**JSON COMPLETAMENTE VALORIZZATO DI ESEMPIO PER LA RICERCA DI TIPO\_PRATICA**

**{**

// dati di input

**"searchFilters": {**

**"visualizza" : true,**

**"fullTextNomeSearch" : "anf ditt att"**

**},**

**"order": [ {"costo" : "asc"},**

**{"nome" : "desc"}],**

**"currentPage": 0,** //amche cosi va bene per nn fare la paginazione

**"elementForPage": 0,** //ma si possono benissimo non mettere proprio

// dati di ritorno

**"totalPages": 1,** //una pagina è il minimo che ti torna. Anche se vuota

**"totalElements": 3,** //il nome del campo parla da solo

**"list": [ { "id": 81,**

**"codice": "cod\_081",**

**"nome": "RISCATTO LAUREA ( INPS )",**

**"descrizione": null,**

**"costo": 0,**

**"visualizza": true**

**},**

**{**

**"id": 82,**

**"codice": "cod\_082",**

**"nome": "SEGNALAZIONE CONTRIBUTIVA",**

**"descrizione": null,**

**"costo": 0,**

**"visualizza": true**

**},**

**{**

**"id": 83,**

**"codice": "cod\_083",**

**"nome": "VARIAZIONE UFFICIO PAGATORE",**

**"descrizione": null,**

**"costo": 0,**

**"visualizza": true**

**}**

**]**

**}**

**SPIEGAZIONE**

**{**

**// parametri input**

**"searchFilters": {},**

**"order": [{}],**

**"currentPage": 0,**

**"elementForPage": 0,**

**// parametri di ritorno**

**"list": [{}]**

**"totalElements": 0,**

**"totalPages": 0,**

**}**

Per le ricerche questo è il *prototipo* di json che deve essere inserito nel BODY delle POST

Questo oggetto sarà lo stesso in invio che in ricezione, chiaramente con attributi valorizzati di conseguenza.

Quando si effettua la richiesta si devono popolare solo i parametri di input.

Di ritorno si avrà lo stesso oggetto inviato, ma con aggiunti i risultati trovati messi nei parametri di ritorno (output).

Chiaramente in fase di richiesta non si devono valorizzare i parametri di ritorno, anzi si possono pure non mettere proprio.

Il BODY può adirittura essere vuoto. la search capirà che si vuole tutta la tabella e di ritorno popola il bodi normalmente

**Spiegazione parametri:**

**"searchFilters": {},**

- La mappa dei filtri. Si inseriscono tutti gli attributie e i valori che si vuole usare per filtrare la ricerca:

"searchFilters": {

"nome\_campo\_stringa" : "valore\_stringa",

"nome\_campo\_numero" : valore\_numero,

"nome\_campo\_booleano" : valore\_booleano (true/false)

},

tutti questi campi sono messi in AND tra loro e il nome deve corrispondere a quello stabilito dal be (di solito quello della colonna in camelCase "nome\_completo"->"nomeCompleto").

Se il confronto è per = o like o <=, etc etc lo si deve stabilire prima.

Di base imposto tutte le ricerca degli attributi descrittivi con like, il resto uguali.

Caso particolare le ricerche "fulltext" che sono particolari, vedi esempio sotto.

**FULLTEXT**: ad esempio nel caso **tipo\_pratica** abbiamo una ricerca fulltext sul nome.

Si usa un campo fittizio, ad esempio ho scelto "**fullTextNomeSearch**" per inserire la stringa particolare da usare per la ricerca:

"searchFilters": {

**"fullTextNomeSearch" : "anf ditt att",**

},

Cerca tutte le pratiche che hanno nel nome tutte e 3 i pezzi singolarmente.

Questa condizione va in AND con le altre se messe. Per esempio si può aggiungere.

"searchFilters": {

**"fullTextNomeSearch" : "anf ditt att",**

**"visualizza" : true,**

},

e filtra in aggiunta tutte le visibili oltre a quelle che corrispondono alla fulltext

**"order": [{}],**

Una lista di oggetti a 2 parametri e serve per avere il risultato ordinato in base ad uno o più campi, esempio:

"order": [ {"costo" : "asc"},

{"nome" : "desc"}],

in questo caso ordina prima per costo ascendente, e tra quelli a costo uguale ordina per nome discendente.

**"currentPage": 0,**

**"elementForPage": 0,**

nel caso idati siano tanti e si voglia fare una ricerca paginata si possono indicare quale pagina vedere e il nr di elementi da mostrare per ogni pagina.

se non vengono valorizzati la paginazione non si fa, restituisce tutto. Ma se anche uno solo viene valorizzato parte la paginazione e imposta l’altro con il valore di default.

**DATI TI RISPOSTA AUTOMATICAMENTE VALORIZZATI NEL BODY DI RITORNO**

**"totalElements": 0,**

**"totalPages": 0,**

come da nome. nel caso non si richiede la pagina "totalPages" ritorna 1.

**"list": [{}]**

*la lista degli oggetti json trovati.* **La struttura dipende da cosa si cerca**.

Nel nostro esempio di **tipo\_pratica** se ci ritornano 3 elementi avremo una cosa del genere

"list": [ { "id": 81,

"codice": "cod\_081",

"nome": "RISCATTO LAUREA ( INPS )",

"descrizione": null,

"costo": 0,

"visualizza": true

},

{

"id": 82,

"codice": "cod\_082",

"nome": "SEGNALAZIONE CONTRIBUTIVA",

"descrizione": null,

"costo": 0,

"visualizza": true

},

{

"id": 83,

"codice": "cod\_083",

"nome": "VARIAZIONE UFFICIO PAGATORE",

"descrizione": null,

"costo": 0,

"visualizza": true

}

]