Bleve

modern text indexing for Go

GopherCon India - Feb 2015

Say What?

blev-ee

bih-leev

Marty Schoch

- NoSQL Document Database
- Official Go SDK

- Projects Using Go
  - N1QL Query Language
  - Secondary Indexing
  - Cross Data-Center Replication

Why?

elasticsearch.

Apache Solr

Lucene

- Lucene/Solr/Elasticsearch are awesome
- Could we build 50% of Lucene's text analysis, combine it with off-the-shelf KV stores and get something interesting?
  - Pluggable text analysis
  - Pluggable KV storage

What is Search?

Simple Search

Advanced Search

Google

Advanced Search

Find pages with...

all these words:	golang
this exact word or phrase:	
any of these words:	
none of these words:	
numbers ranging from:	to

Search Results

About 4,750 results (0.87 seconds)

Did you mean: ***bleve search***

Spelling Suggestions

blevesearch/bleve · GitHub
https://github.com/**blevesearch**/bleve
A modern text indexing library for go. Contribute to bleve development by creating an account on GitHub.

Result Text Snippets

bleve - modern text indexing for Go
www.**blevesearch**.com/
import "github.com/**blevesearch**/bleve" func main() { // open a new index mapping := bleve.NewIndexMapping() index, [illegible] New("example.bleve" ...

Highlighted Search Terms

bleve (@blevesearch) | Twitter
https://twitter.com/**blevesearch**
The latest Tweets from bleve (@**blevesearch**). modern text indexing for go.

Faceted Search

1-12 of 15 results for **Books** : **"golang"**

Show results for

- Any Category
- **Books**
  - Programming (11)
  - Computer Programming Language & Tool (9)
  - Reference (10)
  - Introductory & Beginning Programming (2)
  - Software Development (2)
  - Software (2)
  - Web Services (1)
  - Internet & Web Culture (2)
  - Software Utilities (1)
  - Linux Operating System (2)
  - Computers & Technology (12)
- + See more

Related Searches: go, go programming, haskell.

Book Format: Kindle Edition | Paperback

An Introduction to Programming in Go Sep 3, 2012
by Caleb Doxsey

Paperback
$10.00 Prime
Get it by **Thursday, Jan 22**

More Buying Choices
$5.78 used & new (6 offers)

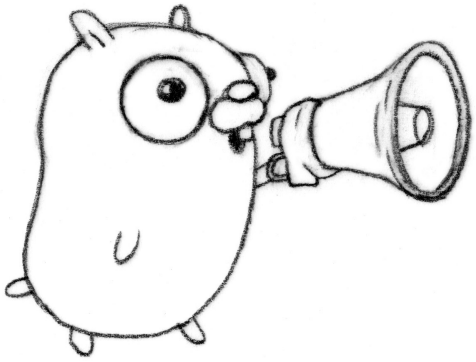
Kindle Edition
$3.00
Auto-delivered wirelessly

Go: Up and Running Apr 25, 2015
by Alan Harris

Paperback

Getting Started

Install Bleve



go get github.com/blevesearch/bleve/...

Import

```
1  import "github.com/blevesearch/bleve"
2
3  type Person struct {
4      Name string
5  }
6
7  func main() {
8      mapping := bleve.NewIndexMapping()
9      index, err := bleve.New("people.bleve", mapping)
10     if err != nil {
11         log.Fatal(err)
12     }
13
14     person := Person{"Marty Schoch"}
15     err = index.Index("m1", person)
16     if err != nil {
17         log.Fatal(err)
18     }
19     fmt.Println("Indexed Document")
20 }
```

Data Model

```go
import "github.com/blevesearch/bleve"

type Person struct {
    Name string
}

func main() {
    mapping := bleve.NewIndexMapping()
    index, err := bleve.New("people.bleve", mapping)
    if err != nil {
        log.Fatal(err)
    }

    person := Person{"Marty Schoch"}
    err = index.Index("m1", person)
    if err != nil {
        log.Fatal(err)
    }
    fmt.Println("Indexed Document")
}
```

Index Mapping

```go
import "github.com/blevesearch/bleve"

type Person struct {
    Name string
}

func main() {
    mapping := bleve.NewIndexMapping()
    index, err := bleve.New("people.bleve", mapping)
    if err != nil {
        log.Fatal(err)
    }

    person := Person{"Marty Schoch"}
    err = index.Index("m1", person)
    if err != nil {
        log.Fatal(err)
    }
    fmt.Println("Indexed Document")
}
```

Create New Index

```
1  import "github.com/blevesearch/bleve"
2
3  type Person struct {
4      Name string
5  }
6
7  func main() {
8      mapping := bleve.NewIndexMapping()
9      index, err := bleve.New("people.bleve", mapping)
10     if err != nil {
11         log.Fatal(err)
12     }
13
14     person := Person{"Marty Schoch"}
15     err = index.Index("m1", person)
16     if err != nil {
17         log.Fatal(err)
18     }
19     fmt.Println("Indexed Document")
20 }
```

Index Data

```
1  import "github.com/blevesearch/bleve"
2
3  type Person struct {
4      Name string
5  }
6
7  func main() {
8      mapping := bleve.NewIndexMapping()
9      index, err := bleve.New("people.bleve", mapping)
10     if err != nil {
11         log.Fatal(err)
12     }
13
14     person := Person{"Marty Schoch"}
15     err = index.Index("m1", person)
16     if err != nil {
17         log.Fatal(err)
18     }
19     fmt.Println("Indexed Document")
20 }
```

Index Data

```go
import "github.com/blevesearch/bleve"

type Person struct {
    Name string
}

func main() {
    mapping := bleve.NewIndexMapping()
    index, err := bleve.New("people.bleve", mapping)
    if err != nil {
        log.Fatal(err)
    }

    person := Person{"Marty Schoch"}
    err = index.Index("m1", person)
    if err != nil {
        log.Fatal(err)
    }
    fmt.Println("Indexed Document")
}
```

```
Indexed Document

Program exited.
```

Open Index

```
1  func main() {
2      index, err := bleve.Open("people.bleve")
3      if err != nil {
4          log.Fatal(err)
5      }
6
7      query := bleve.NewTermQuery("marty")
8      request := bleve.NewSearchRequest(query)
9      result, err := index.Search(request)
10     if err != nil {
11         log.Fatal(err)
12     }
13     fmt.Println(result)
14 }
```

Build Query

```
1  func main() {
2      index, err := bleve.Open("people.bleve")
3      if err != nil {
4          log.Fatal(err)
5      }
6
7      query := bleve.NewTermQuery("marty")
8      request := bleve.NewSearchRequest(query)
9      result, err := index.Search(request)
10     if err != nil {
11         log.Fatal(err)
12     }
13     fmt.Println(result)
14 }
```

Build Request

```
1  func main() {
2      index, err := bleve.Open("people.bleve")
3      if err != nil {
4          log.Fatal(err)
5      }
6
7      query := bleve.NewTermQuery("marty")
8      request := bleve.NewSearchRequest(query)
9      result, err := index.Search(request)
10     if err != nil {
11         log.Fatal(err)
12     }
13     fmt.Println(result)
14 }
```

Search

```
1  func main() {
2      index, err := bleve.Open("people.bleve")
3      if err != nil {
4          log.Fatal(err)
5      }
6
7      query := bleve.NewTermQuery("marty")
8      request := bleve.NewSearchRequest(query)
9      result, err := index.Search(request)
10     if err != nil {
11         log.Fatal(err)
12     }
13     fmt.Println(result)
14 }
```

Search

```go
func main() {
    index, err := bleve.Open("people.bleve")
    if err != nil {
        log.Fatal(err)
    }

    query := bleve.NewTermQuery("marty")
    request := bleve.NewSearchRequest(query)
    result, err := index.Search(request)
    if err != nil {
        log.Fatal(err)
    }
    fmt.Println(result)
}
```

```
1 matches, showing 1 through 1, took 24.713µs
    1. m1 (0.216978)


Program exited.
```

More Realistic Data

Conference Schedule

Friday, 20th February 2015

7.00 AM	Registration
8.15 AM	Opening Keynote - Gopher: From Darkness to Enlightenment by Francesc Campoy Flores
9.00 AM	Herding Gophers - Aaron Cruz
9.25 AM	The Gopher from the Future - Gabriel Aszalos
	Tea Break
10.20 AM	A Journey From Ruby to Go - Mike Gehard
10.45 AM	Go In Action - William Kennedy
11.10 AM	Go for Front End Developers - Julia Poladsky
11.35 AM	Joy of single purpose services in Go - Niket Patel
	Lunch Break
1.00 PM	Go Faster: Optimising Go Programs - Jason Moiron
1.25 PM	Practical Tips for Creating a Go Package Successfully - Keiji Yoshida
1.50 PM	Building RESTful Services With Go and MongoDB - Shiju Varghese
2.15 PM	Web development using Go and Ember.js - Baiju Muthukadan
	Tea Break
3.10 PM	How we test a large-scale Go web app - Beyang Liu
3.35 PM	Concurrency by Data Structures (and nasty examples) - Verónica López
4.00 PM	Principles of designing Go APIs with channels - Alan Shreve
4.25 PM	Cgo - Go under the hood - Rajesh Ramachandran
4.50 PM	Closing Keynote: The Roots of Go - Baishampayan Ghose
5.35 PM	Lightning Talks
	Will be announced later

bleve - modern text indexing for Go

Saturday 21st February, 2015 **10:30 am to 10:55 am (IST)**

Nearly every application today has a search component. But delivering high quality search results requires a long list of text analysis and indexing techniques. With the bleve library, we bring advanced text indexing and search to your Go applications.

This talk will start with a brief introduction to text search concepts. We'll discuss the importance of choosing the right analyzer for your text. Then we'll look at examples that demonstrate how to index your existing data model. Finally, we'll look at how you can integrate advanced features, like result highlighting and faceting, to complete the user experience.

Martin Schoch
@mschoch

Marty Schoch is a software engineer at Couchbase. As a part of the Couchbase Labs team, Marty has built many experimental projects in Go, contributing to its increased adoption inside Couchbase. The most successful of these projects was N1QL, an SQL-like query language for Couchbase Server. He is also a core contributor to the Go Couchbase SDK. Currently, Marty is leading development of bleve, a modern text indexing library for Go.

```go
type Event struct {
    UID         string    `json:"uid"`
    Summary     string    `json:"summary"`
    Description string    `json:"description"`
    Speaker     string    `json:"speaker"`
    Start       time.Time `json:"start"`
    Duration    float64   `json:"duration"`
}
```

Phrase Search

```
1  func main() {
2
3      index, err := bleve.Open("gopherconin.bleve")
4      if err != nil {
5          log.Fatal(err)
6      }
7
8      phrase := []string{"quality", "search", "results"}
9      q := bleve.NewPhraseQuery(phrase, "description")
10     req := bleve.NewSearchRequest(q)
11     req.Highlight = bleve.NewHighlightWithStyle("html")
12     req.Fields = []string{"summary", "speaker"}
13     res, err := index.Search(req)
14     if err != nil {
15         log.Fatal(err)
16     }
17     fmt.Println(res)
18 }
```

Phrase Search

```go
func main() {

    index, err := bleve.Open("gopherconin.bleve")
    if err != nil {
        log.Fatal(err)
    }

    phrase := []string{"quality"
    q := bleve.NewPhraseQuery(ph
    req := bleve.NewSearchReques
    req.Highlight = bleve.NewHig
    req.Fields = []string{"summa
    res, err := index.Search(req
    if err != nil {
        log.Fatal(err)
    }
    fmt.Println(res)
}
```

```
1 matches, showing 1 through 1, took 46.134µs
    1. bleve_-_modern_text_indexing_for_go
(1.033644)
    description
        …earch component. But delivering high
quality search results requires a long list of text
analysis and indexing techniques. With the bleve
library, we bring advanced text indexing and search
to your Go…
    summary
        bleve - modern text indexing for Go
    speaker
        Martin Schoch

Program exited.
```

Query String

```
1  func main() {
2
3      index, err := bleve.Open("gopherconin.bleve")
4      if err != nil {
5          log.Fatal(err)
6      }
7
8      qString := `+description:text `
9      qString += `summary:"text indexing" `
10     qString += `summary:believe~2 `
11     qString += `-description:lucene `
12     qString += `duration:<30`
13     q := bleve.NewQueryStringQuery(qString)
14     req := bleve.NewSearchRequest(q)
15     req.Highlight = bleve.NewHighlightWithStyle("html")
16     req.Fields = []string{"summary", "speaker", "description", "duration"}
17     res, err := index.Search(req)
18     if err != nil {
19         log.Fatal(err)
20     }
21     fmt.Println(res)
22 }
```

Query String

```
1  func main() {
2
3      index, err := bleve.Open("gopherconin.bleve")
4      if err != nil {
5          log.Fatal(err)
6      }
7
8      qString := `+description
9      qString += `summary:"tex
10     qString += `summary:beli
11     qString += `-description
12     qString += `duration:<30
13     q := bleve.NewQueryStrin
14     req := bleve.NewSearchRe
15     req.Highlight = bleve.Ne
16     req.Fields = []string{"s
17     res, err := index.Search
18     if err != nil {
19         log.Fatal(err)
20     }
21     fmt.Println(res)
22 }
```

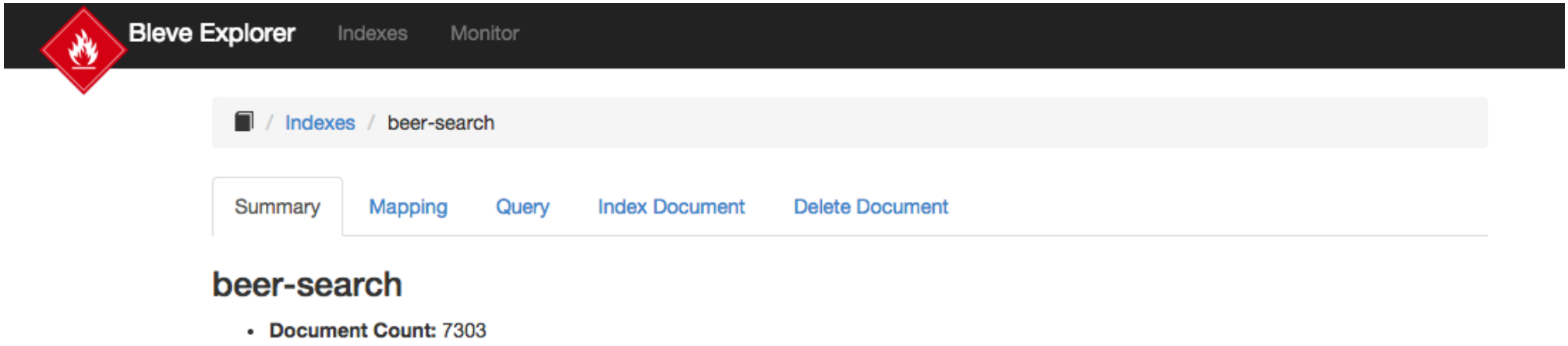
```
1 matches, showing 1 through 1, took 220.08µs
    1. bleve_-_modern_text_indexing_for_go (0.338882)
    summary
        bleve - modern text indexing for Go
    description
        …ist of text analysis and indexing techniques.
With the bleve library, we bring advanced text indexing
and search to your Go applications. This talk will start
with a brief introduction to text search …
    speaker
        Martin Schoch
    duration
        25

Program exited.
```

HTTP Handlers (optional)

```
import "github.com/blevesearch/bleve/http"
```

- All major bleve operations mapped
- Assume JSON document bodies
- See bleve-explorer sample app

```
https://github.com/blevesearch/bleve-explorer
```



Putting It All Together

Schedule Search

Unofficial GopherCon India Schedule Search

Search

Learn how to build search apps like this with bleve in my GopherCon India talk.

http://gopherconin.blevesearch.com/

Unofficial GopherCon India Schedule Search

go

28 results (8ms)

Building RESTful Services With Go and MongoDB

0.281

Shiju Varghese on Friday at 13:50 (25 min)

...o build REST based web services in Go with MongoDB as a persistence storage. This session will also leverage Go package mgo for working with MongoDB from Go apps. Audience : Intermediate to Advanced d...

A Journey From Ruby to Go

0.275

Mike Gehard on Friday at 10:20 (25 min)

...ake a look at Go and wonder if they should learn it. Many dabble in Go and then go back to the warm embrace of Ruby while others dive head first into the Go community and never look back. This talk le...

Practical Tips for Creating a Go Package Successfully

0.271

Keiji Yoshida on Friday at 13:25 (25 min)

...eating a Go package successfully. I have created and maintained some Go packages such as a CSS preprocessor and an HTML template engine. I have learned and found useful manners to create a Go package ...

Go for Front End Developers

0.269

Julia Poladsky on Friday at 11:10 (25 min)

Go is perceived as a systems programming language for developers who develop complex backend software and infrastructure. In her talk, she will show how front end developers don't need to shy away fro...

Joy of single purpose services in Go

0.254

Niket Patel on Friday at 11:35 (25 min)

...experience of integrating Go in his company's tech stack. They were primarily a Ruby shop; they have a large application written in Ruby which had some problems in scaling (only in some areas). Then t...

Go Faster: Optimising Go Programs

0.251

Jason Moiron on Friday at 13:00 (25 min)

... the past year of using Go in production. It will cover things like use of the Go profiler, the go test benchmarks, runtime behavior, cgo, compiler optimisation/inlining, go's asm support, and some im...

Principles of designing Go APIs with channels

0.228

Alan Shreve on Friday at 16:00 (25 min)

...te. Most Go resources, however, have little to say about how to design APIs that use channels. This talk will discuss the principles both how and when it is appropriate to use channels in Go APIs draw...

Refine Results

Day

- [ ] Friday (15)
- [ ] Saturday (13)

Duration

- [ ] <=30 min (25)
- [ ] 30-60 min (3)

Unofficial GopherCon India Schedule Search

go

13 results (5ms)

EMBD

0.005

Kunal Powar on Saturday at 8:45 (25 min)

...amework completely written in Go. It'll showcase various aspects and patterns of Go used in the framework and its applications, thereby justifying the choice of using Go. The talk will also be followe...

Opening Keynote - Embracing the Standard Library

0.005

Bryan Liles on Saturday at 8:00 (45 min)

...w Go developers follow a typical pattern: Find Go. Start learning Go. Pull in a bunch of external dependencies into their projects. While this pattern allows good software to be created, it often negl...

How to keep wall street chatting using Go

0.005

Matthew Campbell on Saturday at 14:45 (25 min)

... instant messaging platforms using GO. The service hosted more then 350k users with peaks of 20 megabits of dynamic messaging traffic. The service was switched to go mid day and has been running for m...

Closing Keynote - Simplicity and the ideas that Go left behind

0.004

Dave Cheney on Saturday at 16:30 (45 min)

A discussion on ideas which Go's designers decided to leave _out_ of the language and the effect of those decisions on a language designed for simplicity and scale.

Raytracing in Go

0.004

Guillaume J. Charmes on Saturday at 16:05 (25 min)

... and why Go is a great language for implementing these types of resources demanding algorithms. I will show how the language syntax and features and the built-in concurrency support makes Go a perfect...

Image Processing in Scale with Go

0.004

Jyotiska NK on Saturday at 15:40 (25 min)

... and we have been successfully able to use port our image processing infrastructure from Python to Go. This talk will explore our journey of how using the OpenCV bindings, native image package and few...

Concurrent, High-Performance Data-Access With Go

0.003

Khosrow Afroozeh on Saturday at 9:10 (25 min)

... fore-runners of integrating a more involved part of the data access library in its Smart Clients. Go, by providing a non-blocking runtime and native concurrency features in the language, has proved a...

Refine Results

Day

- [x] Saturday (13)

Duration

- [ ] <=30 min (11)
- [ ] 30-60 min (2)

Unofficial GopherCon India Schedule Search

go

2 results (2ms)

Opening Keynote - Embracing the Standard Library

0.004

Bryan Liles on Saturday at 8:00 (45 min)

...w Go developers follow a typical pattern: Find Go. Start learning Go. Pull in a bunch of external dependencies into their projects. While this pattern allows good software to be created, it often negl...

Closing Keynote - Simplicity and the ideas that Go left behind

0.004

Dave Cheney on Saturday at 16:30 (45 min)

A discussion on ideas which Go's designers decided to leave _out_ of the language and the effect of those decisions on a language designed for simplicity and scale.

Refine Results

Day

- [x] Saturday (2)

Duration

- [x] 30-60 min (2)

Learn how to build search apps like this with bleve in my GopherCon India talk.

Join the Community

Community

freenode

- #bleve is small/quiet room, talk to us real time

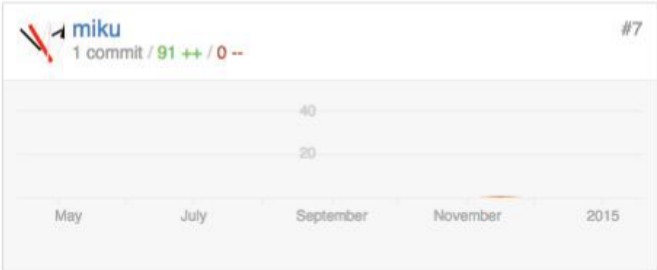
Google groups

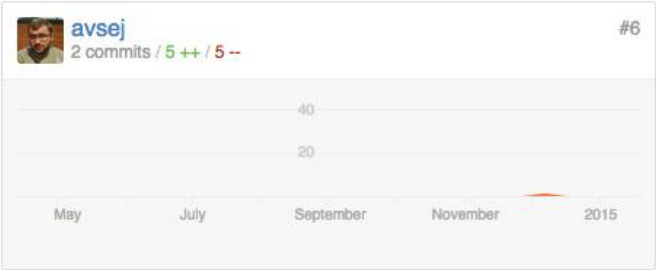
- Discuss your use-case
- Plan a feature implementation

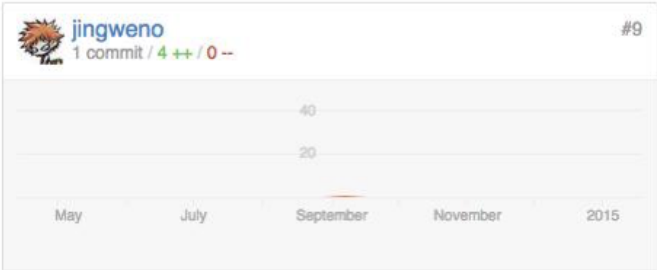
GitHub

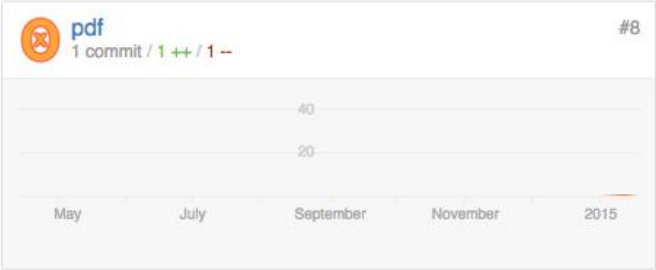
- Apache License v2.0
- Report Issues, Submit Pull Requests

Contributors



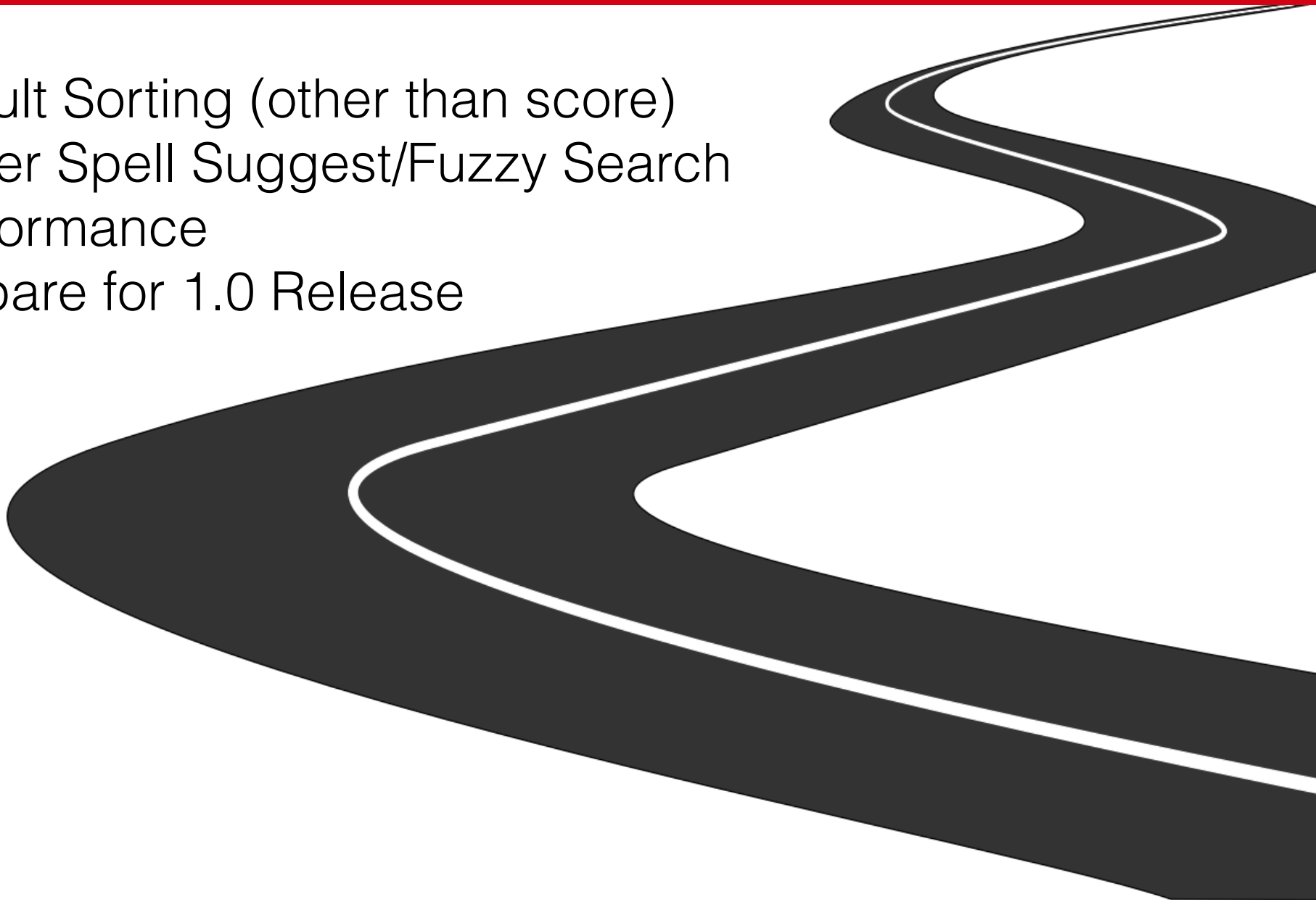






Roadmap

- Result Sorting (other than score)
- Better Spell Suggest/Fuzzy Search
- Performance
- Prepare for 1.0 Release



One More Thing…

Hindi Analyzer

हिन्दी विकिपीडिया

कंप्यूटर

1066 results (2.969s)

भारत में कंप्यूटर युग की शुरुआत

1.847

Sanjeev bot modified 5 months ago

...कंप्यूटर]] युग की शुरुआत सन १९५२ में [[भारतीय सांख्यिकी संस्थान]] [[कोलकाता]] से हुई थी। सन १९५२ में आई एस आई में एक [[एनालॉग कंप्यूटर]] की स्थापना की गई थी जो भारत का प्रथम कंप्यूटर था। यह कंप्यूटर ...

कंप्यूटर ग्राफिक्स

1.593

modified 6 months ago

...ठ।]] '''कंप्यूटर ग्राफिक्स''' [[कंप्यूटर]] के सहयोग से सृजित ग्राफिक्स यानि रेखाचित्र होते हैं। इनमें अधिकांशतः चित्र डाटा का कंप्यूटर के द्वारा प्रदर्शन और परिवर्तन किया गया होता है। कंप्यूटर ग्राफ़ि...

डाइनेमिक होस्ट कान्फिग्रेष्ण प्रोटोकॉल

1.543

Orbot1 modified 2 years ago

...डाइनेमिक रूप से काम करता है। इसे मुख्य रूप से कंप्यूटर में स्वचालित IP देने के लिए पर्योग में लाया जाता है। ये कंप्यूटर को जभी IP देगे जब आपका कंप्यूटर इस से किसी माध्यम से जुरा हो. ये मुख्यता एसे जग...

Refine Results

Categories

- ☐ गूगल परियोजना (158)
- ☐ संगणक (81)
- ☐ श्रेष्ठ लेख योग्य लेख (71)
- ☐ हिन्दी विकि डीवीडी परियोजना (46)
- ☐ जीवित लोग (41)

Last Modified

- ☐ This Year (925)
- ☐ > Year (141)

Thanks

- Marty Schoch
- marty@couchbase.com
- http://github.com/blevesearch/bleve
- @mschoch
- @blevesearch