

classAttributoElemento.js

Descrizione

Rappresenta un attributo con un valore e un tipo associato.

Metodi

```
constructor
Crea un nuovo Attributo.
@param {} valore - Il valore dell'attributo.
@param {string} tipo - Il tipo dell'attributo.

get valore
Ottiene il valore dell'attributo.
@returns {} Il valore corrente.

get tipo
Ottiene il tipo dell'attributo.
@returns {string} Il tipo corrente.

set valore
Imposta un nuovo valore per l'attributo.
@param {} nuovoValore - Il nuovo valore da assegnare.

set tipo
Imposta un nuovo tipo per l'attributo.
@param {string} nuovoTipo - Il nuovo tipo da assegnare.

toJSON
Ritorna una rappresentazione JSON dell'attributo.
@returns {{valore: , tipo: string}} Oggetto JSON con valore e tipo.
```

classElementoTassonomia.js

Descrizione

Rappresenta un elemento con nome, id padre, attributi, sinonimi e nomi precedenti.

Metodi

constructor

Crea un nuovo elemento.

@param {number|string} id - Identificativo univoco dell'**elemento** (non incluso nel **JSON**).

@param {string} nome - Nome dell'elemento.

@param {number|string|null} idPadre - Identificativo del genitore dell'elemento.

@param {Attributo[]} [listaAttributi=[]] - Lista degli attributi associati all'elemento.

@param {string[]} [listaSinonimi=[]] - Lista di sinonimi per l'elemento.

@param {Elemento[]} [listaNomiPrecedenti=[]] - Lista di nomi precedenti dell'**elemento** (come elementi).

get nome

@returns {string} Nome corrente dell'elemento

get idPadre

@returns {number|string|null} Id del genitore

get listaAttributi

@returns {Attributo[]} Lista degli attributi

get listaSinonimi

@returns {string[]} Lista dei sinonimi

get listaNomiPrecedenti

@returns {Elemento[]} Lista dei nomi precedenti come elementi

get id

@returns {number|string} Id interno dell'elemento

set nome

@param {string} nuovoNome - Nuovo nome da assegnare

@param {number|string|null} nuovoIdPadre - Nuovo id padre

set listaAttributi

@param {Attributo[]} nuovaListaAttributi - Nuova lista di attributi

set listaSinonimi

@param {string[]} nuovaListaSinonimi - Nuova lista di sinonimi

set listaNomiPrecedenti

@param {Elemento[]} nuovaListaNomiPrecedenti - Nuova lista di nomi precedenti

set id

@param {number|string} id - Imposta un nuovo id interno

toJSON

Converte l'**elemento** in un oggetto **JSON** escludendo l'id interno.

Esegue la serializzazione di attributi e nomi precedenti.
@returns {Object} Rappresentazione JSON dell'elemento.

classNomePrecedente.js

Descrizione

Rappresenta un nome precedente associato a una data di modifica.

```
constructor
Crea un nuovo NomePrecedente.
@param {string} nome - Il nome precedente.
@param {string|Date} dataModifica - La data in cui il nome è stato modificato.

get nome
@returns {string} Il nome precedente

get dataModifica
@returns {string|Date} La data di modifica del nome

set nome
Imposta un nuovo nome precedente.
@param {string} nuovoNome - Il nuovo nome precedente.

set dataModifica
Imposta una nuova data di modifica.
@param {string|Date} nuovaData - La nuova data di modifica.

toJSON
Restituisce la rappresentazione JSON dell'istanza.
@returns {{nome: string, dataModifica: string|Date}} Oggetto JSON con nome e data di modifica.
```

classTassonomia.js

Descrizione

Rappresenta una tassonomia composta da un insieme di elementi.

```
constructor
Crea una nuova Tassonomia.
@param {string} nome - Il nome della tassonomia.
```

```
@param {string} descrizione - La descrizione della tassonomia.  
@param {Elemento[]} [listaElementi=[]] - Lista degli elementi associati alla  
tassonomia.  
  
get nome  
@returns {string} Il nome della tassonomia  
  
get descrizione  
@returns {string} La descrizione della tassonomia  
  
get listaElementi  
@returns {Elemento[]} La lista degli elementi associati  
  
set nome  
Imposta un nuovo nome per la tassonomia.  
@param {string} nuovoNome - Il nuovo nome.  
  
set descrizione  
Imposta una nuova descrizione.  
@param {string} nuovaDescrizione - La nuova descrizione.  
  
set listaElementi  
Imposta una nuova lista di elementi.  
@param {Elemento[]} nuovaListaElementi - La nuova lista di elementi.  
  
toJSON  
Restituisce la rappresentazione JSON dell'istanza.  
@returns {{nomeTassonomia: string, descrizione: string, elementi: Object[]}}  
Oggetto JSON della tassonomia.
```

creaCategoria.js

Metodi

```
getId()  
Legge e ritorna l'id passato in GET e letto dal querystring.  
@returns {number} id passato in GET e letto dal querystring  
  
stampaErrore  
Stampa un messaggio di errore e ritorna all'index.  
@returns {void}  
  
aggiungiVoce  
Aggiunge input voci quando viene aumentato il contatore  
@param {number} numeroScelto numero di voci scelto dall'utente
```

```
@param {number} numeroAttuale numero di voci attuale
@returns {void}

rimuoviVoce
Rimuove gli input in eccesso quando diminuisce il numero
@param {number} numeroAttuale numero attuale di elementi
@param {number} numeroScelto numero scelto di elementi
@returns {void}
```

creaElemento.js

Metodi

```
creaFormElemento
crea il form in base al tipo (attributo sinonimo)
@param {number} num numero di input da mettere
@param {jQuery} $container container che contiene l'input e la scelta della categoria
@param {string} tipo stringa che specifica se attributo o sinonimo
@returns {any}

creaOpzioniSelect
crea le option per scegliere la categoria
@param {Array} data array con le categorie che l'utente può scegliere
@param {string} strOpzioni stringa che contiene il codice HTML per generare il select
@returns {string} ritorno la stringa completa con i vari option

creaOpzioniCategoria
crea le option per scegliere l'elemento di una categoria
@param {any} data
@param {any} strOpzioni
@returns {any}

creaCategoriale
crea l'elenco di select per scegliere il categoriale
@param {number} idElemento id elemento categoriale
@param {jQuery} oldInput oggetto jquery con il blocco <input> presente al momento del cambio
@returns {any}

getInfoCategoria
ottiene e poi stampa a schermo le informazioni delle categorie
@param {number} idCategoria id della categoria da cui prendere le informazioni
@param {any} idElemento id dell'elemento da usare come value
```

```
@returns {any}

getListaAttributi
legge tutti gli attributi per ottenere la lista
@returns {Array} ottiene la lista degli attributi inseriti dall'utente

getListaSinonimi
Legge la lista di sinonimi inseriti dall'utente
@returns {Array} lista con tutti i sinonimi inseriti

aggiungiAttributi
aggiunge al DB gli attributi agli elementi
@param {Array} listaAttributi lista degli attributi
@param {number} idElemento id dell'elemento a cui associare gli attributi
@returns {void}

aggiungiSinonimi
aggiunge al DB i sinonimi
@param {Array} listaSinonimi lista dei sinonimi da aggiungere
@param {number} idTassonomia id della tassonomia in cui aggiungere i sinonimi
@param {number} idElemento id dell'elemento a cui associare i sininomi
@returns {void}

salvaElemento
entro in questa funzione se è stato cliccato "salva elemento"
@returns {void}

stampaErrore
Stampa un messaggio di errore e rimanda all'index
@returns {void}
```

creaTassonomia.js

Metodi

```
salvaTassonomia
Invia la richiesta al server per salvare la tassonomia
@returns {void}
```

downloadJson.js

Metodi

getTassonomia

Funzione asincrona che, dato un **ID** di tassonomia, esegue una chiamata **AJAX GET** per recuperare i dati della **tassonomia** (nome, descrizione, ecc.) dal backend.

Restituisce una **Promise** che risolve con i dati **in** formato **JSON**.

@param {number} idTassonomia id della tassonomia da ottenere

@returns {Object} risultati della chiamata **in** **JSON**.

getElementi

Funzione asincrona che recupera la lista degli elementi associati alla tassonomia tramite chiamata **AJAX GET** al backend, usando l'id tassonomia e un flag funzione diverso.

Restituisce una **Promise** con la lista degli elementi **in** **JSON**.

@param {number} idTassonomia id della tassonomia di cui voglio gli elementi

@returns {Object} risultati della chiamata **in** **JSON**.

getAttributiElemento

Funzione asincrona che, dato l'id di un elemento, richiede al backend gli attributi di quell'elemento.

Utilizza un endpoint **diverso** ("get_elemento.php") con parametro scelta=1 per richiedere gli attributi.

Restituisce una **Promise** che risolve con gli attributi **in** formato **JSON**.

@param {number} idTassonomia

@returns {void}

getSinonimiElementi

Funzione che richiede i sinonimi degli elementi

@param {number} idElemento id dell'elemento principale così da avere sinonimi

@returns {Object} risultati della chiamata **in** **JSON**.

getNomiPrecedenti

Funzione che richiede i nomi precedenti

@param {number} idElemento

@returns {Object} risultati della chiamata **in** **JSON**.

creaJSON

Funzione principale asincrona che coordina la costruzione del modello dati **JSON** partendo dall'id della tassonomia.

La funzione crea un oggetto tassonomia e una lista di elementi con i loro attributi associati.

@param {number} idTassonomia

@returns {void}

eliminaTassonomia.js

Metodi

```
rimossaConSuccesso
```

Se la tassonomia viene rimossa con successo, lo stampo poi rimuovo la riga

```
@param {any} idRiga
```

```
@returns {any}
```

```
eliminaTassonomia
```

Funziona che inizia la rimozione della tassonomia

```
@param {number} idTassonomia
```

```
@returns {void}
```

```
pulisciTassonomia
```

Funzione che richiede la rimozione degli elementi della tassonomia

```
@param {number} idRadice id della radice della tassonomia
```

```
@param {boolean} isPadre true = elemento è padre, false in caso contrario
```

```
@param {number} idTassonomia id della tassonomia da eliminare
```

```
@returns {void}
```

```
eliminaTassonomiaDefinitiva
```

Rimuovo effettivamente la tassonomia. Qui effettuo la richiesta di cancellazione

```
@param {number} idTassonomia
```

```
@returns {void}
```

index.js

Metodi

```
modificaInfoTassonomia
```

```
@param {number} idTassonomia id della tassonomia di cui voglio modificare le informazioni
```

```
@returns {void}
```

modificaInfoTassonomia.js

Metodi

```
backHome
```

Fa tornare l'utente alla homepage

```
@returns {void}
```

```
impostaValori
```

Imposto i valori negli input


```
@param {string} nomePrecedente nome attuale (letto dal DB)
@param {string} descrizionePrecedente descrizione attuale (letto dal DB)
@returns {void}

risultatiModifica
Verifico i risultati
@param {Object | Object[]} data Risultato della chiamata
@returns {void}

salvaModifiche
Legge i nuovi valori e dopo averli verificati fa una richiesta per l'inserimento
@returns {void}

impostaTitle
Imposto il titolo alla pagina HTML
@param {string} titolo
@returns {void}
```

nomiPrecedenti.js

Metodi

```
getId
Legge dal query string l'id passato e lo ritorna
@returns {number} ritorno l'id

creaRiga
Creo la riga da mettere nella tabella per visualizzare i nomi precedenti
@param {any} nomePrecedente
@returns {string} riga da mettere poi nell'html

popolaTabella
Si occupa di popolare la tabella
@param {String[]} nomiPrecedenti array con i nomi precedenti
@returns {void}

stampaErrore
Stampa l'errore e ritorna all'index
@returns {void}
```

sostituisciElemento.js

Metodi

```
stampaErrore
Stampa l'errore e ritorna all'index
@returns {void}

creaFormElemento
Crea il form per inserire l'elemento
@param {number} num numero di elementi
@param {jQuery} $container oggetto JQ che rappresenta il container per gli input
@param {string} tipo due possibili valori: attributo o sinonimo
@returns {void}

creaOpzioniSelect
Crea le option per scegliere la categoria
@param {Object[]} data array con gli elementi da scegliere
@param {string} strOpzioni stringa a cui aggiungere le opzioni
@returns {string} stringa con tutte le opzioni

creaOpzioniCategoria
Crea le option per scegliere l'elemento di una categoria
@param {Object[]} data array con gli elementi categoriali
@param {string} strOpzioni stringa con le varie opzioni da mettere nel select
@returns {any}

creaCategoriale
Crea l'elenco di select per scegliere il categoriale
@param {number} idElemento
@param {any} oldInput
@returns {void}

getInfoCategoria
Ottiene e poi stampa a schermo le informazioni delle categorie
@param {number} idCategoria id della categoria
@param {any} idElemento id dell'elemento categoriale
@returns {void}

getListaSinonimi
Ottiene la lista dei sinonimi
@returns {string[]} lista sinonimi

impostaSinonimi
Effettua la richiesta per inserire i sinonimi nel DB
@returns {void}

impostaAttributi
Effettua la richiesta per inserire gli attributi nel DB
@returns {void}
```

salvaElemento

Entro **in** questa funzione se è stato cliccato "**salva elemento**"

Effettua la richiesta per iniziare il processo di salvataggio di un elemento

@returns {**void**}

tabellaCategoria.js

Metodi

getId

Legge e ritorna l'id passato **in GET** e letto dal querystring.

@returns {number} id passato **in GET** e letto dal querystring

eliminaCategoria

Richiedo l'eliminazione della categoria

@param {number} idCategoria id della categoria da eliminare

@returns {**void**}

creaTabella

Creo e stampo la tabella con le categorie

@param {Object[]} categorie array la lista delle categorie

@returns {**void**}

stampaErrore

Stampa l'**errore e ritorna all'**index

@returns {**void**}

visualizzaCategoria.js

Metodi

getId

legge e ritorna l'id passato **in get** e letto dal querystring

@returns {number} restituisce l'id

popolaTabella

Riempie la tabella

@param {Array} categoria contiene le informazioni della categoria passati dal **DB**

@returns {**void**}

eliminaElemento

Procedura di eliminazione di una voce della categoria

@param {number} idVoce id della voce da cancellare

```

@returns {void}

creaTabella
Inizia la procedura di creazione della tabella
@returns {void}

impostaInformazioni
Imposta il titolo e i bottoni della pagina
@param {Object} info informazioni da mettere nei vari bottoni
@returns {void}

stampaErrore
Stampa un messaggio di errore e torna all'index
@param {string} msg
@returns {void}

```

visualizzaElemento.js

Metodi

```

popolaTabellaValori
Riempie la tabella con i valori
@typedef {string} Catoriale - Valore categoriale
@typedef {Object} Attributo
@property {int | string | Catoriale} Attributo
@param {Attributo[]} data - L'Array contiene tutti gli attributi dell'elemento di cui si sta leggendo la scheda

popolaTabellaNomiPrecedenti
Riempie la tabella con i nomi precedenti
@param {Array} data contiene le informazioni dei nomi precedenti
@returns {void}

popolaTabellaSinonimi
Riempio la tabella dei sinonimi
@param {Array} data contiene la lista dei sinonimi
@returns {void}

crea le tabelle per le informazioni dell'elemento
@param {Array} data - array che può contenere valori degli attributi, nomi precedenti, sinonimi, attributi precedenti
@param {int} sceltaTabella - valore che permette di capire quale funzione chiamare, così da popolare ogni volta la tabella corretta

creaIntestazione
Crea l'intestazione (nome, bottone indietro, ecc...)
@param {string} data contiene nome e id per impostare dinamicamente back e titolo
@returns {void}

```

```
getValori  
Raccoglie i valori dal DB per poi creare la pagina  
@param {int} id - id dell'elemento. Serve per la query
```

visualizzaTassonomia.js

Descrizione

Passaggi:

recupero l'id dal query string

mando la richiesta per recuperare nome e descrizione

mando la richiesta per recuperare i vari elementi

ricostruisco la tassonomia in base agli id padre

uso un file file .php e passerò un secondo parametro (1, 2, 3) in base a cosa devo fare

- 1 per recuperare nome e descrizione
- 2 per gli elementi
- 3 per controllare che non sia vuota

Metodi

```
sostituisciElemento  
Passo le informazioni per sostituire l'elemento  
@param {number} idElemento id dell'elemento da sostituire  
@param {number} idTassonomia id della tassonomia a cui appartiene  
@param {number} idPadre id del padre dell'elemento da sostituire  
@returns {void}  
  
creaAlbero  
Funzione che crea l'albero coi dati della tassonomia  
@param {int} padreId id del padre (serve per i figli)  
@param {Array} tassonomia tassonomia che andrà a riempire l'albero  
@returns {jQuery} ritorna un pezzo di albero  
  
popolaAlbero  
Altro pezzo dalla costruzione dell'albero  
@param {Array} tassonomia tassonomia che andrà a riempire l'albero  
@returns {void}  
  
getInfoTassonomia  
Funzione che prende le info della tassonomia  
@param {number} id id della tassonomia di cui prendere le info  
@returns {void}
```

checkEmpty

Controlla se la tassonomia è vuota

@param {number} id **ID** della tassonomia

@param {function} callback Funzione chiamata al termine della richiesta, riceve **in** input il numero di elementi trovati

@returns {void}

generaCreaRadice

Genera bottone per creare la **radice** (solo se tassonomia vuota)

@param {number} id id della tassonomia da generare

@returns {void}

aggiungiFiglio

Va sulla pagina di aggiunta del figlio passandogli i dati

@param {number} id_padre id del padre dell'elemento da aggiungere

@returns {void}

rimuoviElemento

Funzione per eliminare un elemento

@param {number} id_elemento id dell'elemento da aggiungere

@param {boolean} isPadre indica se l'elemento è padre

@param {number} idTassonomia id della tassonomia **in** cui è contenuto l'elemento

@returns {void}
