

3-5Extractor

2020年4月6日 12:09

actix-web提供了方便的类型安全的请求信息的访问,称为extractor.

也就是实现了FromRequest的类型.

handler函数的参数只要是实现了FromRequest就是一个extractor,可以多达10个,参数的位置无关紧要.

Path:

是为了处理类型的route: /users/{userid}/{friend}

HttpRequest:的参数也是可以查询路径参数的,通过使用

`req.match_info().get("friend").unwrap()`或者query路径参数进行查询.

Query:类型是来处理请求具体参数的(这里并不是路径参数):

使用Json的结构体,或者使用HashMap,只要是包含名字都是可以的.

Json:可以用在解析请求体是Json格式的API中.

Form:是用来处理表格的.

Data:application之间的Data数据共享:

```
async fn main() -> std::io::Result<()> {
    use actix_web::{App, HttpServer};
    let data = AppState {
        count: Cell::new(0),
    };
    HttpServer::new(move || {
        App::new()
            .data(data.clone())
            .route("/", web::to(show_count))
            .route("/add", web::to(add_one))
    })
```

```
    })  
    .bind("127.0.0.1:8088")?  
    .run()  
    .await  
}
```

像这样的代码:data数据在多个线程和多个请求处理上是不同的,因为书副本.

如果我们实现多个线程共享这个data的话,一个是使用app_data,一个是使用Arc和AtomicUsize.