# Building search app for public mailing lists in 15 minutes with ElasticSearch

by Lukáš Vlček, #BBUZZ - 2011

# *Agenda*

1. *Who & Why*
2. *Searching mailing lists*
3. *How to do it*
4. *Implementation details*
5. *Challenges*
6. *Give aways (free stuff inside!)*

# *Who am I?*


*Lukáš Vlček*

*Senior Software Engineer*
*JBoss Community team*
*Red Hat, Czech Republic*

# *Why I am here?*

*My mission is to **improve search***
*for JBoss.org content.*

# *Why I am here?*

*My mission is to **improve search** for JBoss.org content.*

# *Why I am here?*

*My mission is to **improve search** for JBoss.org content (and make it really rock!).*

# *And we are starting with...*

***mailing lists***

*of JBoss community projects (†)*

*(†) https://lists.jboss.org/mailman/listinfo/*

# *It is show time!*

*Searching mailing lists*

# *Starring:*

## Front end side:

jQuery

Sammy.js & {{ mustache }}

Highcharts

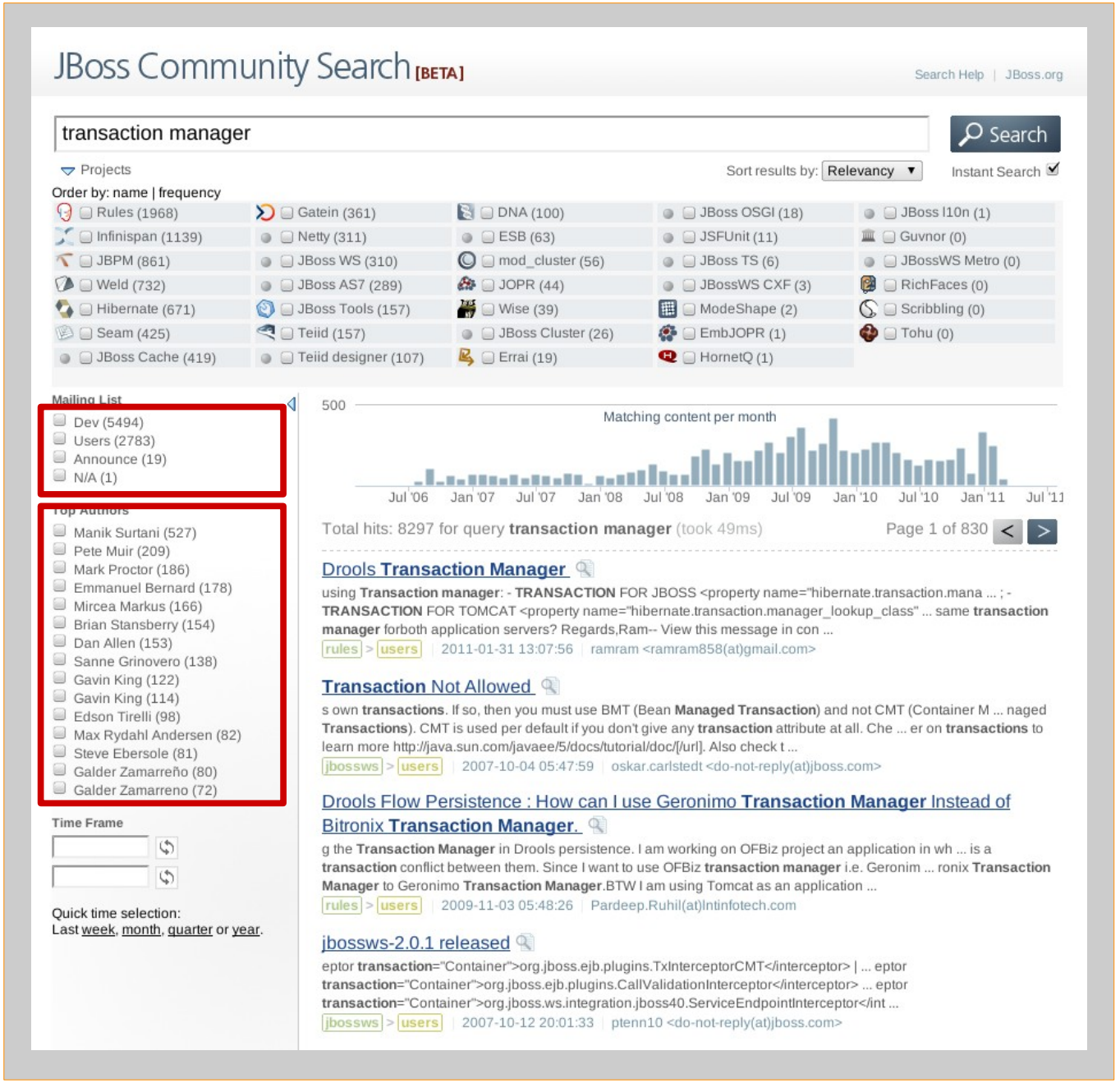
Isotope

## Back end side:

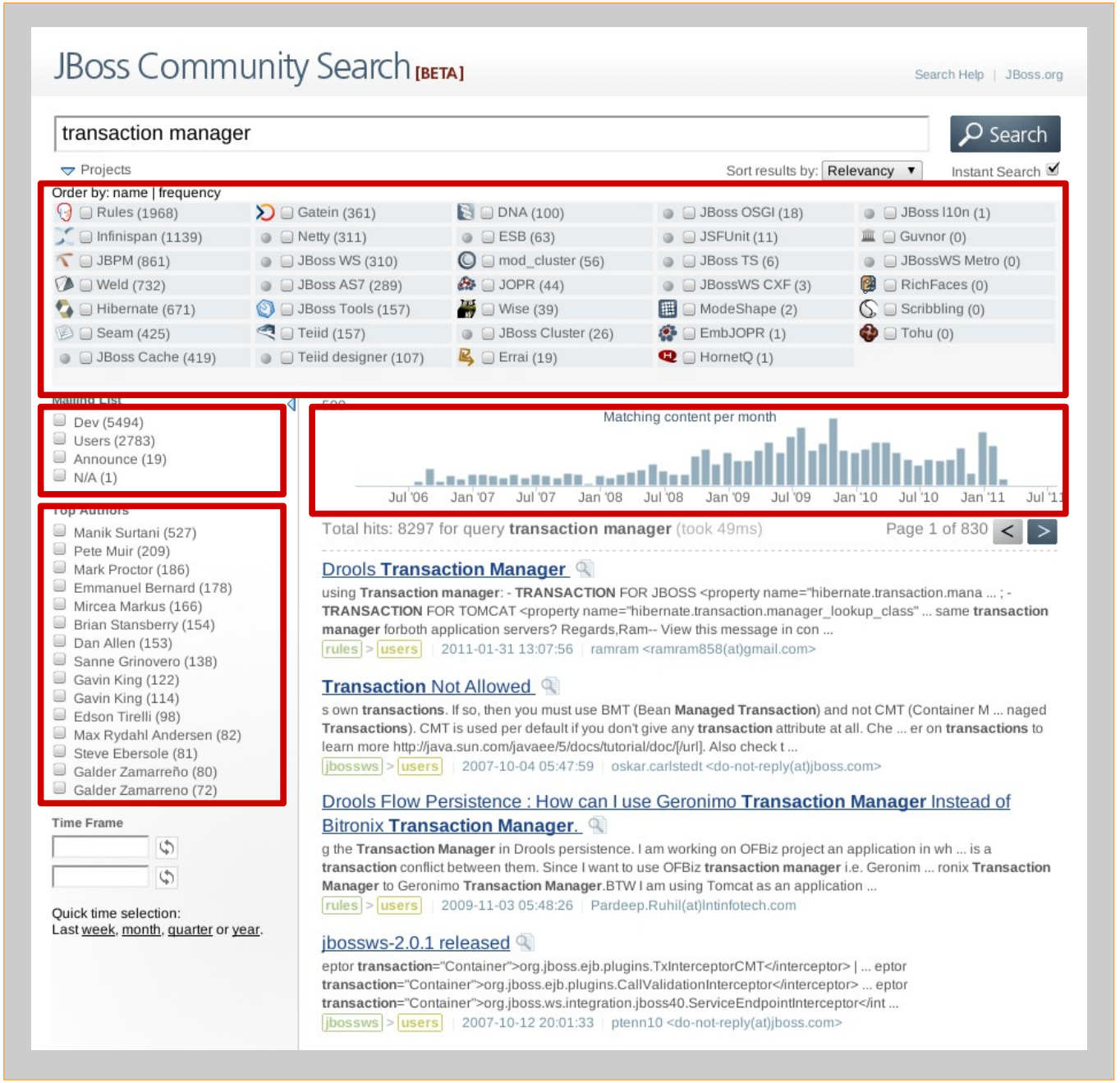
Elastic Search


*Note: the web UI can be different once we go into production.*

*Facets* ⇒




*Note: the web UI can be different once we go into production.*




Facets

*Note: the web UI can be different once we go into production.*

# *Search query in detail*

```
{
  from : 0,
  query : { ... },
  fields : [ ... ],
  highlight : { ... },
  facets : { ... }
}
```

*Full example: https://gist.github.com/1012051*

# *query : {...}*

query_string : { query : "jboss server" }

*Full example: https://gist.github.com/1012051*

# *query : {...}*

```
filtered : {
    query : { query_string : { query : "jboss server" } },
    filter: {
      and : [
        { range : { date : { from :"2007-07-25", to :"2010-12-16"}}},
        { terms : { _index : ["weld"]}},
        { terms : { mail_list : ["dev"]}},
        { terms : { from.not_analyzed : [
            "Galder Zamarreno <galder.zamarreno@redhat.com>",
            "Pete Muir <pmuir@redhat.com>"]}}
      ]}
    }
```

*Full example: https://gist.github.com/1012051*

# *fields : [...]*

"date", "document_url", "from", "mail_list",
"message_id", "message_snippet", "subject",
"subject_original", "to", "in_reply_to", "references"

*Full example: https://gist.github.com/1012051*

# *highlight : {...}*

```
pre_tags : ["<span class='hlt'>"],
post_tags : ["</span>"],
fields : {
  first_text_message : { number_of_fragments : 3 },
  first_html_message :{ number_of_fragments : 3 },
  message_attachments :{ number_of_fragments : 3 },
  subject : { number_of_fragments : 0 }
}
```

*Full example: https://gist.github.com/1012051*

# *facets : {...}*

histogram : {...},
projects : {...},
listType : {...},
author : {...},
author_filter : {...}

*Full example: https://gist.github.com/1012051*

# *facets.histogram : {...}*

date_histogram : { field : "date", interval : "month" }

*Full example: https://gist.github.com/1012051*

# *facets.author : {...}*

```
terms : { field : "from.not_analyzed", size : 15 },
facet_filter : {
  and : [
    { query :{ query_string : { query : "jboss server" }}},
    { range :{ date :{ from :"2007-07-25", to :"2010-12-16"}}},
    { terms :{ _index :["weld"]}},
    { terms :{ mail_list :["dev"]}}
  ]
},
global : true
```

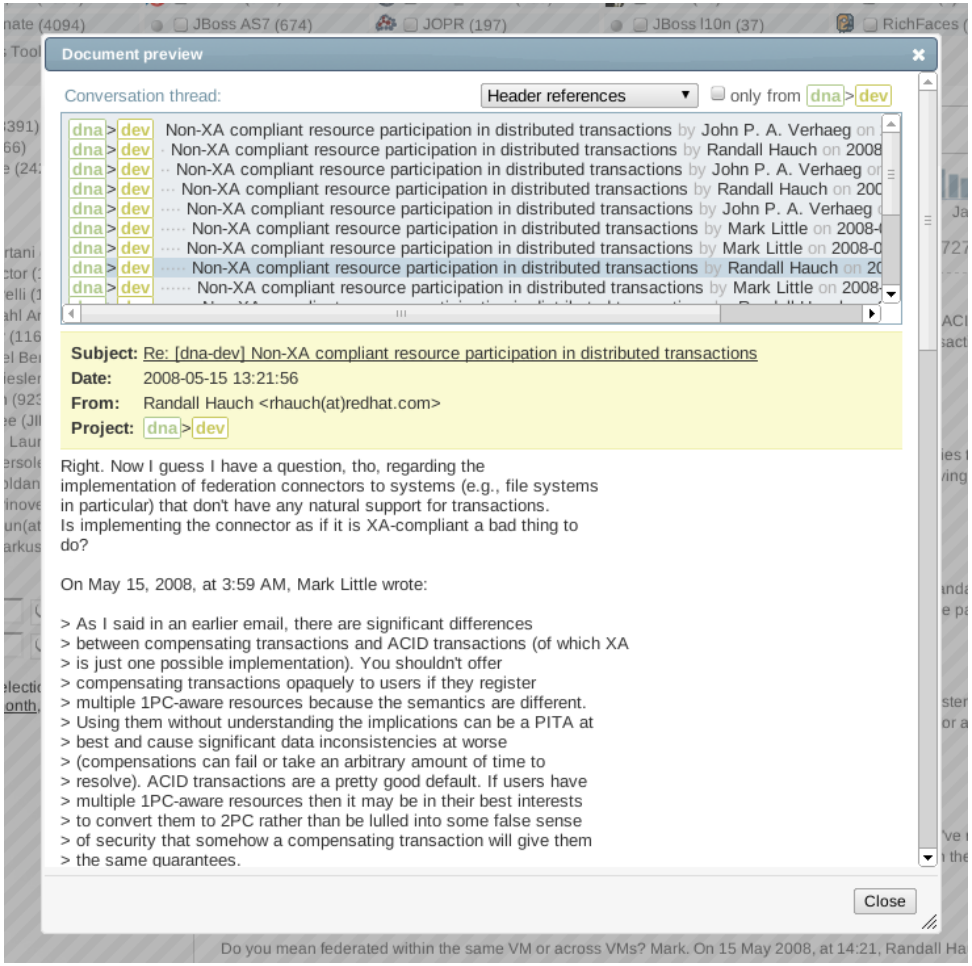
*Full example: https://gist.github.com/1012051*

# *facets.author_filter : {...}*

```
terms : { field : "from.not_analyzed", size : 15 },
facet_filter : {
  and : [
    { query :{ query_string : { query : "jboss server" }}},
    { terms :{ from.not_analyzed : [
        "Galder Zamarreno <galder.zamarreno@redhat.com>",
        "Pete Muir <pmuir@redhat.com>" ]}
    },
    { range :{ date :{ from :"2007-07-25", to :"2010-12-16"}}},
    { terms :{ _index :["weld"]}},
    { terms :{ mail_list :["dev"]}}
  ]
},
global : true
```

*Full example: https://gist.github.com/1012051*

# Document Detail Preview



# *Search query in detail*

```
{
  fields : [ ... ],
  query : { ... },
  filter : { ... },
  highlight : { ... }
}
```

*Full example: https://gist.github.com/1012096*

# *query : {...} & filter : {...}*

```
query : {
  bool : {
    should : [{
      query_string : { query : _userQuery_ }
    }],
    must : { term : { message_id : _documentId_ }}
  }
},
filter : { ids : { type : "mail", values : [ _documentId_ ] } }
```

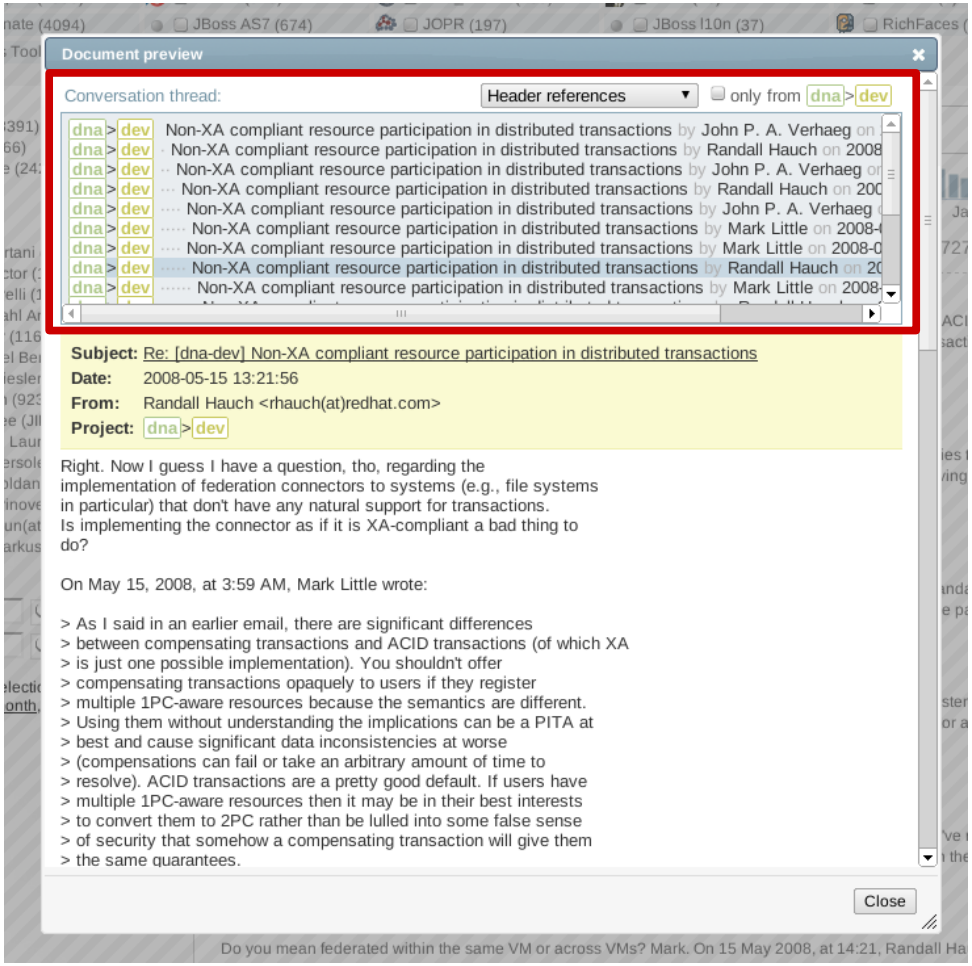
*Full example: https://gist.github.com/1012096*

# *highlight : {...}*

```
highlight : {
  pre_tags : [ "<span class='phlt'>" ],
  post_tags : [ "</span>" ],
  number_of_fragments : 0,
  fields : {
    subject_original : { },
    first_text_message : { },
    first_html_message : { },
    message_attachments : { number_of_fragments : 3 }
  }
}
```

*Full example: https://gist.github.com/1012096*

# Document Detail Preview




Threads

```
{from : 0, size : _count_,
  fields : [ ... ],
    script_fields : { millis : { script : "doc['date'].date.millis" }
    },
  query : {
    constant_score : {
      filter : {
        or : {
          filters : [
            { terms : { references : _references_ } },
            { terms : { message_id_original : _references_ } }
          ]}}}}}
```

*Full example: https://gist.github.com/1012115*

# *Challenges*

*Parsing and Indexing emails*

*Mail Subject normalization*

*Mail conversations*

*Quoted content*

*Author identification*

...

# *Challenges*

## *1) Parsing and indexing emails*

Emails can have very complex hierarchical structure with different content types in it. They can be even malformed (typically SPAM emails).

Apache **Mime4J** and **jsoup** can do a very good job.

ES mapping example: https://gist.github.com/1011913

# *Challenges*

## *2) Mail subject normalization*

Emails in public mailing lists contain specific patterns in subject that should be handled carefully. Probably you do not want to index them as a part of the mail subject at all.

# *Challenges*

## *2) Mail subject normalization*

Typically something like the following:

- [infinispan-dev] Feedback on Infinispan patch
- Re: [infinispan-dev] Feedback on Infinispan patch
- Re: [hibernate-dev] [infinispan-dev] Feedback on Infi...

# *Challenges*

## *2) Mail subject normalization*

But it can be very funny too:

[jbossws-users] [JBossWS] – Re:
[jbossws-users] [JBossWS] - "
[rules-dev] ;

# *Challenges*

## *2) Mail subject normalization*

Or quite complex:

[infinispan-dev] [Fwd: [Fwd: Sometimes TCP responses not getting through on localhost]]

[hibernate-dev] [Fwd: [redhat.com #1341720] [Fwd: Re: Unable to checkout core/trunk/core]]

# *Challenges*

## *3) Identification of conversation threads*

Probably the two biggest challenges are Thread hijacking and Non-linear threads.

Probably the two most straightforward approaches are using mail headers (<in-reply-to>,<references>) and mail subjects.

# *Challenges*

## *4) Quoted content*

Should the quoted content be included?

*(I think it should be included...)*

# *Challenges*

## *5) Author identification*

Email value is not always enough.

" 이희승  (Trustin Lee)" <trustin@gmail.com>
Trustin Lee <tlee@redhat.com>
"Trustin Lee ( 이희승 )" <trustin@gmail.com>
Trustin Lee ( 이희승 ) <trustin@gmail.com>
Trustin Lee <trustin@gmail.com>
Trustin Lee <trustin@gleamynode.net>
이희승  <trustin@gmail.com>

# *Want to try yourself?*

*You are welcome!*

*Some parts of the search application to be soon available on GitHub for anybody to index and search mbox files.*

# *And one more thing...*

*Don't go home emptyhanded.*

## *BigDesk*

*A tiny monitoring tool for Elastic Search.*

*https://github.com/lukas-vlcek/bigdesk*

*Thank you!*

*lvlcek@redhat.com*
*lukas.vlcek@gmail.com*