

FieldAliasesAndGlobsInParams

This was prompted by some ideas put forth in [SOLR-247](#) and in the mailing list threads linked to from that issue. See also: [SOLR-456](#)

For now this is a brainstorming page, if/when any of this gets implemented it can be reworked into a documentation page for users.

Background

Currently the `fl` param supports two "special" field names: `***` which means "any stored field", and `score` which not only means "include the score in the response", but also informs the request handler that scores should be computed. the `fl` param is split on the regex Pattern `",| "`.

The splitting happens in `SolrPluginUtils.setReturnFields`, which parses (one and only one) string `fl` param, and sets the field list on the `SolrQueryResponse`, as well as returning info about whether or not the list contained `score` so the handler has that info to work with.

Small problems with this (that most people have never cared about)...

- it makes it hard to use field names with spaces (or `|` or `,`) ...no other code in Solr cares what chars are in field names.
- you can't have a field named `score`

Some people expressed a desire to have `***` work for the `facet.field` param as well ... see [SOLR-247](#) for reasons why this is probably a bad idea, but having more generic glob syntax support (in both the `fl` and `facet.field` params) would be handy.

related issues

- `fl` can't be used as a multi-value param
- no way to prevent certain users from getting certain fields
- no way to prevent faceting on certain fields
- in most cases, searching and sorting on the same logical field requires clients to know two different field names (ie: `q=name:foo&sort=name_sortable+asc`)
- configuring language specific overrides for some field names (ie: `q=title:foo+summary:bar&lang=fr` should cause the `title_fr` and `summary_fr` fields to be used)

Broad Idea

Add robust support for letting solr admins configure what special syntax or aliases can be used *at query type* to refer to fields based on context (sorting, returned fields, search fields, facet fields, etc...)

new syntax in `solrconfig.xml` – most of which should be on a per handler basis (probably via a new Component) – that let's Solr administrators say things like:

- "this is the regex pattern to be used when processing fieldname related params"
 - `fl` becomes a multivalued field
 - default: `",| "`
 - if not specified, then fieldname params like `fl` and `facet.field` are taken literally (what do do about "sort" ?)
- "for this param, alias this string to this real field"
 - ie: `sort=name+asc` ultimately sorts on `name_sortable`
- "for this param, alias this string to the documents score"
 - defaults to `score` for `fl` and `sort`
 - ie: `fl=name,importance` ... importance might be the score fields
- "for this param, take any string that looks like a fieldname and append/prepend this string to it"
 - ie: any field name specified in the sort param can have `_sort` appended to it.
- "for this param, take any string that looks like a fieldname and append/prepend this other param to it"
 - ie: any field name specified in the sort param can have `_sort` appended to it
 - ie: any field name specified in the q param can have the value of the lang param appended via some syntax like `_$lang`
- "for this param, alias this string to a regex or glob"
 - ie: `fl=stockFields&fl=priceFields&facet.field=catFields` might mean return all fields matching two configured regexes and facet on all fields related to categorization.
- "allow users to specify globs for this param" or "allow users to specify regexes for this param"
 - ie: if globbing is turned on for `facet.field`, then `facet.field=facet_*` is legal
 - ie: if regexes are turned on for `fl` then, then `fl=name&fl=.*text` is legal
- "fields (not)-matching this glob or regex pattern are to be treated as if they didn't exist when using dealing with this param"
 - allows fields to be hidden in various contexts, even if the user guesses/knows they exist
 - ie: only allow the `sort` param to contain something matching the glob `**_sort`
 - ie: only return fields matching the regex `name|.price.*([short|long]summary)` ... even if the users uses a glob `fl` param (return the intersection of fields matching the regex and the glob)
- "for this param, ignore field names that aren't recognized or allowed by the configured rules"
- "for this param, error if a field name isn't recognized or allowed by the configured rules"

...all of these things should be combinable in an order specified by the solr admin, they can say things like "when dealing with the `facet.field` param, let users specify regexes to identify the fields to facet on, and map the string `price` to the field `price_dollars_facet` but ultimately ignore any field that doesn't match the glob `**_facet`"

Implementation

The best way to do this may be to have a Component which can be configured with all of these rules (and reused by multiple handlers). The component would parse the input params, error if necessary, and construct an object put into the request context that subsequent Components can call methods on to get field name Sets (or iterators) based on the param name being processed, the schema, the rules defined, the context of operation (ie: dealing with stored fields, dealing with indexed indexed fields, a specific document for returned fields, etc...)

It should not be too difficult if one uses the "new" queryParser mechanism from Lucene contrib. The Processor/Builder chain is suited for these changes. All this aliasing can be configured by configfile or on a per request basis.