# 项目相关测试

# 通信框架

CoreFactory作用：

public class CoreFactory {  
  
 */\*\*  
 \* 当调用getCore(IPraiseCore.class)方法时，它会判断key对应的value是否为空，如果为空实例化对应的类  
 \* 这里就会调用对应类的构造函数PraiseCoreImpl.class，从而触发CoreManager.addClient(this) 方法  
 \*/* public static final HashMap<Class<? extends IBaseCore>, IBaseCore> *cores*;  
  
 */\*\*  
 \* 这里仅保存注册相关的信息，例如 CoreFactory.registerCoreClass(IPraiseCore.class, PraiseCoreImpl.class);  
 \* IPraiseCore.class -> PraiseCoreImpl.class  
 \*/* public static final HashMap<Class<? extends IBaseCore>, Class<? extends AbstractBaseCore>> *coreClasses*;

}

这里面仅仅是保存注册的类，并且外部调用getCore的时候，判断保存的类的实例是否存在，如果不存在就创建一个实例，并保存到cores这个map中。

CoreManager里面

public class CoreManager {  
  
 public static final String *TAG* = "CoreManager";  
 public static final String *TAG\_EVENT* = "CoreManager\_Event";  
  
 */\*\*  
 \*  
 \*/* private static Map<Class<? extends ICoreClient>, CopyOnWriteArraySet<ICoreClient>> *clients* = new HashMap<>();  
  
 private static Map<Class<? extends ICoreClient>, Map<String, Method>> *clientMethods* = new HashMap();  
  
 private static Map<Class<?>, CopyOnWriteArraySet<Object>> *coreEvents* = new HashMap<>();  
  
 private static Map<Object, Map<String, Method>> *coreEventMethods* = new HashMap();

}

clients和clientMethods这两个Map应该是没有用到的，因为IcoreClient并没有对应的实现类，是有接口继承自该类，例如

public interface IPraiseClient extends ICoreClient {  
  
 public static final String *METHOD\_ON\_PRAISE* = "onPraise";  
  
 void onPraise(long uid);  
  
}

这里面真正用到的是有METHOD\_ON\_PRAISE，这是在IBaseCore的实现类PraiseCoreImpl使用的：

public class PraiseCoreImpl extends AbstractBaseCore implements IPraiseCore {  
  
 private LongSparseArray<Boolean> mPraisePlayer = new LongSparseArray<>();  
  
 public PraiseCoreImpl() {  
 CoreManager.*addClient*(this);  
 }  
  
 @Override  
 public void praise(final int likedUid) {  
 // handle something  
 *log*("praise handle something");  
  
 notifyClients(IPraiseClient.class, IPraiseClient.*METHOD\_ON\_PRAISE*, likedUid);  
 }  
  
}

就上面notifyClients的时候用了，还有就是添加注解@CoreEvent的时候用了。其实这个常量完全可以定义在IBaseCore的子类IPraiseCore里面。

IBaseCore是真正的业务接口，继承自该 接口的类必须定义业务功能接口，例如

public interface IPraiseCore extends IBaseCore {  
  
 */\*\*  
 \* uid：点赞人uid，必填  
 \* likedUId：被点赞人uid，必填  
 \* type:喜欢操作类型，1是喜欢，2是取消喜欢，必填  
 \* ticket：必填  
 \* 如果双方为相互喜欢，业务方像网易云发送直接加好友请求，双方直接成为好友。  
 \* 详见：  
 \*/* void praise(int likedUid);  
  
  
}

IBaseCore和实现类必须注册到CoreFactory的coreClasses映射中，在之后调用getCore的时候会实例化具体的类。

式样

微软雅黑 Light 小四

段落 固定值 23磅

# 目录1

## 目录2

### 目录3

#### 目录4

##### 目录5