



链滴

# 写框架的时候遇到的一个问题

作者: [Ever](#)

原文链接: <https://ld246.com/article/1548158210341>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

目前我在写一个简单的mvc框架，然后因为想实现的好玩一点，就参考了biezhi大哥的blade，注册路由的操作可以是这样的

```
public class ExampleApplication {
    public static void main(String[] args) {
        Reverie.up()
            .get("/get", ctx -> { ctx.response().end("你好"); })
            .start(ExampleApplication.class, args);
    }
}
```

因为用的Vertx，所以注册路由的操作比较简单：

```
public void get(String path, RouteHandler routeHandler) {
    Vertx vertx = Vertx.vertx();
    Router route = Router.router(vertx);
    //这一步就是注册路由的操作，routeHandler就是上面启动函数中的ctx
    route.route(HttpMethod.GET, path).handler(routeHandler);
    vertx.createHttpServer().requestHandler(route::accept).listen(8080);
}
```

目前我想做的就是，这个注册路由时的handler方法，传入的参数是Handler<RoutingContext>，一个RoutingContext就是启动函数中的ctx，我想自己重新定义一个RouteContext，定义我自己的api比如，如果使用RoutingContext，那么在main函数中返回一串文本就要这样写：

```
//RoutingContext
get("/get", ctx -> { ctx.response().end("你好"); })
//RouteContext，我自己定义的api为
get("/get", ctx -> { ctx.text("你好"); })
```

这是RouteHandler接口

```
@FunctionalInterface
public interface RouteHandler extends Handler<RoutingContext> {
    void handle(RoutingContext routingContext);
}
```

如果修改为：

```
@FunctionalInterface
public interface RouteHandler extends Handler<RouteContext> {
    void handle(RouteContext routingContext);
}
```

但是在注册路由处理器的时候，出现：

```
public Reverie get(String path, RouteHandler routeHandler) {
    Vertx vertx = Vertx.vertx();
    Router route = Router.router(vertx);
    route.route(HttpMethod.GET, path).handler(routeHandler);
    vertx.createHttpServer().requestHandler(route::accept).listen(8080);
    this.router.get(path).handler(routeHandler);
    return this;
}
```

handler (io.vertx.core.Handler<io.vertx.ext.web.RoutingContext>) in Route cannot be applied to (com.reverie.mvc.RouteHandler)

类型不匹配这种，我尝试过继承或者实现RoutingContext，都没有效果，

handle方法的入参要去为Handler<RoutingContext>类型的，试过不同办法，不知道该咋办了。