Simple Searching with ElasticSearch

Richard Miller
Jeremy Mikola

Symfony Live London
September, 14, 2012

Who are we?

Richard Miller, @mr_r_miller

Software Engineer @SensioLabsUK

Jeremy Mikola, @jmikola

Software Engineer @10gen

Search is important.

Database searching is good enough, though, right?

But search engines are difficult to install, configure, and use.

ElasticSearch

http://www.elasticsearch.org/

ElasticSearch in a nutshell

- Based on Lucene
- Schema-less
- RESTful
- Document-oriented (JSON)
- Fast and scalable

Who is using ElasticSearch?

- Mozilla
- StumbleUpon
- Klout
- GOV.UK
- FourSquare

Getting Started

- Download
  - http://www.elasticsearch.org/download/
- Launch
  - Shell script (background/foreground)
  - Service
- Configuration (optional)
  - Runtime parameters
  - File-based
  - REST API

Indexes, Types

URL Structure

http://localhost:9200/cookbook/recipes/

Inserting Documents

```
$ curl -XPOST http://localhost:9200/cookbook/recipes -d '{
  "name": "Welsh Rarebit",
  "tags": ["cheese", "bread"]
}'

{
  "ok": true,
  "_index": "cookbook",
  "_type": "recipes",
  "_id": "IcYOL_NuT-ymRwI4lz2NyA",
  "_version": 1
}
```

Inserting Documents

```
$ curl -XPOST http://localhost:9200/cookbook/recipes/2 -d '{
  "name": "Beef Wellington",
  "tags": ["beef"]
}'

{
  "ok": true,
  "_index": "cookbook",
  "_type": "recipes",
  "_id": "2",
  "_version": 1
}
```

Inserting Documents

```
$ curl -XPUT http://localhost:9200/cookbook/recipes/3 -d '{
  "name": "Yorkshire Pudding",
  "tags": ["pastry"]
}'

{
  "ok": true,
  "_index": "cookbook",
  "_type": "recipes",
  "_id": "3",
  "_version": 1
}
```

Updating Documents

```
$ curl -XPOST http://localhost:9200/cookbook/recipes/2 -d '{
  "name": "Beef Wellington",
  "tags": ["beef", "steak", "pastry"]
}'

{
  "ok": true,
  "_index": "cookbook",
  "_type": "recipes",
  "_id": "2",
  "_version": 2
}
```

Basic Searching with URI Requests

```
$ curl -XGET http://localhost:9200/cookbook/recipes/_search?q=tags:pastry
```

```
{
  "took": 31, "timed_out": false, "_shards": { "total": 5, "successful": 5, "failed": 0
},
  "hits": {
    "total": 2, "max_score": 0.5, "hits": [
      { "_index": "cookbook", "_type": "recipes", "_id": "2", "_score": 0.5,
        "_source" : { "name": "Beef Wellington", "tags": ["beef", "steak", "pastry"] }
},
      { "_index": "cookbook", "_type": "recipes", "_id": "3", "_score": 0.30685282,
        "_source" : { "name": "Yorkshire Pudding", "tags": ["pastry"] } }
    ]
  }
}
```

Querying Across Indexes and Types

```
$ curl -XGET http://localhost:9200/cookbook/recipes,foods/_search?q=tags:
pastry

$ curl -XGET http://localhost:9200/cookbook/_search?q=tags:pastry

$ curl -XGET http://localhost:9200/cookbook,guide/_search?q=tags:pastry

$ curl -XGET http://localhost:9200/_all/recipes/_search?q=tags:pastry

$ curl -XGET http://localhost:9200/_search?q=tags:pastry
```

Advanced Searching with Query DSL

- Basic queries
  - Term(s)
  - Prefix
  - Fuzzy
  - Range
- Compound queries
  - Bool
  - Disjunction max
  - Constant score
- Filtered
- Faceted
- "More like this"

Filters

- Not scored
- Cacheable
- Familiar operators
- Boolean logic
- Geospatial

Query DSL in JSON Request Body

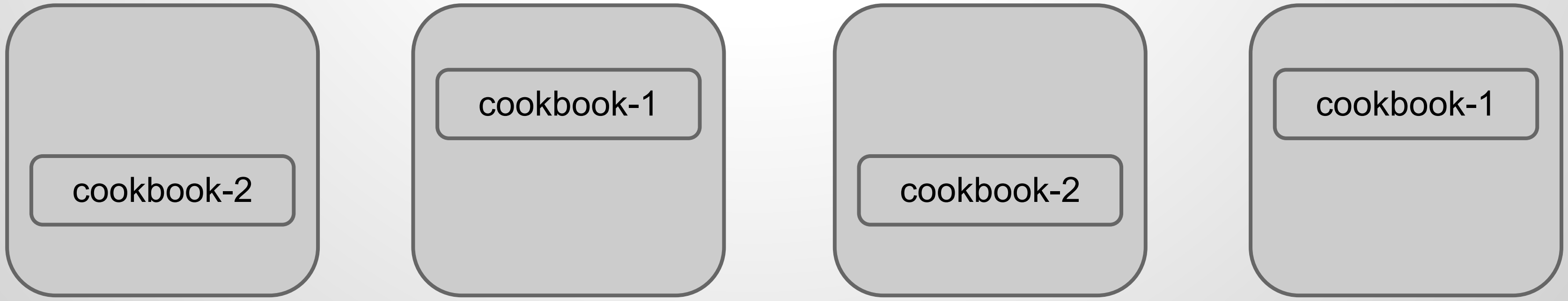
```
$ curl -XGET http://localhost:9200/cookbook/recipes/_search -d '{
  "query":  { "fuzzy": { "name": "Welington" } },
  "filter": { "term":  {"tags": "beef" } }
}'

{
  "took": 5, "timed_out": false, "_shards": { "total": 5, "successful": 5, "failed": 0 },
  "hits": {
    "total": 1, "max_score": 0.625, "hits": [
      { "_index": "cookbook", "_type": "recipes", "_id": "2", "_score": 0.625,
        "_source": { "name": "Beef Wellington", "tags": ["beef", "steak", "pastry"] } }
    ]
  }
}
```

Mappings

- Associated with types
- Dynamic (schema-less) by default
- Field types
    - Core
    - Objects
    - Arrays
    - Special (IP, geo, files)
- Analyzers
    - Defined at index level, assigned to types and fields
    - Stopwords, n-grams, stemming

Distributed Architecture

- Sharding
- Replication
- Node discovery
- Scatter/gather search
- Request redirection
- Automatic balancing, failover
- Multi-tenant indexes



Additional Features

- Update API
- Routing
- Parents and children
- Timestamps
- TTL

Elastica

https://github.com/ruflin/Elastica/

Elastica is a PHP client for ElasticSearch.

```
$elasticaClient = new Elastica_Client();


$elasticaClient = new Elastica_Client([
    'host' => 'mydomain.org',
    'port' => 12345,
]);


$elasticaClient = new Elastica_Client([
    'servers' => [
        [ 'host' => 'localhost', 'port' => 9200 ],
        [ 'host' => 'localhost', 'port' => 9201 ],
    ]
]);


$elasticaIndex = $elasticaClient->getIndex('cookbook');
$elasticaType = $elasticaIndex->getType('recipes');
```

Searching is straightforward.

```
$elasticaClient = new Elastica_Client();
$elasticaIndex = $elasticaClient->getIndex('cookbook');
$elasticaType = $elasticaIndex->getType('recipes');

$resultSet = $elasticaType->search('pastry');
```

Search results have an object-oriented representation, too.

```
/** @var Elastica_ResultSet */
$resultSet = $elasticaType->search('pastry');

$totalHits = $resultSet->getTotalHits();

foreach ($resultSet->getResults() as $result) {
    /** @var Elastica_Result */
    $result->getScore();
    $result->getExplanation();
    $data = $result->getData();
}
```

Complex searches may be built up from objects.

```
$mltQuery = new Elastica_Query_MoreLikeThis();
$mltQuery->setLikeText('a sample recipe');
$mltQuery->setFields(['name', 'description']);

$existsFilter = new Elastica_Filter_Exists('ingredients');
$notTagFilter = new Elastica_Filter_Not(
    new Elastica_Filter_Term([ 'tags', 'stodgy' ])
);

$andFilter = new Elastica_Filter_And();
$andFilter->addFilter($notTagFilter);
$andFilter->addFilter($existsFilter);

$mltQuery->setFilter($andFilter);
```

Mappings may be set for types.

```
$elasticaType = $elasticaIndex->getType('recipes');

$mapping = new Elastica_Type_Mapping($elasticaType);
$mapping->setProperties([
    'name' => [ 'type' => 'string', 'boost' => 5 ],
    'tags' => [ 'type' => 'string', 'index_name' => 'tag', 'boost' => 3 ],
]);
$mapping->send();

// Alternatively...
$elasticaType->setMapping([
    'name' => [ 'type' => 'string', 'boost' => 5 ],
    'tags' => [ 'type' => 'string', 'index_name' => 'tag', 'boost' => 3 ],
]);
```

It'd be helpful if searches matched more than just complete words.

```
$elasticaIndex->create([
    'number_of_shards' => 4,
    'number_of_replicas' => 2,
    'analysis' => [
        'analyzer' => [
            'indexAnalyzer' => [ 'type' => 'snowball' ],
            'searchAnalyzer' => [ 'type' => 'snowball' ],
        ],
    ],
], true);

// If we search for "baking", we can get results for "baked", "bakes", etc.
```

Analyzers may be applied to types.

```
$mapping = new Elastica_Type_Mapping($elasticaType);
$mapping->setParam('index_analyzer', 'indexAnalyzer');
$mapping->setParam('search_analyzer', 'searchAnalyzer');
$mapping->setProperties(
    // ...
);
$mapping->send();
```

We also want to avoid hits for uninteresting, common words.

```
$elasticaIndex->create([
    'analysis' => [
        'analyzer' => [
            'url_analyzer' => [
                'type' => 'custom',
                'tokenizer' => 'lowercase',
                'filter' => [ 'stop', 'url_stop' ],
            ],
        ],
        'filter' => [
            'url_stop' => [ 'type' => 'stop', 'stopwords' => [ 'http', 'https' ]],
        ],
    ],
], true);
```

Analyzers may also apply to specific fields.

```
$mapping->setProperties([
    'url' => [ 'type' => 'string', 'analyzer' => 'url_analyzer' ],
    // ...
]);

$urlQuery = new \Elastica_Query_Text();
$urlQuery->setFieldQuery('url', 'pastry');
$urlQuery->setFieldParam('url', 'analyzer', 'url_analyzer');
```

We may want to present faceted navigation for a search.

```
$elasticaFacet = new Elastica_Facet_Terms('myFacetName');
$elasticaFacet->setField('tags');
$elasticaFacet->setSize(10);

// Add that facet to the search query object.
$elasticaQuery->addFacet($elasticaFacet);
```

Facet data will be included with query results.

```
// Get facets from the result of the search query
$elasticaFacets = $elasticaResultSet->getFacets();

// Note: "myFacetName" is the name of the facet we defined
foreach ($elasticaFacets['myFacetName']['terms'] as $elasticaFacet) {
    printf("%s: %s\n", $elasticaFacet['term'], $elasticaFacet['count']);
}
```

```
beef: 3
pastry: 2
roast: 1
pie: 1
```

FOQElasticaBundle

https://github.com/Exercise/FOQElasticaBundle

Indexes and types are defined in the bundle's configuration.

```
foq_elastica:
    clients: { default: { host: localhost, port: 9200 } }
    indexes:
        cookbook:
            client: default
            types:
                recipes: ~
```

Mapping Types to DB Entities

```
# foq_elastica / indexes / types
    recipes:
        mappings:
            name: { type: string, boost: 5 }
            tags: { type: string, boost: 3 }
        persistence:
            driver: orm
            model: CookbookBundle\Entity\Recipe
            provider: { query_builder_method: createIsPublishedQueryBuilder }
            listener: { is_indexable_callback: isPublished }
```

Index and Type Services

```
/** @var Elastica_Index */
$cookbookIndex = $this->container->get('foq_elastica.index.cookbook');

/** @var Elastica_ResultSet */
$resultSet = $cookbookIndex->search('pastry');

/** @var Elastica_Type */
$recipesType = $this->container->get('foq_elastica.index.cookbook.recipes');

/** @var Elastica_ResultSet */
$resultSet = $recipesType->search('pastry');
```

Transforming Search Results

```
# foq_elastica / indexes / types
    recipes:
        persistence:
            driver: orm
            model: CookbookBundle\Entity\Recipe
            finder: ~
```

```
/** @var FOQ\ElasticaBundle\Finder\TransformedFinder */
$finder = $container->get('foq_elastica.finder.cookbook.recipes');

/** @var array of CookbookBundle\Entity\Recipe objects */
$recipes = $finder->find('pastry');
```

Results and Entities Together

```
/** @var array of FOQ\ElasticaBundle\HybridResult */
$hybridResults = $finder->findHybrid('pastry');

foreach ($hybridResults as $hybridResult) {
    /** @var CookbookBundle\Entity\Recipe */
    $recipe = $hybridResult->getTransformed();

    /** @var Elastica_Result */
    $elasticaResult = $hybridResult->getResult();
}
```

Console Commands

```
$ php app/console foq:elastica:populate --index cookbook --no-debug

$ php app/console foq:elastica:search --index cookbook --type recipes \
    --query pastry --show-field name
```

Repository Classes

```
# foq_elastica / indexes / types
    recipes:
        persistence:
            driver: orm
            model: CookbookBundle\Entity\Recipe
            repository: CookbookBundle\Search\RecipeRepository
```

Complex queries can be encapsulated in repositories.

```
<?php

use FOQ\ElasticaBundle\Repository;

namespace CookbookBundle\Search;

class RecipeRepository extends Repository
{
    public function findWithCustomQuery($searchText)
    {
        // Build complex $query with Elastica objects
        return $this->find($query);
    }
}
```

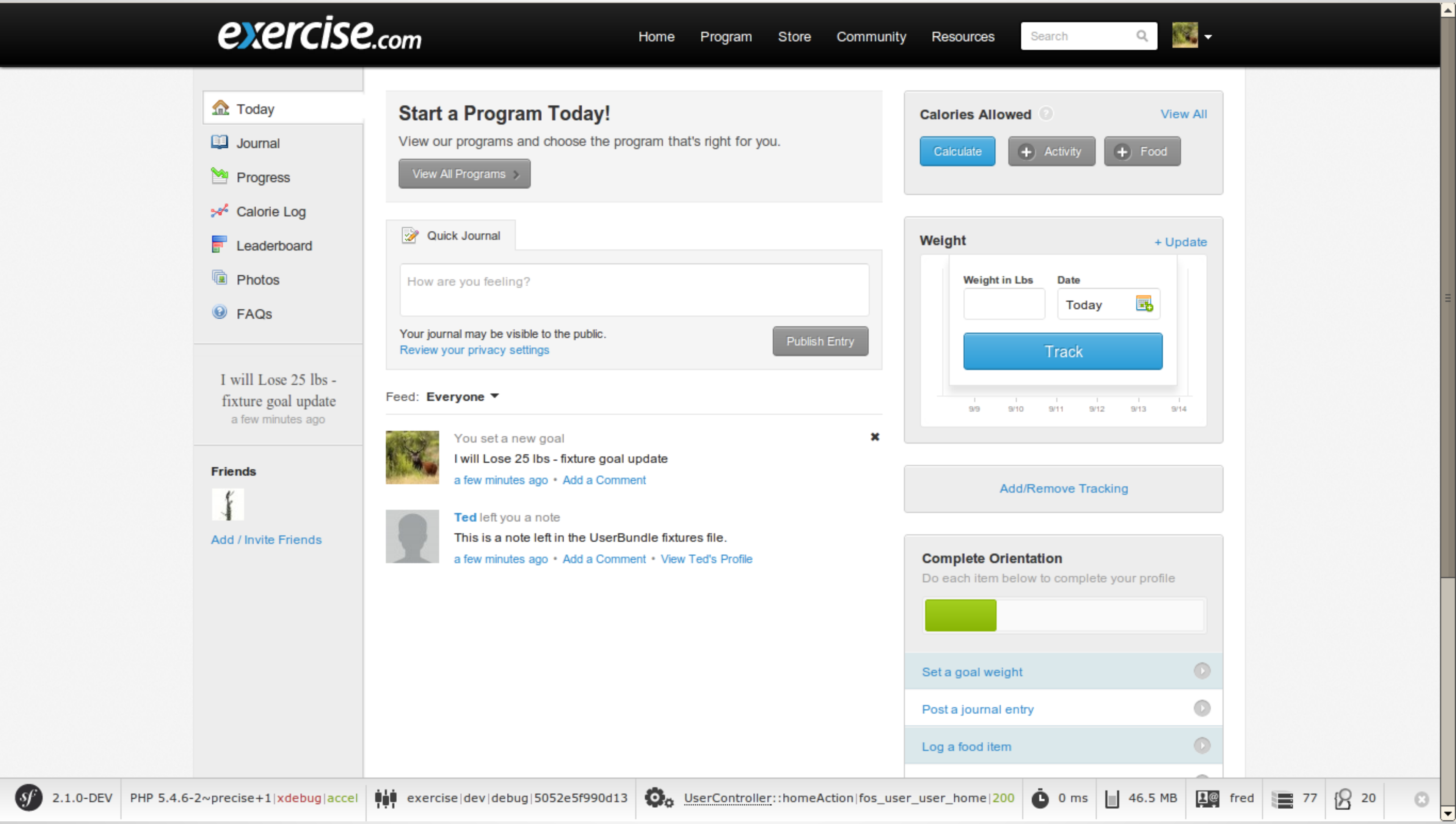
Querying with Repository Services

```
/** @var FOQ\ElasticaBundle\Manager\RepositoryManager */
$repositoryManager = $container->get('foq_elastica.manager');

/** @var FOQ\ElasticaBundle\Repository */
$repository = $repositoryManager->getRepository(CookbookBundle:Recipe');

/** @var array of CookbookBundle\Entity\Recipe */
$recipes = $repository>findWithCustomQuery('pastry');
```

Indexing Files

```
# foq_elastica / indexes / types
    recipes:
        mappings:
            attachedFile: { type: attachment }
```

WDT and Profiler Integration


exercise.com

Home · Program · Store · Community · Resources · Search

Today
Journal
Progress
Calorie Log
Leaderboard
Photos
FAQs

I will Lose 25 lbs - fixture goal update
a few minutes ago

Friends

Add / Invite Friends

Start a Program Today!

View our programs and choose the program that's right for you.

View All Programs

Quick Journal

How are you feeling?

Your journal may be visible to the public.
Review your privacy settings

Publish Entry

Feed: Everyone

You set a new goal
I will Lose 25 lbs - fixture goal update
a few minutes ago • Add a Comment

Ted left you a note
This is a note left in the UserBundle fixtures file.
a few minutes ago • Add a Comment • View Ted's Profile

Calories Allowed — View All

Calculate · Activity · Food

Weight — + Update

Weight in Lbs	Date
	Today

Track

9/9 9/10 9/11 9/12 9/13 9/14

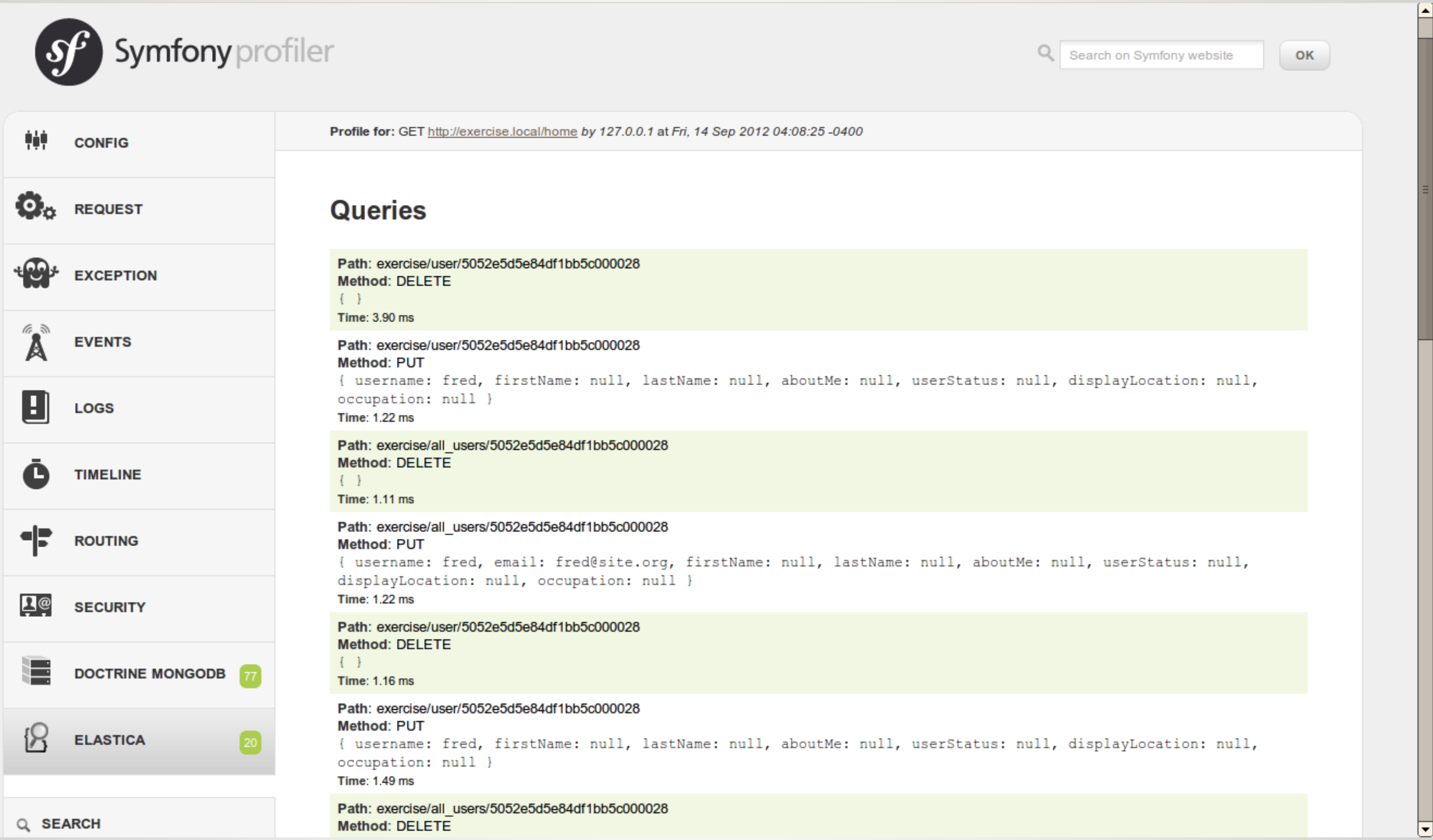
Add/Remove Tracking

Complete Orientation

Do each item below to complete your profile

Set a goal weight
Post a journal entry
Log a food item

```
2.1.0-DEV | PHP 5.4.6-2~precise+1|xdebug|accel | exercise|dev|debug|5052e5f990d13 | UserController::homeAction|fos_user_user_home|200 | 0 ms | 46.5 MB | fred | 77 | 20
```



Profile for: GET http://exercise.local/home *by 127.0.0.1* at *Fri, 14 Sep 2012 04:08:25 -0400*

Queries

Path: exercise/user/5052e5d5e84df1bb5c000028
Method: DELETE
```
{ }
```
Time: 3.90 ms

Path: exercise/user/5052e5d5e84df1bb5c000028
Method: PUT
```
{ username: fred, firstName: null, lastName: null, aboutMe: null, userStatus: null, displayLocation: null,
occupation: null }
```
Time: 1.22 ms

Path: exercise/all_users/5052e5d5e84df1bb5c000028
Method: DELETE
```
{ }
```
Time: 1.11 ms

Path: exercise/all_users/5052e5d5e84df1bb5c000028
Method: PUT
```
{ username: fred, email: fred@site.org, firstName: null, lastName: null, aboutMe: null, userStatus: null,
displayLocation: null, occupation: null }
```
Time: 1.22 ms

Path: exercise/user/5052e5d5e84df1bb5c000028
Method: DELETE
```
{ }
```
Time: 1.16 ms

Path: exercise/user/5052e5d5e84df1bb5c000028
Method: PUT
```
{ username: fred, firstName: null, lastName: null, aboutMe: null, userStatus: null, displayLocation: null,
occupation: null }
```
Time: 1.49 ms

Path: exercise/all_users/5052e5d5e84df1bb5c000028
Method: DELETE

Future Plans

- Annotation-based mapping
  - Methods and/or properties
- Improve ORM/ODM agnosticity
  - Propel support is incomplete
- ElasticSearch ODM?
  - ElasticSearch is a document store
  - Is transforming to DB entities always necessary?

Final Takeaways

- Applications benefit from *real* search
- ElasticSearch is one answer
  - Simple to get up and running
  - Depth of functionality
- FOQElasticaBundle can help
  - Elastica is well-designed
  - Integrates with services and ORM/ODM
- You can improve searching in your app today

Thanks!

Questions?

https://joind.in/7058