoria
 - -5: errore esecuzione query getElementoCategoria

Funzioni:

getInfoCategoria

```
@param \PDO $pdo connessione al DB
@param int $idCategoria id della categoria di cui voglio le informazioni
@return array array da encodare in json. Contiene i risultati.
```

getElementoCategoria

```
@param \PDO $pdo connessione al DB
@param int $idCategoria id della categoria di cui voglio ottenere gli elementi
@return array array con le info della tassonomia/il codice di errore
```

get_elemento.php

Intestazione:

Script per il recupero delle informazioni relative a un elemento.

Questo script permette di:

- Ottenere l'intestazione dell'elemento (scelta = 0)
- Ottenere i valori dell'elemento (scelta = 1)
- Ottenere i nomi precedenti dell'elemento (scelta = 2)
- Ottenere i sinonimi dell'elemento (scelta = 3)
- Ottenere i valori precedenti dell'elemento (scelta = 4)

I parametri sono ricevuti tramite GET:

- id (int): ID dell'elemento
- scelta (int): tipo di richiesta (0–4)

Restituisce i dati in formato JSON (o CSV per l'intestazione).

Funzioni:

getValori

```
@param int $id id dell'elemento. Verrà usato per cercare gli attributi  
dell'elemento nel DB  
@param \PDO $pdo Connessione al database
```

getNomiPrecedenti

```
@param int $id id dell'elemento. Verrà usato per cercare i nomi precedenti  
dell'elemento nel DB  
@param \PDO $pdo Connessione al database
```

getSinonimi

```
@param int $id id dell'elemento da cui ottenere i sinonimi  
@param \PDO $pdo connessione al DB  
@return void
```

getValoriPrecedenti:

```
@param int $id id dell'elemento da cui prendere i valori precedenti
@param \PDO $pdo connessione al DB
@return void
```

getIntestazione:

```
@param int $id id dell'elemento da cui leggere le informazioni
@param \Pdo $pdo connessione al DB
@return void
```

get_index.php

Intestazione:

Script per il recupero delle informazioni base delle tassonomie.

Recupera l'elenco di tutte le tassonomie presenti nel database, restituendo per ciascuna il nome e l'ID.

Il risultato è restituito in formato JSON e utilizzato da index.php.

Funzioni:

Nessuna funzione.

get_nomi_precedenti.php

Intestazione:

Recupera i nomi precedenti degli elementi associati a una tassonomia specifica.

Endpoint: questo script riceve un parametro GET `idTassonomia`, esegue dei controlli di validità e restituisce in formato JSON l'elenco dei nomi precedenti legati a quella tassonomia.

Parametri:

- `idTassonomia (int)`: ID numerico della tassonomia da interrogare.

Codici di esito:

- `-1`: parametro mancante

- -2: parametro non numerico
- -3: errore nell'esecuzione della query
- 1: successo, con array `nomiPrecedenti` contenente i risultati

Output JSON:

```
{
  "esito": int,
  "nomiPrecedenti": [
    {
      "nome_elemento": string,
      "nome_precedente": string,
      "data_modifica": string (formato dd-mm-yyyy)
    },
    ...
  ]
}
```

Funzioni:

Nessuna funzione

get_tab_categorie.php

Intestazione:

Restituisce le categorie associate a una tassonomia specifica.

Endpoint: accetta un parametro GET `idTassonomia`, verifica che sia un intero positivo, e restituisce in formato JSON tutte le categorie appartenenti a quella tassonomia.

Parametri:

- `idTassonomia` (int): ID numerico della tassonomia di riferimento.

Codici di esito:

- `status = -1`: parametro mancante o non valido
- `status assente` (default): successo, restituisce array di categorie

Output JSON:

```
[
  {
    "nome": string,
```

```
        "id": int
    },
    ...
]
```

Funzioni:

Nessuna funzione

get_tassonomia.php

Intestazione:

Restituisce le categorie associate a una tassonomia specificata tramite parametro GET.

Se il parametro 'idTassonomia' è valido, esegue una query sul database per ottenere l'elenco delle categorie appartenenti a quella tassonomia e restituisce i risultati in formato JSON. In caso contrario, restituisce un errore JSON con status -1.

```
@param int $_GET['idTassonomia'] ID della tassonomia per cui recuperare le
categorie
@return string JSON con la lista delle categorie o un messaggio di errore
```

Funzioni:

getInfoTassonomia:

```
@param \PDO $pdo Connessione al database
@param int $id_tassonomia id della tassonomia di cui voglio ottenere le
informazioni di base
```

getElementiTassonomia:

```
@param \PDO $pdo Connessione al database
@param int $id id della tassonomia di cui voglio ottenere la lista degli
elementi
```

getNumeroElementi:

```
@param \PDO $pdo connessione al DB
@param int $id id della tassonomia su cui contare il numero di elementi
```



```
@return void
```

rimuovi_categoria.php

Intestazione:

Gestisce la richiesta di eliminazione di una categoria.

Riceve un ID via POST, controlla se le voci associate sono in uso, ed eventualmente elimina la categoria dal database.

Ritorna un JSON con l'esito dell'operazione:

- -1: ID non valido
- -2: Errore durante il recupero degli ID delle voci
- -3: Errore nella query di controllo
- -4: La categoria non può essere eliminata (è in uso)
- -5: Errore durante l'eliminazione
- altrimenti, nessun output in caso di eliminazione riuscita

Richiede una connessione PDO (\$pdo) e un parametro POST 'idCategoria'.

Funzioni

getIDsVoci

```
@param \PDO $pdo connessione al DB
@param int $idCategoria id della categoria da cui ottengo gli ID delle voci
@return array | int ritorno o l'array con gli IDs o un numero di errore
```

controllaCategoria:

```
@param \PDO $pdo connessione al DB
@param array $arrayIdsVoci array contenente gli IDs da controllare.
@return bool | int Se anche solo uno è utilizzato, ritorno true. Se ci sono errori, ritorno un codice di errore
```

eliminaCategoria:

```
@param \PDO $pdo connessione al DB
@param int $idCategoria della categoria da rimuovere
```

```
@return bool ritorno il risultato dell'eliminazione
```

rimuovi_elemento.php

Intestazione:

Nessuna intestazione.

Funzioni:

getListaFigli

Ottiene la lista di tutti i discendenti (figli, nipoti, ecc.) di un elemento includendo anche se stesso, tramite una query ricorsiva.

```
@param PDO $pdo Istanza della connessione al database  
@param int $idElemento ID dell'elemento radice  
@return array|false Array di ID dei discendenti o false in caso di errore
```

filtraSinonimi

Filtra dalla lista di ID gli elementi che sono sinonimi, mantenendo solo quelli principali.

```
@param PDO $pdo Istanza della connessione al database  
@param array $listaIDs Lista di ID da filtrare  
@return array|false Lista di ID principali (non sinonimi), oppure false in caso di errore
```

getValoriAssociati

Ottiene l'elenco degli ID dei valori associati agli elementi forniti.

```
@param PDO $pdo Istanza della connessione al database  
@param array | int $listaIDsPrincipali Lista di ID degli elementi principali  
@return array|false Lista degli ID dei valori associati o false in caso di errore
```

copia_E_rimuoviElementi

```
@param \PDO $pdo connessione al DB  
@param array | int $listaIDs lista degli id da copiare e rimuovere (int in caso di un singolo elemento)  
@return bool ritorno il risultato dell'eliminazione
```

rimuoviSinonimi

```
@param \PDO $pdo connessione al DB
@param array $arrayIDsSinonimi array contenete gli Ids dei sinonimi
@return bool restituisce il risultato dell'eliminazione
```

rimuoviValori

```
@param \PDO $pdo connessione al database
@param array $listaValori lista id valori
@return bool ritorno il risultato della cancellazione
```

getSinonimi

```
@param \PDO $pdo connessione al DB
@param int $idElemento id dell'elemento di cui cercare i sinonimi
@return array ritorno la lista degli IDs dei sinonimi
```

eliminaNodo

```
@param \PDO $pdo connessione al DB
@param int $idElemento id dell'elemento da eliminare
@return bool ritorno il risultato dell'eliminazione
```

eliminaFoglia

```
@param \PDO $pdo connessione al DB
@param int $idElemento id dell'elemento da eliminare
@return bool ritorno il risultato dell'eliminazione
```

salvaModifiche.php

Intestazione:

Aggiorna il nome e/o la descrizione di una tassonomia nel database.

Riceve tramite POST i parametri:

- 'nome' (string|null): nuovo nome della tassonomia (opzionale).
- 'descrizione' (string|null): nuova descrizione della tassonomia (opzionale).

- 'idTassonomia' (int): ID della tassonomia da aggiornare (obbligatorio).

Risponde con un JSON contenente:

- 'risultato' (int): codice di stato dell'operazione.
 - 1: aggiornamento riuscito.
 - -1: ID tassonomia mancante.
 - -2: nessun dato da aggiornare fornito.
 - -3: errore durante l'esecuzione della query.
- 'errore' (string, opzionale): messaggio d'errore se presente.

Funzioni:

Nessuna funzione

set_categoria.php

Intestazione:

Nessuna intestazione

Funzioni:

inserisciCategoria

```
@param \PDO $pdo connessione al database
@param string $nome nome della categoria da creare
@param int $idTassonomia id della tassonomia a cui associarla
@return int ritorno l'id della nuova tassonomia o -1 in caso di errori
```

inserisciVoci:

```
@param \PDO $pdo connessione al db
@param array $listaVoci lista delle voci da inserire
@param int $idCategoria della categoria a cui associare le voci
@return bool ritorno il risultato dell'inserimento
```

controllaDati:

```
@param \PDO $pdo connessione al DB
@return bool restituisce il risultato del controllo sui dati (se sono inseriti
```

set_elemento.php

Introduzione:

Questo script riceve dati tramite POST per eseguire diverse operazioni sulla tassonomia:

- inserire un nuovo elemento (nome)
- aggiungere attributi a un elemento esistente
- aggiungere sinonimi a un elemento esistente

Parametri POST attesi a seconda dell'operazione scelta:

- scelta (int): 1 per inserire nome, 2 per aggiungere attributi, 3 per aggiungere sinonimi

Per inserire un nome:

- idTassonomia (int)
- idPadre (int|null)
- nomeElemento (string)

Per aggiungere attributi:

- idElemento (int)
- listaAttributi (JSON array di oggetti con proprietà "valore" e "tipo")

Per aggiungere sinonimi:

- idElemento (int)
- idTassonomia (int)
- idPadre (int|null)
- listaSinonimi (JSON array di stringhe)

Restituisce:

- id dell'elemento inserito (per scelta=1)
- 1 in caso di successo (per scelta=2 e 3)
- -1 in caso di errore

Funzioni:

aggiungiNome

```
@param \PDO $pdo connessione al DB
@return int ritorna -1 in caso di errore, negli altri casi l'id dell'elemento
inserito
```

aggiungiAttributi

```
@param \PDO $pdo connessione al DB
@return void
```

aggiungiSinonimi

```
@param \PDO $pdo connessione al DB
@return void
```

sostituzioneElemento.php

Introduzione

Gestione sostituzione nome elemento con storico e pulizia dati associati

Questo script permette di aggiornare il nome di un elemento esistente nella tassonomia, salvandone il nome precedente nella tabella 'nomi_precedenti', eliminando i sinonimi collegati e rimuovendo i valori (attributi) associati all'elemento.

Riceve tramite POST i seguenti parametri:

- scelta (int): indica l'operazione da eseguire (1 = sostituzione nome)
- id_elemento (int): id dell'elemento da modificare
- nuovoNome (string): nuovo nome da assegnare all'elemento

Restituisce un JSON con:

- risultato (int):
 - 1 = successo
 - 1 = id_elemento non valido o mancante
 - 2 = scelta mancante o non gestita
 - 3 = nuovoNome mancante o vuoto
 - 4 = errore in transazione PDO, con messaggio in 'errore'

Funzioni

copia_e_sostiusciNome

```
@param \PDO $pdo connessione al DB  
@param int $idElemento id dell'elemento da sostituire  
@return array ritorno un array con risultato ed eventuali errori
```