elasticsearch.

PLUGGABLE ARCHITECTURE UNDER THE HOOD

`Not affiliated with elasticsearch.org or elasticsearch.com`

Alexander Reelsen / @spinscale

~# whoami

```
~# whoami

Alexander Reelsen, almost 30, random developer

Likes: Elasticsearch, dropwizard, simple java web frameworks
Likes: Document stores, scaling web architectures

Wrote: Playframework cookbook

Day job: Software Engineer @ Lusini.de

Sports: Basketball, Badminton
```

AGENDA

- What is so important about search?
- Elasticsearch high level overview
- Configuration, Mapping & Analyzers
- Sharding, Replication
- Querying, Facetting & Percolation
- Plugins

AGENDA

Walking through the **FST suggester plugin**

- Plugin setup, request workflow
- REST endpoint
- Client integration
- Shardservice, Node Service
- Refreshing in-memory structures
- Lifecycle listeners
- Testing, Plugin assembly

AGENDA
IMPLEMENTING OWN PLUGINS

- Facets
- Rivers: JSON streaming river
- Analyzers: Morphologic analyzer
- Other plugins, drivers

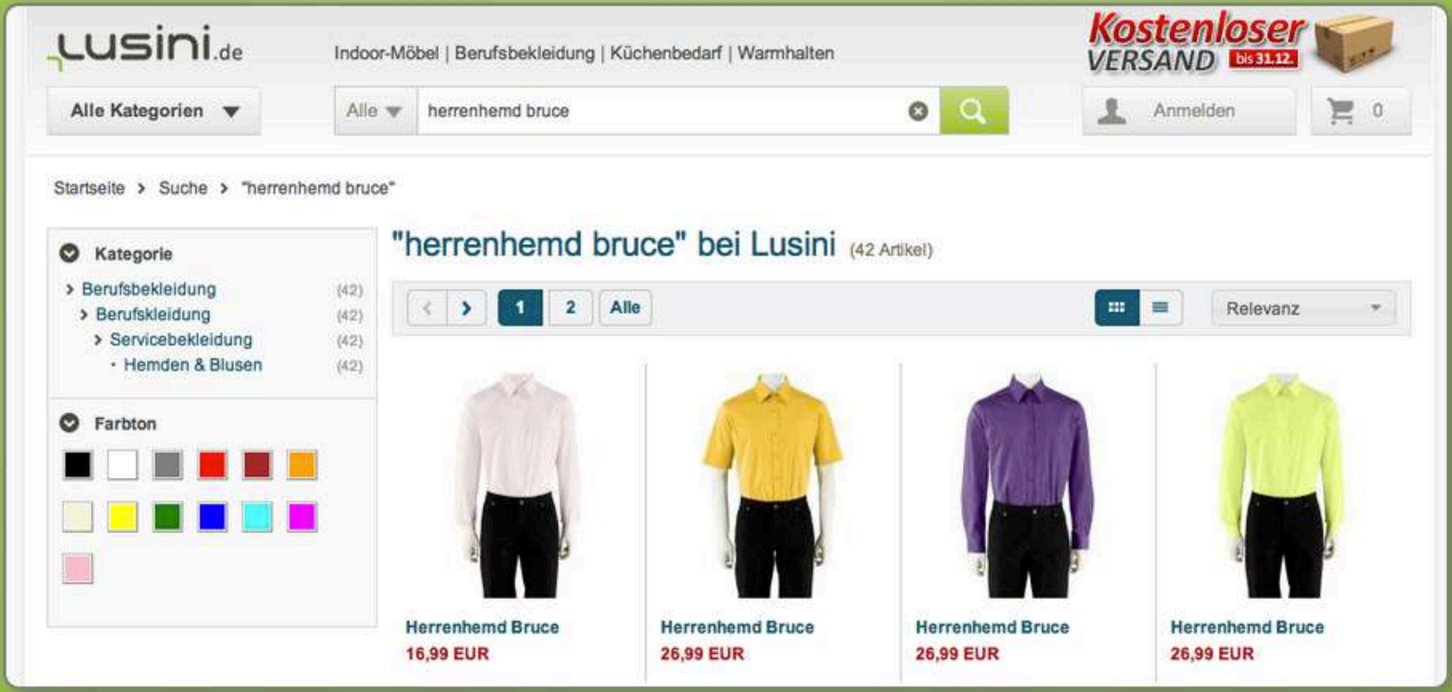
SEARCH IS HARD

Functional requirements
Find the right things (effectivity)

Non-functional requirements
Find the things right (efficiency)

NO EFFICIENCY WITHOUT EFFECTIVITY

TERM SEARCH



lusini.de

Indoor-Möbel | Berufsbekleidung | Küchenbedarf | Warmhalten

Kostenloser VERSAND bis 31.12.

Alle Kategorien

Alle

herrenhemd bruce

Anmelden

0

Startseite > Suche > "herrenhemd bruce"

Kategorie

> Berufsbekleidung	(42)
> Berufskleidung	(42)
> Servicebekleidung	(42)
· Hemden & Blusen	(42)

Farbton

"herrenhemd bruce" bei Lusini (42 Artikel)

1 2 Alle

Relevanz

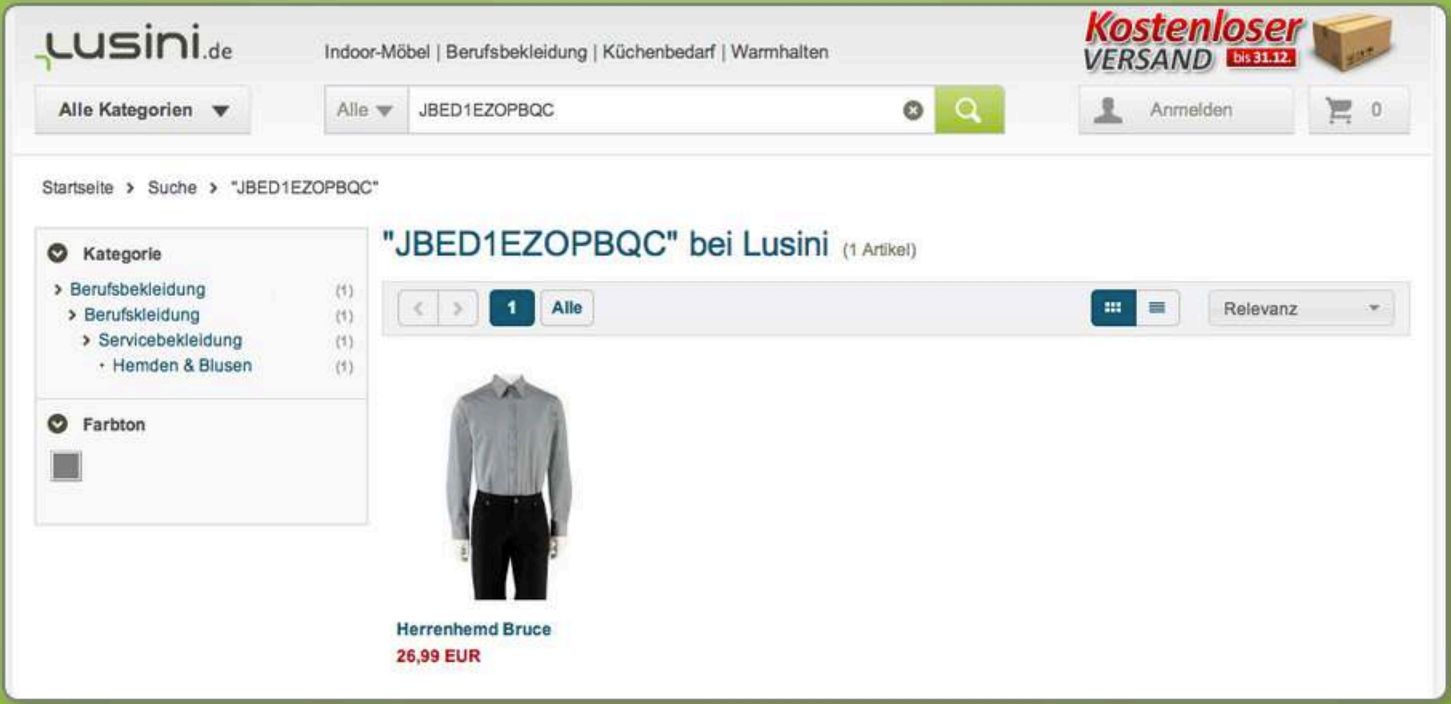
Herrenhemd Bruce
16,99 EUR

Herrenhemd Bruce
26,99 EUR

Herrenhemd Bruce
26,99 EUR

Herrenhemd Bruce
26,99 EUR

ID SEARCH



Lusini.de

Indoor-Möbel | Berufsbekleidung | Küchenbedarf | Warmhalten

Kostenloser VERSAND bis 31.12.

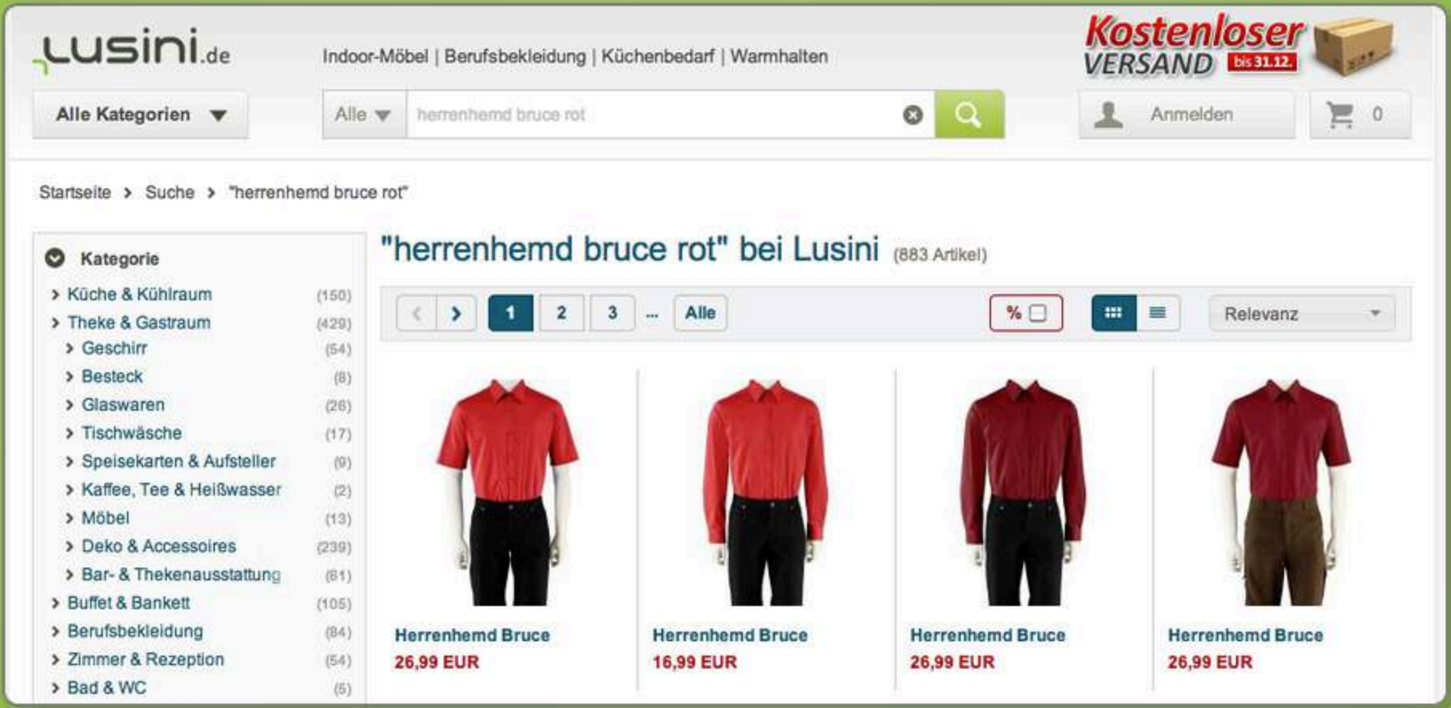
Alle Kategorien ▼ | Alle ▼ | JBED1EZOPBQC | Anmelden | 0

Startseite › Suche › "JBED1EZOPBQC"

"JBED1EZOPBQC" bei Lusini (1 Artikel)

Herrenhemd Bruce
26,99 EUR

SEARCH BY COLOR



lusini.de
Indoor-Möbel | Berufsbekleidung | Küchenbedarf | Warmhalten
Kostenloser VERSAND bis 31.12.
Alle Kategorien
Alle
herrenhemd bruce rot
Anmelden
0

Startseite › Suche › "herrenhemd bruce rot"

"herrenhemd bruce rot" bei Lusini (883 Artikel)

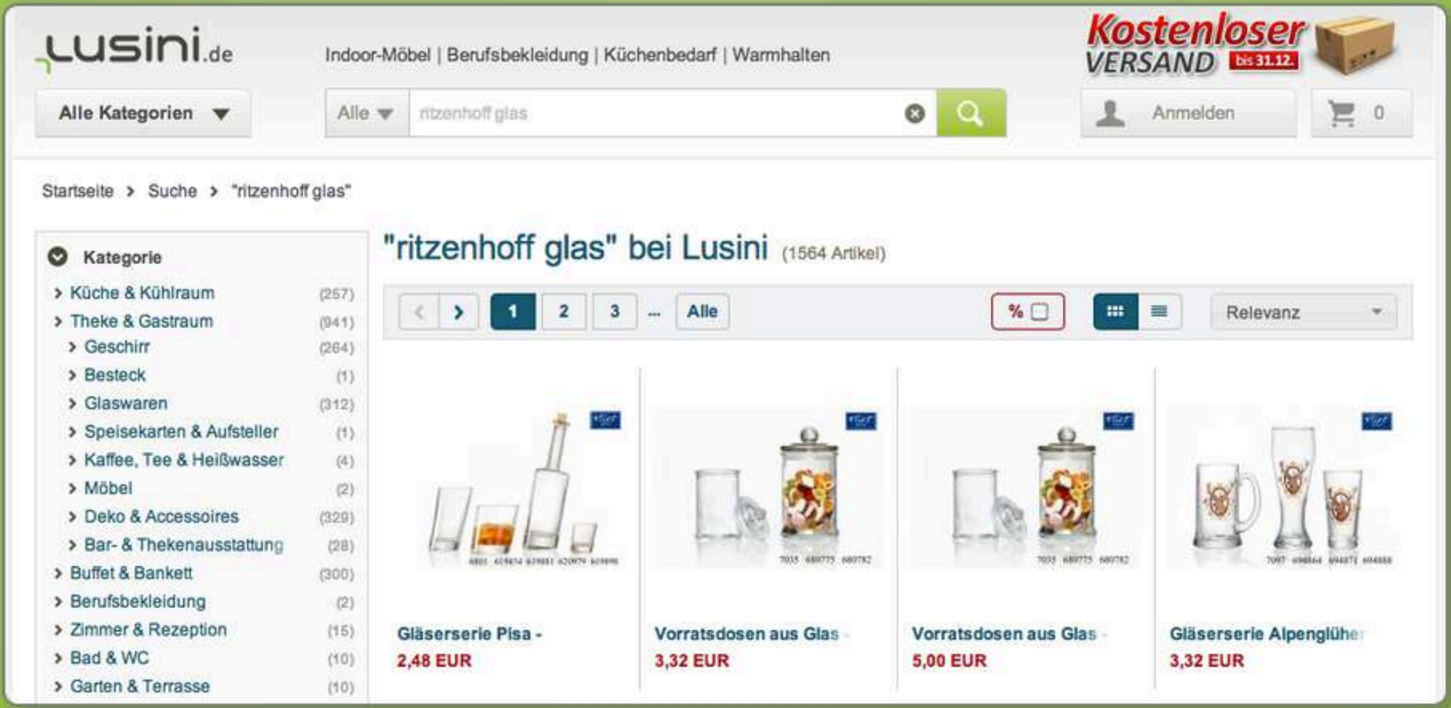
Kategorie

Küche & Kühlraum	(150)
Theke & Gastraum	(429)
Geschirr	(54)
Besteck	(8)
Glaswaren	(26)
Tischwäsche	(17)
Speisekarten & Aufsteller	(9)
Kaffee, Tee & Heißwasser	(2)
Möbel	(13)
Deko & Accessoires	(239)
Bar- & Thekenausstattung	(61)
Buffet & Bankett	(105)
Berufsbekleidung	(84)
Zimmer & Rezeption	(54)
Bad & WC	(5)

1 2 3 ... Alle
%
Relevanz

Herrenhemd Bruce	26,99 EUR
Herrenhemd Bruce	16,99 EUR
Herrenhemd Bruce	26,99 EUR
Herrenhemd Bruce	26,99 EUR

SEARCH BY BRAND



Lusini.de

Indoor-Möbel | Berufsbekleidung | Küchenbedarf | Warmhalten

Kostenloser VERSAND bis 31.12.

Alle Kategorien | Alle | ritzenhoff glas | Anmelden | 0

Startseite > Suche > "ritzenhoff glas"

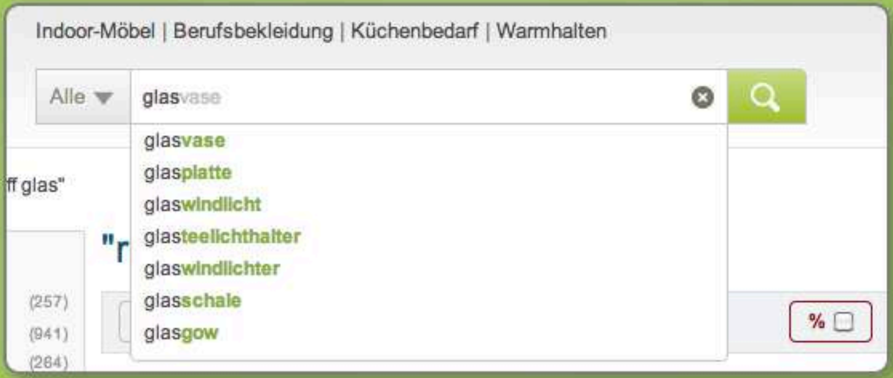
"ritzenhoff glas" bei Lusini (1564 Artikel)

Kategorie	
Küche & Kühlraum	(257)
Theke & Gastraum	(941)
Geschirr	(264)
Besteck	(1)
Glaswaren	(312)
Speisekarten & Aufsteller	(1)
Kaffee, Tee & Heißwasser	(4)
Möbel	(2)
Deko & Accessoires	(329)
Bar- & Thekenausstattung	(28)
Buffet & Bankett	(300)
Berufsbekleidung	(2)
Zimmer & Rezeption	(15)
Bad & WC	(10)
Garten & Terrasse	(10)

< > 1 2 3 ... Alle | % | Relevanz

Gläserserie Pisa -	Vorratsdosen aus Glas -	Vorratsdosen aus Glas -	Gläserserie Alpenglühe
2,48 EUR	3,32 EUR	5,00 EUR	3,32 EUR

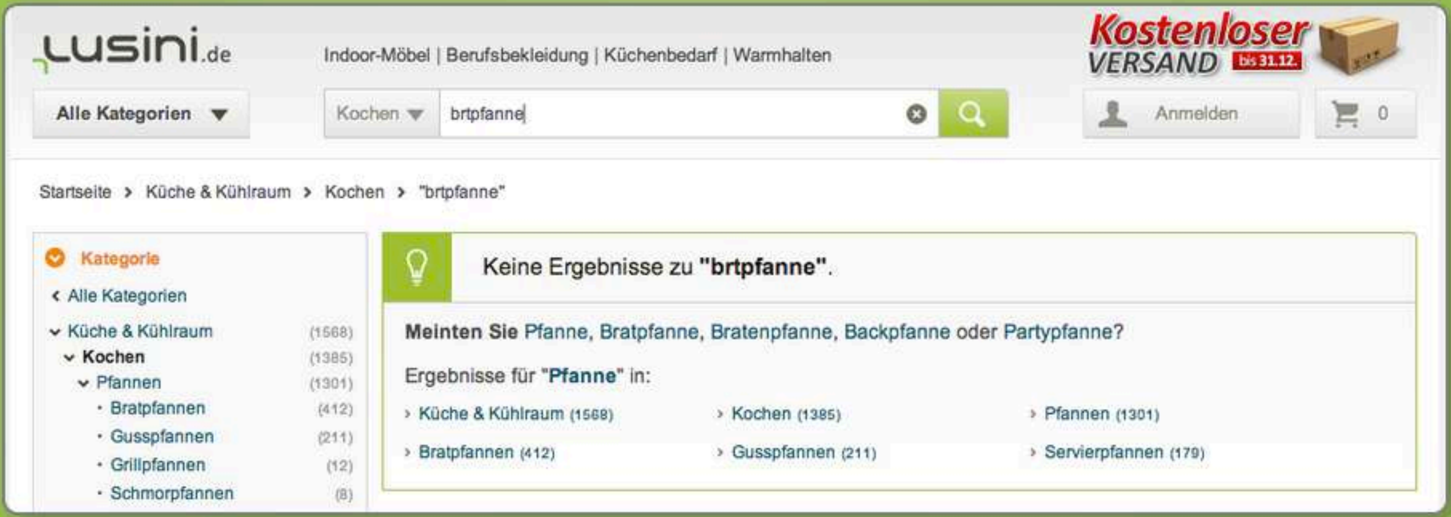
SUGGESTIONS



Indoor-Möbel | Berufsbekleidung | Küchenbedarf | Warmhalten

glasvase
glasplatte
glaswindlicht
glasteelichthalter
glaswindlichter
glasschale
glasgow

CORRECTIONS


lusini.de
Indoor-Möbel | Berufsbekleidung | Küchenbedarf | Warmhalten
Kostenloser VERSAND bis 31.12.
Alle Kategorien
Kochen
brtpfanne
Anmelden
0

Startseite › Küche & Kühlraum › Kochen › "brtpfanne"

Kategorie

‹ Alle Kategorien

Küche & Kühlraum	(1568)
Kochen	(1385)
Pfannen	(1301)
· Bratpfannen	(412)
· Gusspfannen	(211)
· Grillpfannen	(12)
· Schmorpfannen	(8)

Keine Ergebnisse zu **"brtpfanne"**.

Meinten Sie Pfanne, Bratpfanne, Bratenpfanne, Backpfanne oder Partypfanne?

Ergebnisse für "**Pfanne**" in:

› Küche & Kühlraum (1568)	› Kochen (1385)	› Pfannen (1301)
› Bratpfannen (412)	› Gusspfannen (211)	› Servierpfannen (179)

PREREQUISITES

- Basic elasticsearch experience
- Java experience
- Plus: Lucene, Guice, Guava, Jackson
- Plus: HTTP, REST, JSON

ELASTICSEARCH - BIRDS EYE VIEW

- Schema-free, REST and JSON based, document store **(having extra-ordinary search capabilites)**
- Multi-tenancy, distributed **(sharding, replication)**
- Facetting, highlighting, custom scripting & scoring
- Language specific drivers
- Highly extensible via plugins
- Zero configuration

ZERO CONFIGURATION - I KID U NOT!

```
~# wget --no-check-certificate https://github.com/downloads/elasti
csearch/elasticsearch/elasticsearch-0.19.11.zip

~# unzip elasticsearch-0.19.11.zip

~# cd elasticsearch-0.19.11

~# bin/elasticsearch -f
```

```
~# curl -X PUT http://localhost:9200/products/product/1 -d '{ "nam
e" : "high quality search engine" }'

{"ok":true,"_index":"products","_type":"product","_id":"1","_versi
on":1}
```

ZERO CONFIGURATION - I KID U NOT!

```
~# curl -X POST http://localhost:9200/products/product/_search?pre
tty=1 -d '{ "query" : { "term" : { "name" : "search" }} }'

{
  "took" : 2, "timed_out" : false,
  "_shards" : {
    "total" : 5, "successful" : 5, "failed" : 0
  },
  "hits" : {
    "total" : 1, "max_score" : 0.15342641,
    "hits" : [ {
      "_index" : "products", "_type" : "product", "_id" : "1",
      "_score" : 0.15342641, "_source" : { "name" : "high quality
search engine" }
    } ]
  }
}
```

CONFIGURATION

- `config/elasticsearch.json` or `config/elasticsearch.yml`
- Application-wide settings (zen discovery, available analyzers)
- Index default configurations (number of shards)
- Seperate logging file: config/logging.yml (simplified log4)

CONFIGURATION SAMPLE

```
discovery.zen.multicast.enabled: false

http:
  max_content_length: 100000

index:
  number_of_shards: 1

  analysis:
    analyzer:
      default:
        type: standard

      lowercase_analyzer:
        type: custom
        tokenizer: standard
        filter: [standard, lowercase]
```

MAPPING

- JSON data is parsed on indexing
- Mapping is done on first field indexing
- Inferred if not configured **(dangerous!)**
- Types: float, long, boolean, date (+formatting), object, nested
- String type can have arbitrary analyzers
- Fields can be split up in more fields

SAMPLE MAPPING

```
{
  "product": {
    "properties": {
      "ProductId":       { "type": "string", "index": "not_analyzed" },
      "ProductEnabled": { "type": "boolean" },
      "PiecesIncluded": { "type": "long" },
      "Price":           { "type": "float" },
      "LastModified": { "type": "date", "format": "yyyy-MM-dd HH:mm:ss.SSS" },

      "ProductName" : {
        "type" : "multi_field",
        "include_in_all" : true,
        "fields" : {
          "ProductName": { "type": "string", "index": "not_analyzed" },
          "lowercase": { "type": "string", "analyzer": "lowercase_analyzer" },
          "suggest" :  { "type": "string", "analyzer": "suggest_analyzer" }
        }
      }
```

ANALYZERS

- An analyzer consists of a tokenizer and an arbitrary amount of filters
- Example:

```
suggest_analyzer:
  type: custom
  tokenizer: whitespace
  filter: [standard, lowercase, shingle]
```

- Stripping html code:

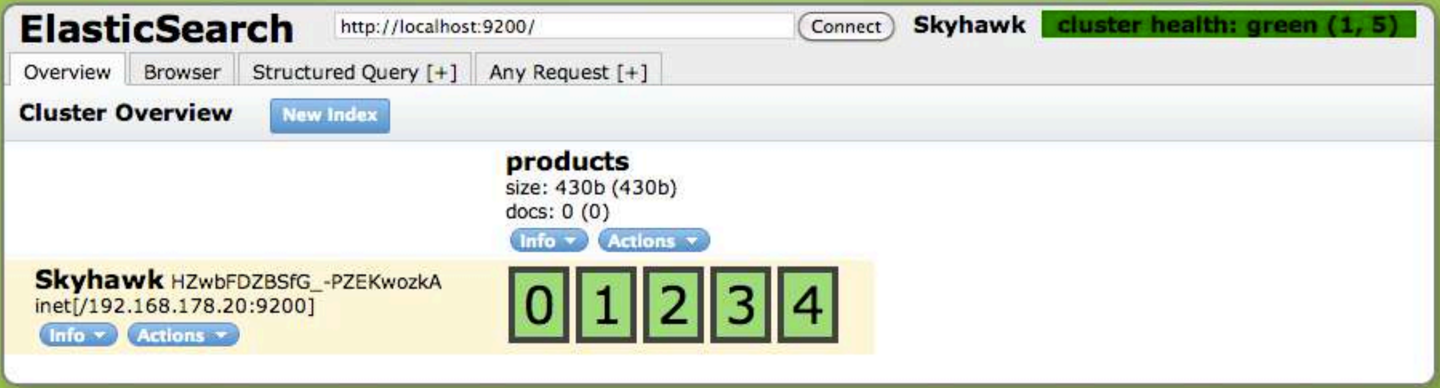
```
char_filter: html_strip
```

INDICES - BIRDS EYE VIEW

- Dataset ready to execute searches in
- An index consists of arbitrary amount of shards
- An index can span over arbitrary amount of nodes
- One shard is mapped to one lucene index

SHARDING

```
~# curl -X PUT localhost:9200/products -d '{ "settings" : { "index
" : { "number_of_shards" : "5", "number_of_replicas" : "0" } } }'
```

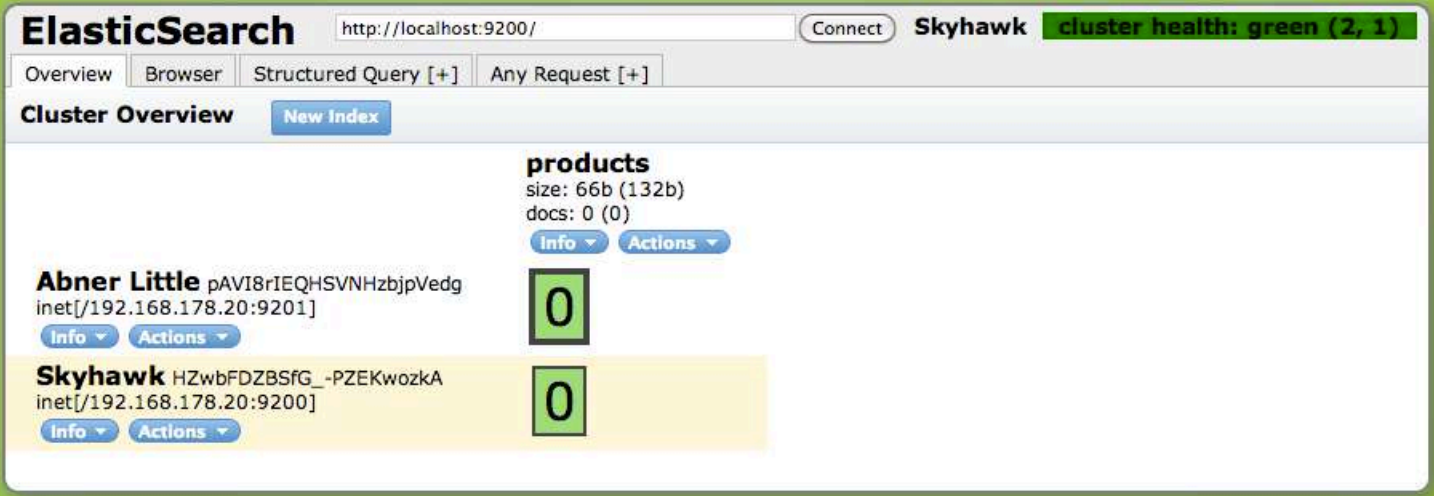


products
size: 430b (430b)
docs: 0 (0)

Skyhawk HZwbFDZBSfG_-PZEKwozkA
inet[/192.168.178.20:9200]

REPLICATION

```
~# curl -X PUT localhost:9200/products -d '{ "settings" : { "index
" : { "number_of_shards" : "1", "number_of_replicas" : "1" } } }'
```



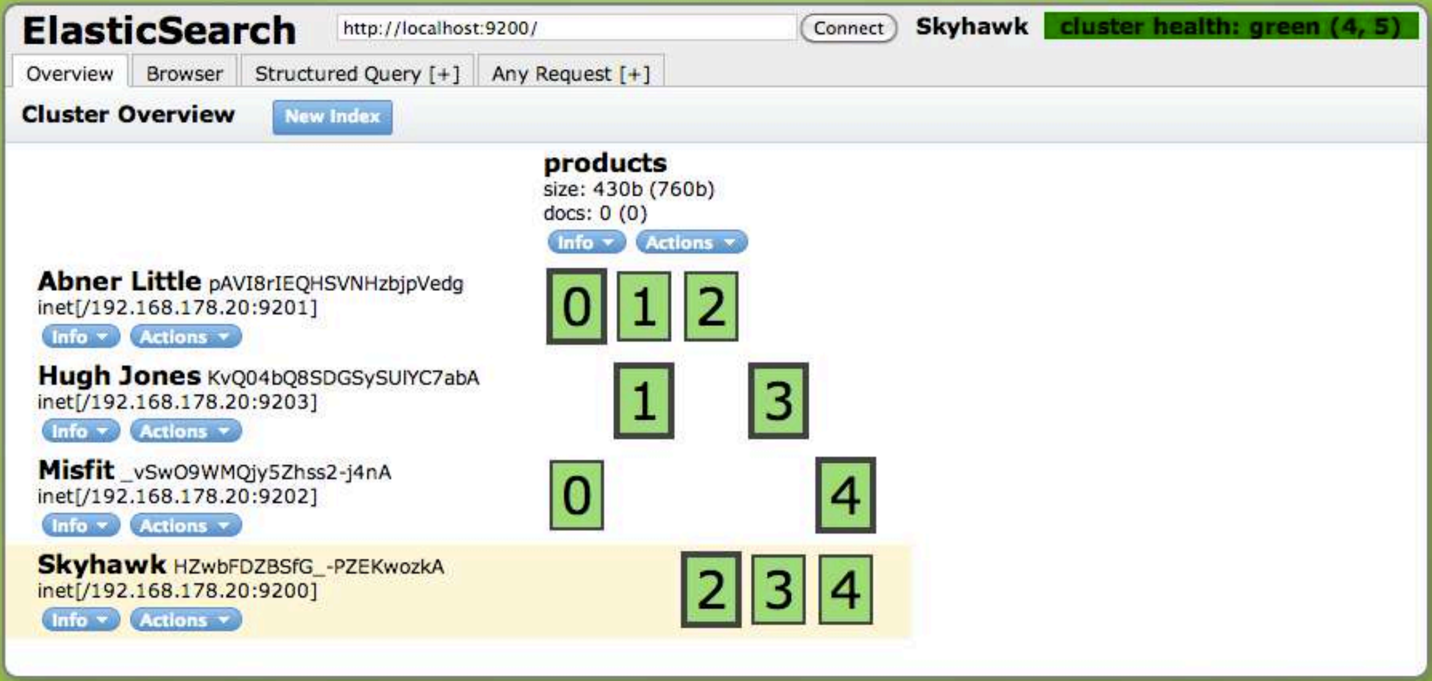
products
size: 66b (132b)
docs: 0 (0)

Abner Little pAVI8rIEQHSVNHzbjpVedg
inet[/192.168.178.20:9201]
0

Skyhawk HZwbFDZBSfG_-PZEKwozkA
inet[/192.168.178.20:9200]
0

SHARDING & REPLICATION

```
~# curl -X PUT localhost:9200/products -d '{ "settings" : { "index
" : { "number_of_shards" : "5", "number_of_replicas" : "1" } } }'
```



products
size: 430b (760b)
docs: 0 (0)

Abner Little pAVI8rIEQHSVNHzbjpVedg
inet[/192.168.178.20:9201]
0 1 2

Hugh Jones KvQ04bQ8SDGSySUIYC7abA
inet[/192.168.178.20:9203]
1 3

Misfit _vSwO9WMQjy5Zhss2-j4nA
inet[/192.168.178.20:9202]
0 4

Skyhawk HZwbFDZBSfG_-PZEKwozkA
inet[/192.168.178.20:9200]
2 3 4

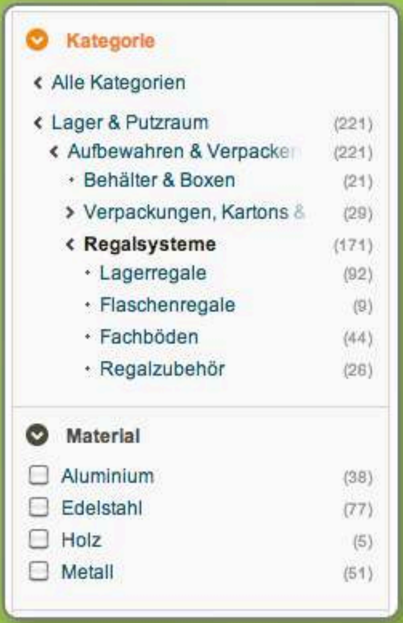
QUERYING

- Search/Count queries (term query, prefix query, id, fuzzy...)
- Geo-based queries, TTL
- More like this, Highlighting
- Facetting, Percolation, Scripting

SEARCH - FACETTING

- Facetting adds aggregated information to a standard search query
- Term: Group results by a term
- Range: Group by price or date ranges
- Histogram: Group results in equally sized buckets, also as date histogram
- Statistical: Include statistical data like min, max, sum, avg & some more
- Geo distance: Group results around a coordinate

SEARCH - FACETTING



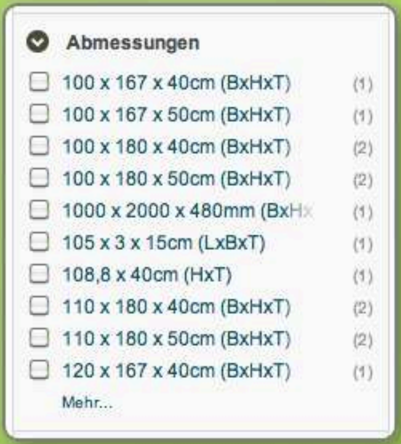
Kategorie

‹ Alle Kategorien

‹ Lager & Putzraum	(221)
‹ Aufbewahren & Verpacken	(221)
· Behälter & Boxen	(21)
› Verpackungen, Kartons &	(29)
‹ **Regalsysteme**	(171)
· Lagerregale	(92)
· Flaschenregale	(9)
· Fachböden	(44)
· Regalzubehör	(26)

Material

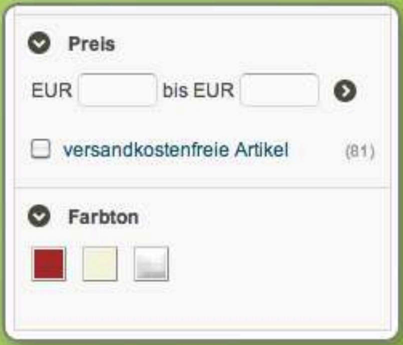
☐ Aluminium	(38)
☐ Edelstahl	(77)
☐ Holz	(5)
☐ Metall	(51)



Abmessungen

☐ 100 x 167 x 40cm (BxHxT)	(1)
☐ 100 x 167 x 50cm (BxHxT)	(1)
☐ 100 x 180 x 40cm (BxHxT)	(2)
☐ 100 x 180 x 50cm (BxHxT)	(2)
☐ 1000 x 2000 x 480mm (BxHx	(1)
☐ 105 x 3 x 15cm (LxBxT)	(1)
☐ 108,8 x 40cm (HxT)	(1)
☐ 110 x 180 x 40cm (BxHxT)	(2)
☐ 110 x 180 x 50cm (BxHxT)	(2)
☐ 120 x 167 x 40cm (BxHxT)	(1)

Mehr...



Preis

EUR ____ bis EUR ____

☐ versandkostenfreie Artikel (81)

Farbton

🟥 ⬜ ⬜

SEARCH - SCRIPTING

THIS IS WHERE YOUR OWN INTEGRATION BEATS ALL OTHERS

- Score down all your products without an image
- Score up products by an attribute like its product quality or stock
- Change score depending on the age of the product (score higher if auction ends today)
- Apply math operations on fields to change score

SEARCH API - PERCOLATION
IMPLEMENT A PRICE AGENT FOR FREE!

```
curl -X PUT localhost:9200/_percolator/products/pricecheck -d '{
"query" : {
  "bool" : {
    "must" : { "term" : { "name" : "MacBook Air" } },
    "must" : { "range" : { "price" : { "from" : 200, "to" : 999 }
} }
    }
  }
}'
{"ok":true,"_index":"_percolator","_type":"products","_id":"pricec
heck","_version":1}
```

SEARCH API - PERCOLATION

```
curl -X PUT 'localhost:9200/products/product/1?percolate=*' -d '{
"price": 1000, "name" : "MacBook Air" }'
{"ok":true,"_index":"products","_type":"product","_id":"1","_versi
on":1,"matches":[ ]}

curl -X PUT 'localhost:9200/products/product/2?percolate=*' -d '{
"price": 999, "name" : "MacBook Air" }'
{"ok":true,"_index":"products","_type":"product","_id":"2","_versi
on":1,"matches":["pricecheck"]}
```

EXECUTING A QUERY

```
Settings settings = ImmutableSettings.settingsBuilder().put("node.
client", true).build();

InetSocketTransportAddress address = new InetSocketTransportAddres
s("localhost", 9300);

TransportClient client = new TransportClient(settings).addTranspor
tAddress(address);

SearchRequest searchRequest = new SearchRequest("products");
searchRequest.source(new SearchSourceBuilder(termQuery("field", "s
earch")))
searchRequest.types("product")

ActionFuture<SearchResponse> response = client.searchRequest(searc
hRequest);
```

REQUEST BUILDERS

```
SearchRequestBuilder builder = new SearchRequestBuilder(client)
    .setIndices("products")
    .setTypes("product")
    .setQuery(QueryBuilders.termQuery("name", "foo"))
    .execute()
    .actionGet()
```

GOING ASYNC

```
SearchRequestBuilder builder = new SearchRequestBuilder(node.clien
t())
    .setIndices("products")
    .setTypes("product")
    .setQuery(QueryBuilders.termQuery("name", "foo"))
    .execute(new ActionListener<SearchResponse>() {

        public void onResponse(SearchResponse searchResponse) {
            // ...
        }

        public void onFailure(Throwable e) {
            // ...
        }
    });
```

GOOGLE GUICE - MODULES

```
public class BillingModule extends AbstractModule {
  @Override
  protected void configure() {
    bind(TransactionLog.class).to(DatabaseTransactionLog.class);
    bind(CreditCardProcessor.class).to(PaypalCreditCardProcessor.c
lass);
    bind(BillingService.class).to(RealBillingService.class);
  }
}
```

Sample from guice homepage

GOOGLE GUICE - @INJECT

```java
public class RealBillingService implements BillingService {
  private final CreditCardProcessor processor;
  private final TransactionLog transactionLog;

  @Inject
  public RealBillingService(CreditCardProcessor processor,
      TransactionLog transactionLog) {
    this.processor = processor;
    this.transactionLog = transactionLog;
  }

// ...
}
```

GOOGLE GUICE - INJECTOR

```
public class App {

  public static void main(String[] args) {
    Injector injector = Guice.createInjector(new BillingModule());
    BillingService billingService = injector.getInstance(BillingSe
rvice.class);
    ...
  }

}
```

ELASTICSEARCH - GUICE 101

```
ModulesBuilder modules = new ModulesBuilder();
modules.add(new RestModule(settings));
// modules.add(AnyModule.class);

Injector injector = modules.createInjector();
Client client = injector.getInstance(Client.class);
```

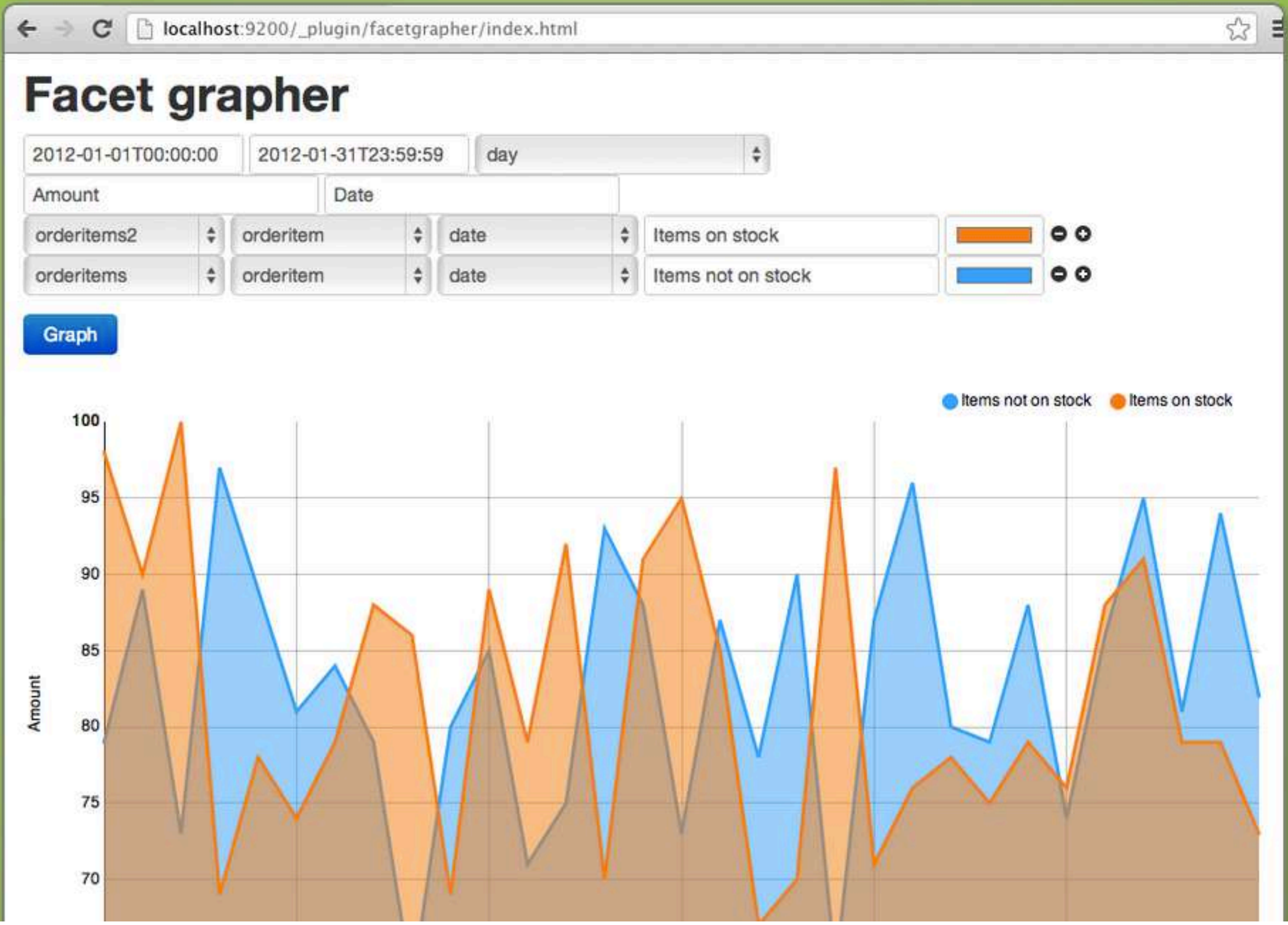
- ModuleBuilder: Node, TransportClient, Index, IndexShard and River

PLUGINS

- Extending anything: Analyzers, rivers, transport, facets, mapping types, discovery, scripting, lucene functionality
- Many plugins available
- Site plugin: Statically serving assets

```
bin/plugin -install spinscale/elasticsearch-facetgrapher
```

SITE PLUGINS



localhost:9200/_plugin/facetgrapher/index.html

Facet grapher

2012-01-01T00:00:00	2012-01-31T23:59:59	day		
Amount	Date			
orderitems2	orderitem	date	Items on stock	🟧
orderitems	orderitem	date	Items not on stock	🟦

PLUGINS

On application start up, `PluginsService` scans for all `es-plugin.properties` files.

```
plugin=org.elasticsearch.plugin.suggest.SuggestPlugin
```

PLUGINS

```java
public class SuggestPlugin extends AbstractPlugin {

    public String name() {
        return "suggest";
    }

    public String description() {
        return "Suggest Plugin";
    }

    public void onModule(RestModule restModule) {
        restModule.addRestAction(RestSuggestAction.class);
    }
}
```

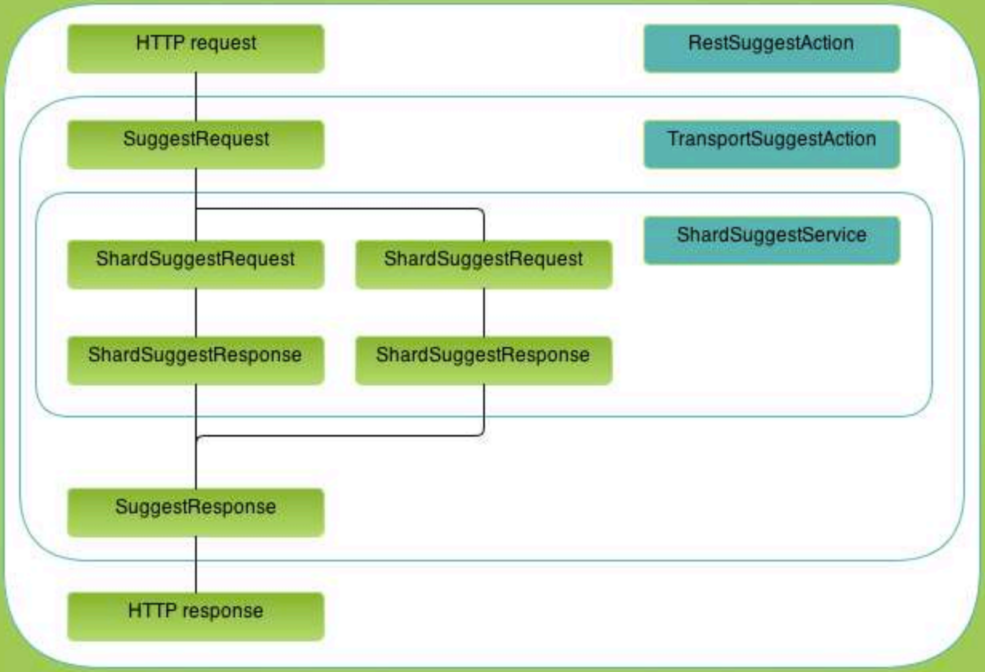
PLUGINS

```
Collection<Class<? extends Module>> modules() {}

Collection<Class<? extends Module>> indexModules() {}

Collection<Class<? extends Module>> shardModules() {}

Collection<Class<? extends LifecycleComponent>> services() {}

Collection<Class<? extends LifecycleComponent>> indexServices() {}

Collection<Class<? extends LifecycleComponent>> shardServices() {}

public void processModule(Module module) {}
```

FST SUGGESTER PLUGIN

- FST - Finite state automata, see this presentation
- Allow auto completion via FSTCompletionLookup
- In-memory structure (per IndexReader)
- Concurrent updates do not perform
- Solution: async updates

SIMPLIFIED EXECUTION FLOW



DEFINING REST ENDPOINTS

```
public class RestSuggestAction extends BaseRestHandler {

  @Inject
  public RestSuggestAction(Settings settings, Client client, RestC
ontroller controller) {
    super(settings, client);
    controller.registerHandler(POST, "/{index}/_suggest", this);
    controller.registerHandler(POST, "/{index}/{type}/_suggest", t
his);
  }

  public void handleRequest(final RestRequest request, final RestC
hannel channel) {
    // execute async request code here
  }
}
```

EXECUTING SUGGEST REQUEST

```
client.execute(SuggestAction.INSTANCE, suggestRequest, new ActionL
istener() {
  public void onResponse(SuggestResponse response) {
    try {
      // create JSON response of SuggestResponse
      channel.sendResponse(new XContentRestResponse(request, OK, b
uilder));
    } catch (Exception e) {
      onFailure(e);
    }
  }

  public void onFailure(Throwable e) {
    // return HTTP error message ...
  }
});
```

SHARD SERVICE

```
public class ShardSuggestService extends AbstractIndexShardCompone
nt {
    @Inject
    public ShardSuggestService(ShardId shardId, @IndexSettings Set
tings indexSettings, IndexShard indexShard) {
      // ...
    }

  public ShardSuggestResponse suggest(ShardSuggestRequest shardSug
gestRequest) {}

  public void update() {}
  public ShardSuggestRefreshResponse refresh(ShardSuggestRefreshRe
quest shardSuggestRefreshRequest) {}

  public void shutDown() {}
}
```

NODE SERVICES

```
public class SuggestService extends AbstractLifecycleComponent {
  @Inject
  public SuggestService(Settings settings, TransportSuggestRefresh
Action suggestRefreshAction, ClusterService clusterService, Indice
sService indicesService) {
  }

  protected void doStart() throws ElasticSearchException {
    // start thread which periodically sends refresh suggest reque
st
    suggestUpdaterThread = EsExecutors.daemonThreadFactory(setting
s, "suggest_updater").newThread(new SuggestUpdaterThread());
    suggestUpdaterThread.start();
  }

}
```

SUGGESTREFRESH

Quiz: Why is this code necessary?

```
public class SuggestUpdaterThread implements Runnable {
  public void run() {
    while (!closed) {
      DiscoveryNode node = clusterService.localNode();
      boolean isClusterStarted = clusterService.lifecycleState().e
quals(Lifecycle.State.STARTED);

      if (isClusterStarted && node != null && node.isMasterNode())
 {
        suggestRefreshAction.execute(new SuggestRefreshRequest()).
actionGet();
      }

      // sleep until next run here
    }
  }
}
```

LIFECYCLE LISTENER

Freeing up resources on shard deletion/cluster move
(and index close)

```
indicesService.indicesLifecycle().addListener(
new IndicesLifecycle.Listener() {
  public void beforeIndexShardClosed(ShardId shardId, @Nullable In
dexShard indexShard, boolean delete) {
    IndexService indexService = indicesService.indexService(shardI
d.index().name());
    if (indexService != null) {
      ShardSuggestService suggestShardService = indexService.shard
InjectorSafe(shardId.id()).getInstance(ShardSuggestService.class);
      suggestShardService.shutDown();
    }
  }
});
```

TRANSPORTCLIENT

Problem: Suggestions should work on clients, without having a node running in the same JVM try to start up server functionality

```
public class SuggestPlugin extends AbstractPlugin {
  public Collection<Class<? extends Module>> modules() {

    Collection<Class<? extends Module>> modules = Lists.newArrayLi
st();
    if (isClient()) {
      modules.add(SuggestClientModule.class);
    } else {
      modules.add(SuggestModule.class);
    }
    return modules;
  }
}
```

TESTING

- `AbstractSuggestTest`: Defines all tests, and amount of nodes running for tests (so clustered environment is also tested)
- Abstract methods: `getSuggestions()` and refreshing suggesters (all, per-index, per-field)
- `RestSuggestActionTest`, `TransportClientTest`, `TransportSuggestActionTest`, `SuggestBuildersTest`
- *Hint*: Start up nodes in parallel for speedup

PLUGIN ASSEMBLY

- maven assembly plugin is used to create a zip file
- Hooks up in the `package` lifecycle
- zip file includes
  - `mvn package` created jar
  - dependencies from `compile` scope
- zip file excludes the `org.elasticsearch:elasticsearch` dependency

FACETS

Implement own facet types to

... be faster

... save memory

... be more flexible

... save post processing work

SAMPLES

- Approximately date facets using stream-lib. See the presentation. *"Counts over 1 million distinct items in 80kb of memory to be accurate within 1% of true value"*
- Lovely systems facet plugins
- Lettercount plugin, plus presentation
- Script facet plugin

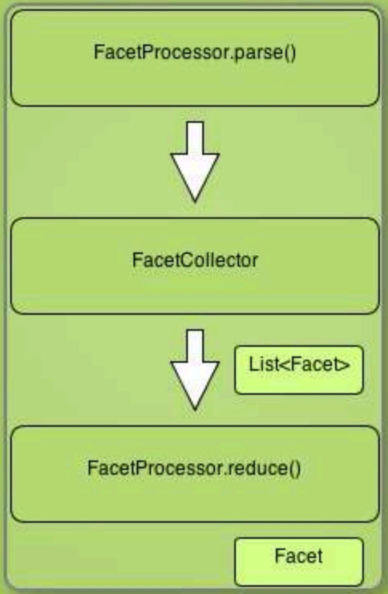
IMPLEMENTATION

- Plugins adds a `FacetProcessor` in the `FacetModule`
- Concrete facet processor implements `FacetProcessor`
- Concrete Facet implements `InternalFacet`
- Concrete facet collector extends `AbstractFacetCollector`

WORKFLOW

- `FacetProcessor` parses possible request options, creates `FacetCollector`
- `FacetCollector` operates per shard, gets docids from matched query, looks up doc values in the fieldCache and emits a `Facet`
- `FacetProcessor` gets a `List<Facet>`, merges it to one `Facet`
- One Facet is returned as facet data

WORKFLOW


FacetProcessor.parse()
FacetCollector
List<Facet>
FacetProcessor.reduce()
Facet

IDEA

- A geo region facet type
- Allows you to define arbitrary non-overlapping geo polygons as regions
- Facet groups results into these regions
- Would allow you to cluster/facet by country/state
- Saves classical pre/post processing work
- Could easily be done by a term facet, if region is added on indexing as string

RIVERS

- Automatically pull data into elasticsearch
- Run only on a single node in the cluster
- CouchDB, RabbitMQ, Twitter, wikipedia, RSS, MongoDB, JDBC, LDAP, ActiveMQ, Confluence, Solr, JSON, St9, MySQL, OAI, AWS SQS, Filesystem, Dropbox

```
curl -XPUT 'localhost:9200/_river/my_river/_meta' -d '{
  "type" : "dummy",
  "config" : {
    "user" : "foo",
    "password" : "bar"
  }
}'
```

JSON STREAMING

- Imports/Deletes JSON data
- Makes bulk requests
- Prevents going OOM due to execute indexing requests regularly
- Really streams data via HTTP
- Tries to fetch incremental updates

IMPLEMENTATION

```
public class JsonRiverPlugin extends AbstractPlugin {

  public String name() {
    return "json";
  }

  public String description() {
    return "River Streaming JSON Plugin";
  }

  public void onModule(RiversModule module) {
    module.registerRiver("json", JsonRiverModule.class);
  }
}
```

JSONRIVERMODULE

```java
public class JsonRiverModule extends AbstractModule {

    protected void configure() {
        bind(River.class).to(JsonRiver.class).asEagerSingleton();
    }
}
```

JSONRIVER

```
public class JsonRiver extends AbstractRiverComponent implements R
iver {

private volatile Thread slurperThread;
private volatile Thread indexerThread;
private final TransferQueue<RiverProduct> stream = new LinkedTrans
ferQueue<RiverProduct>();

}
```

SLURPER

```java
class Slurper implements Runnable {
  public void run() {
    RiverImporter importer = new RiverImporter("URL", stream);

    while (!closed) {
      String lastIndexUpdate = getLastUpdatedTimestamp();
      RiverProductImport result = importer.executeImport(lastIndex
Update);
      storeLastUpdatedTimestamp(result.exportTimestamp);

      try {
        Thread.sleep(RIVER_REFRESH_INTERVAL.getMillis());
      } catch (InterruptedException e1) {}
    }
  }
}
```

INDEXER

```
private class Indexer implements Runnable {
  public void run() {
    while (!closed) {
      try {
        RiverProduct product = stream.take();
        bulk = client.prepareBulk();
        do {
          addProductToBulkRequest(product);
        } while ((product = stream.poll(250, TimeUnit.MILLISECONDS
)) != null && deletedDocuments + insertedDocuments < RIVER_MAX_BUL
K_SIZE);
      } finally {
        bulk.execute().actionGet();
      }
    }
  }
}
```

ANALYZERS

- Extend elasticsearch with existing lucene analyzers
- Lots of analyzers: ICU, SmartCN, Phonetic, Stempel, Kuromoji, Decompounder, Combo, NaturalSort, SKOS, Hunspell, IK, MMSEG, Pinyin, Morphology, Paoding, Morfologik, Korean, Korean 2, Japanese, greeklish
- Attention: Not all analyzers are maintained

MORPHOLOGIC

```
public class AnalysisMorphologyPlugin extends AbstractPlugin {

    public String name() {
        return "analysis-morphology";
    }

    public String description() {
        return "Morphology analysis support";
    }

    public void onModule(AnalysisModule module) {
        module.addProcessor(new MorphologyAnalysisBinderProcessor(
));
    }
}
```

BINDERPROCESSOR

```
public class MorphologyAnalysisBinderProcessor extends AnalysisMod
ule.AnalysisBinderProcessor {

  public void processAnalyzers(AnalyzersBindings analyzersBindings
) {
    analyzersBindings.processAnalyzer("english_morphology", Englis
hMorphologyAnalyzerProvider.class);
  }

  public void processTokenFilters(TokenFiltersBindings tokenFilter
sBindings) {
    tokenFiltersBindings.processTokenFilter("english_morphology",
EnglishMorphologyTokenFilterFactory.class);
  }
}
```

ANALYZERPROVIDER

```
public class EnglishMorphologyAnalyzerProvider extends AbstractInd
exAnalyzerProvider<MorphologyAnalyzer> {

  private final MorphologyAnalyzer analyzer;

  @Inject
  public EnglishMorphologyAnalyzerProvider(...) {
    super(index, indexSettings, name, settings);
    try {
      analyzer = new EnglishAnalyzer();
    } catch (IOException ex) { /* ... */ }
  }

  public MorphologyAnalyzer get() {
    return this.analyzer;
  }
}
```

TOKENFILTERFACTORY

```
public class EnglishMorphologyTokenFilterFactory extends AbstractT
okenFilterFactory {

    private final LuceneMorphology luceneMorph;

    @Inject
    public EnglishMorphologyTokenFilterFactory(...) {
        super(index, indexSettings, name, settings);
        try {
            luceneMorph = new EnglishLuceneMorphology();
        } catch (IOException ex) {  /* ... */ }
    }

    public TokenStream create(TokenStream tokenStream) {
        return new MorphologyFilter(tokenStream, luceneMorph);
    }
}
```

TRANSPORT PLUGINS

- Provides alternative access to elasticsearch via different APIs
- memcached, servlet, Thrift, Jetty, websocket, ZeroMQ, SSL, Couchbase

OTHER PLUGINS

- Scripting plugins: python, JavaScript, Groovy, Ruby, Scala, and of course native scripts
- Site plugins: BigDesk, Paramedic, Facetgrapher, Elasticsearch Head
- Misc plugins: Mapper, Hadoop, AWS Cloud, Zookeeper, FST suggester, reindex, change tracking , partial update, Mock Solr, knapsack, langdetect, index term list, skywalker

DRIVERS & INTEGRATIONS

- Drivers & integrations: Official elasticsearch page, CakePHP, Mongoose
- Monitoring: Server Density, collectd, Munin gists, Nagios, Munin
- Sysadmins: starcluster, MCollective

SOON...

- Lucene 4.0 migration is ongoing
- Result grouping (#256)
- More native autocomplete (AnalyzingSuggester in Lucene 4.1)
- No public roadmap right now

DONE

- Questions?
- Critics?
- Tips?

BOOKS

- Sorry, no elasticsearch book yet...
- Lucene in Action (pre 4.0)
- Taming Text
- Introduction to Information Retrieval
- Modern information retrieval

LINKS & ACKNOWLEDGEMENTS

- Hakim El Hattab for reveal.js
- The whole elasticsearch team
- Creating a pluggable REST endpoint
- Solr vs. elasticsearch comparison, part 1, 2, 3, 4, 5
- How to write a simple plugin
- All the elasticsearch plugin authors