Unit Testing in Go

@joewass

May 29, 2013

Who am I?

My name is Joe Wass.
By day I write Python.
This was a 5-minute ‘microslot’ at Oxford Geek Night last November.
Experiences $\approx$ 1 year ago.

folktunefinder

FolkTuneFinder is a search engine for folk tunes. If you're looking for a specific tune and you know the title or some of the melody, use the forms below.

Tweet 17

Like 220

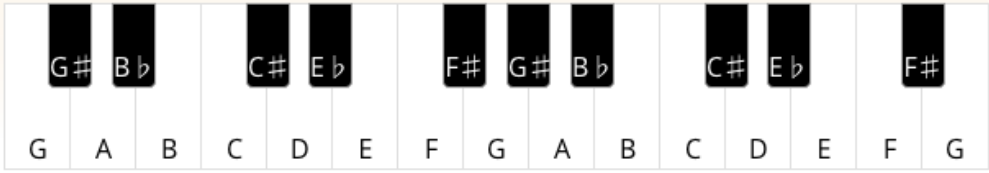
Title Search

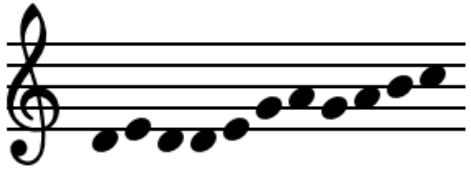
Search

Type in the title of a tune

Melody Search

If you're looking for a tune and you know the melody, type in some notes from the start, at least 5.





delete

Search

folktunefinder

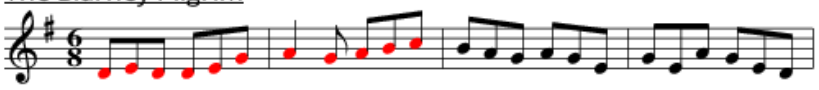
Your search results

Found **49** tunes.

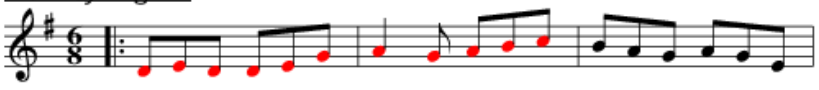
Please note that as of 10th July, these results are produced by a new, experimental algorithm. If you have problems of queries, email joe@folktunefinder.com

If you're confused about the results, read this blog entry on search results.

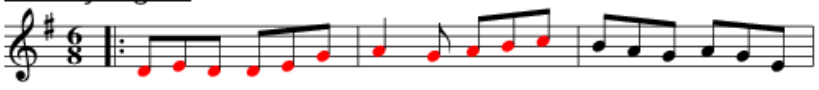
The Blarney Pilgrim



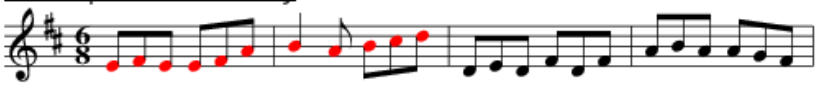
Blarney Pilgrim



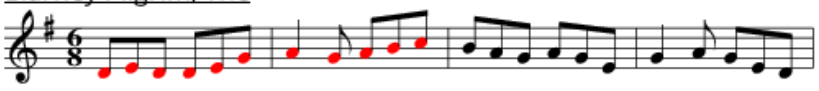
Blarney Pilgrim



Miss Elphinston's Fancy



Blarney Pilgrim, The



folktunefinder

- Melody search engine
- $\approx$ 200,000 tunes
- Front-end written in Django
- Back end written in ...

Go

/search/melody/?melody=62,64,62,62,64,67,69,67,69,71,72


```
{"Results":
[{"DocumentId":140033,"Weight":1,"Starts":0,"Ends":10,"Length":10}
,{"DocumentId":99793,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":134723,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":129001,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":66627,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":113602,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":140026,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":103364,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":140018,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":44703,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":56525,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":128913,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":56526,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":103279,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":140022,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":53663,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":176046,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":50481,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":74291,"Weight":1,"Starts":0,"Ends":10,"Length":10},
{"DocumentId":118021,"Weight":2,"Starts":1,"Ends":10,"Length":9}],
"NextPage":2,"PrevPage":-1,"TotalFound":49,"TrimAmount":0}
```

MIDI Parser Library

`github.com/afandian/go-midi`
Not perfect but solves my problem.
Much easier to test than I expected.

Writing Tests

included in your package, `*_test.go`

each test is a function `Test*`.

```
func TestParse16Bit(t *testing.T)
```

fail a test

```
t.Fatal("Got ", header, " expected MTrk")
```

Running Tests

run tests with go test

```
$ go test
PASS
ok github.com/afandian/go-midi 0.010s
```

The End

Interfaces

```
type MidiLexerCallback interface {
    ...
    NoteOn(channel uint8, pitch uint8, velocity uint8, time
        uint32)
    ...
}
```

```
func NewMidiLexer(input io.ReadSeeker, callback
    MidiLexerCallback) *MidiLexer { ... }
```

```
type CountingLexerCallback struct {
    noteOff int
    ...
}
```

```
func (cbk *CountingLexerCallback) NoteOff(channel uint8,
    pitch uint8, velocity uint8, time uint32) {
    cbk.noteOff++
    cbk.pitch = pitch
    ...
}
```

```
type MockReadSeeker struct {
    data *[]byte
    position int64
}

func (reader *MockReadSeeker) Read(p []byte) (n int, err
    error) {
    ...
    copy(p, (*reader.data)[reader.position:reader.position+
        amount])
    reader.position += amount
    return int(amount), nil
}
```

No Nice Asserts

Not as feature-rich as Django / JUnit / MBUnit / OCUnit.

You have to write your own asserts.

Several times.

No Overloading

```
func assertUint16sEqual(a uint16, b uint16, t *testing.T) {
    if a != b {
        t.Fatal(a, " != ", b)
    }
}
```

```
func assertInt16sEqual(a int16, b int16, t *testing.T) {
    if a != b {
        t.Fatal(a, " != ", b)
    }
}
```

Tools

Code coverage
Run tests by regular expression
Parallell runs
Memory and CPU profiling
Timeouts

The End
(really)

@joewass
blog.afandian.com
joe@afandian.com