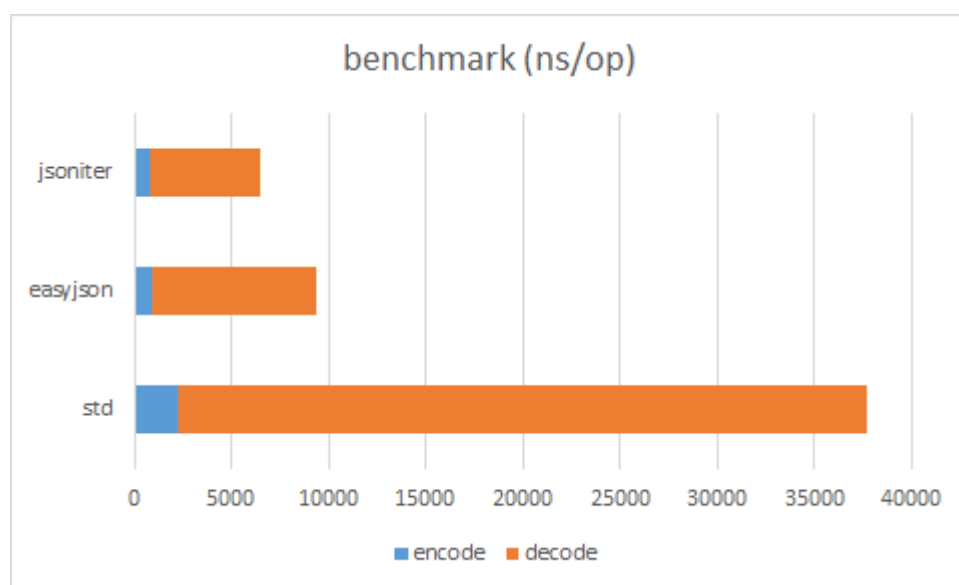


* used by 7.5k projects go documentation build failing codecov 86% go report A+

license mit chat on gitter

A high-performance 100% compatible drop-in replacement of "encoding/json"

Benchmark



Source code: https://github.com/json-iterator/go-benchmark/blob/master/src/github.com/json-iterator/go-benchmark/benchmark_medium_payload_test.go

Raw Result (easyjson requires static code generation)

	ns/op	allocation bytes	allocation times
std decode	35510 ns/op	1960 B/op	99 allocs/op
easyjson decode	8499 ns/op	160 B/op	4 allocs/op
jsoniter decode	5623 ns/op	160 B/op	3 allocs/op
std encode	2213 ns/op	712 B/op	5 allocs/op
easyjson encode	883 ns/op	576 B/op	3 allocs/op
jsoniter encode	837 ns/op	384 B/op	4 allocs/op

Always benchmark with your own workload. The result depends heavily on the data input.

Usage

100% compatibility with standard lib

Replace

```
import "encoding/json"
json.Marshal(&data)
```

with

```
import jsoniter "github.com/json-iterator/go"

var json = jsoniter.ConfigCompatibleWithStandardLibrary
json.Marshal(&data)
```

Replace

```
import "encoding/json"
json.Unmarshal(input, &data)
```

with

```
import jsoniter "github.com/json-iterator/go"

var json = jsoniter.ConfigCompatibleWithStandardLibrary
json.Unmarshal(input, &data)
```

[More documentation](#)

How to get

```
go get github.com/json-iterator/go
```

Contribution Welcomed !

Contributors

- [thockin](#)
- [matttn](#)
- [cch123](#)
- [Oleg Shaldybin](#)
- [Jason Toffaletti](#)

chat on gitter

Report issue or pull request, or email taowen@gmail.com, or