## 目的

在EOS里，分了多个plugin，执行nodeos时，可以选择加载哪些。

有4个默认一定会加载的plugin: chain\_plugin, http\_plugin, net\_plugin, producer\_plugin

各个plugin之间是没什么关系的，但是业务上又是需要合作的，比如producer\_plugin生成一个块，需要让chain\_plugin发出去，咋办呢？

EOS设计了一种方法，能让各个plugin之间进行松散的连接/通信/调用。同时规定了哪些接口可以跨plugin调用。这些接口定义在 plugin\_interface.hpp。

里面有channel和method两种类型。

## Channel

Channel就是通知者模式，可以通过channel.hpp查看其定义，主要有：

* Handler类，即事件发生时的回调
* Subscribe，将handler注册给channel
* Publish，当事件发生时发送通知，可以带数据进去

## Method

就是函数指针，可以通过method.hpp查看其定义，主要有：

* Handler，一个available的函数provider
* register\_provider，将一个函数provider注册为handler，可以提供优先级。

因为同一个函数指针可能有多个available的函数，所以在选择用哪个函数时，提供了两种policy，first\_success是指一个个provider调用过去，直到有一个不出错。First\_provider是指直接使用第一个provider。另外在register\_provider函数里有提供优先级，可以决定各个provider被调用的次序。

## 各个plugin之间如何关联

关联是通过application.hpp实现的。App在各个plugin之间是公用的，它提供了get\_method和get\_channel两个接口。这两个接口在被调用时，会查询传入的key是否已有，如果没有，就emplace一个，如果有了，就取出来返回。

所以无论调用者，还是provider，都可以调用get\_method/get\_channel取得对象，然后对其调用或执行注册，这样就完成了松散关联。