# Chain\_plugin

## 主要结构

Namespace chainApi{

Class read\_only;

Class read\_write;

}

Class chain\_plugin.

## Read\_only

主要实现nodeos对外的一些可读信息接口，包括各种结构定义和接口。如：

Get\_info

Get\_account

Get\_code

Get\_abi

Abi\_json\_to\_bin

Abi\_bin\_to\_json

Get\_required\_keys

Get\_transaction\_id

Get\_block

Get\_table\_rows

Get\_producers

主要数据：

**const controller& db;**

## read\_write

写链的接口，包括结构定义和接口：

Push\_block

Push\_transaction

Push\_transactions

主要数据：

**controller& db;**

## chain\_plugin

继承自application的子类plugin

接口包括实现plugin定义的通用接口：

void plugin\_initialize(const variables\_map& options);

void plugin\_startup();

void plugin\_shutdown();

set\_program\_options

和一些自己的：

get\_read\_only\_api

get\_read\_write\_api

accept\_block

accept\_transaction

bool block\_is\_on\_preferred\_chain(const chain::block\_id\_type& block\_id);

recover\_reversible\_blocks

import\_reversible\_blocks

export\_reversible\_blocks

get\_chain\_id

该类的实体是chain\_plugin\_impl，也是个class，主要的数据结构：

bfs::path blocks\_dir;

bool readonly = false;

flat\_map<uint32\_t,block\_id\_type> loaded\_checkpoints;

fc::optional<fork\_database> fork\_db;

fc::optional<block\_log> block\_logger;

fc::optional<controller::config> chain\_config;

**fc::optional<controller> chain;**

fc::optional<chain\_id\_type> chain\_id;

//txn\_msg\_rate\_limits rate\_limits;

**fc::optional<vm\_type> wasm\_runtime;**

fc::microseconds abi\_serializer\_max\_time\_ms;

fc::optional<bfs::path> snapshot\_path;

以及一堆的channel，methods，provider\_handlers,connections。初始化时就会通过app->get\_channel和app->get\_method把channel和method初始化好。

#### plugin\_initialize

设定chain\_config，初始值是从controller读的，然后把调用plugin\_initialize时传入的参数加载进来，如黑白名单，trusted\_bp。

如果输入参数有check\_point，则读出，加载在loaded\_checkpoints

读出输入参数中的其他参数，初始化自己的本地数据。包括wasm-runtime。

如果输入参数有export-reversible-blocks，会把reversible-blocks export出来放在blocks\_dir/ reversible下。

如果输入参数有delete-all-blocks，会把state\_dir和blocks\_dir下的内容删掉

如果输入参数有hard-replay-blockchain，会执行如下操作：

清空state\_dir

调用block\_log->repair\_log，把区块一个个重新过滤一遍，看是否有错误的。这一过程可以通过在输入参数中加“truncate-at-block”指定查到哪个block为止。

如果backup\_dir里有reversible文件夹，或者输入参数有fix-reversible-blocks，则recover\_reversible\_blocks，会建立new\_reversible数据库并将数据读入。

通过option字段判断是read-mode,还是validation-mode

初始化一堆provider，connection

**my->chain.emplace( \*my->chain\_config );**

my->chain\_id.emplace( my->chain->get\_chain\_id());

my->chain->add\_indices();

其中my->chain.emplace( \*my->chain\_config );需要特别注意。

my->chain就是controller，是这里的核心数据，这里会将其初始化后加到chain\_plugin，为了不显得支离破碎，把它放在后面分析。

#### plugin\_startup

调用my->chain->startup，还是controller的处理

#### plugin\_shutdown

各种资源释放，以及my->chain.reset();