


```
1  #![feature(test)]
2  #[macro_use]
3  extern crate abomonation;
4  extern crate test;
5
6  use test::Bencher;
7  use abomonation::{Abomonation, encode, decode};
8
9  #[bench]
10 fn bench_populate(b: &mut Bencher) {
11     b.iter(|| {
12         Log::new()
13     });
14 }
15
16 #[bench]
17 fn bench_serialize(b: &mut Bencher) {
18     let log = Log::new();
19     let mut bytes = vec![];
20     unsafe { encode(&log, &mut bytes); }
21     b.bytes = bytes.len() as u64;
22     b.iter(|| {
23         bytes.clear();
24         unsafe { encode(&log, &mut bytes); }
25         test::black_box(&bytes);
26     });
27 }
28
29 #[bench]
30 fn bench_deserialize(b: &mut Bencher) {
31     let log = Log::new();
32     let mut bytes = vec![];
33     unsafe { encode(&log, &mut bytes); }
34     b.bytes = bytes.len() as u64;
35     b.iter(|| {
36         test::black_box(unsafe { decode::<Log>(&mut bytes) });
37     });
38 }
```


bench_populate:	267	ns/iter	(+/- 124)	
bench_serialize:	54	ns/iter	(+/- 4)	= 10148 MB/s
bench_deserialize:	8	ns/iter	(+/- 1)	= 68500 MB/s
bench_deserialize_test:	112	ns/iter	(+/- 25)	= 4892 MB/s


```

impl Log {
  pub fn new() -> Log {
    Log {
      timestamp: 2837513946597,
      zone_id: 123456,
      zone_plan: ZonePlan::FREE,
      http: Http {
        protocol: HttpProtocol::HTTP11,
        status: 200,
        host_status: 503,
        up_status: 520,
        method: HttpMethod::GET,
        content_type: "text/html".to_owned(),
        user_agent: "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36".to_owned(),
        referer: "https://www.cloudflare.com/".to_owned(),
        request_uri: "/cdn-cgi/trace".to_owned(),
      },
      origin: Origin {
        ip: "1.2.3.4".to_owned(),
        port: 8000,
        hostname: "www.example.com".to_owned(),
        protocol: OriginProtocol::HTTPS,
      },
      country: Country::US,
      cache_status: CacheStatus::Hit,
      server_ip: "192.168.1.1".to_owned(),
      server_name: "metal.cloudflare.com".to_owned(),
      remote_ip: "10.1.2.3".to_owned(),
      bytes_dlv: 123456,
      ray_id: "10c73629cce30078-LAX".to_owned(),
    }
  }
}

```

```
unsafe_abomodate!(Log : http, origin, server_ip, server_name, remote_ip, ray_id);  
unsafe_abomodate!(Http : content_type, user_agent, referer, request_uri);  
unsafe_abomodate!(Origin : ip, hostname);
```



```
1  #![feature(test)]
2  #[macro_use]
3  extern crate abomonation;
4  extern crate test;
5
6  use test::Bencher;
7  use abomonation::{Abomonation, encode, decode};
8
9  #[bench]
10 fn bench_populate(b: &mut Bencher) {
11     b.iter(|| {
12         Log::new()
13     });
14 }
```

bench_populate:

267 ns/iter (+/- 124)

```
1  #![feature(test)]
2  #[macro_use]
3  extern crate abomination;
4  extern crate test;
5
6  use test::Bencher;
7  use abomination::{Abomination, encode, decode};
8
9  #[bench]
10 fn bench_populate(b: &mut Bencher) {
11     b.iter(|| {
12         Log::new()
13     });
14 }
```