# Content\_update 数据结构整理

## 1. 新增用户授予平台权限

\\libraries\chain\include\graphene\chain\account\_object.hpp

1. **enum** Platform\_Auth\_Permission
2. {
3. Platform\_Permission\_Forward  = 1,   //允许平台转发文章，默认允许
4. Platform\_Permission\_Liked    = 2,   //允许平台点赞或打分，默认允许
5. Platform\_Permission\_Buyout   = 4,   //允许平台买断文章收益，默认允许
6. Platform\_Permission\_Comment  = 8,   //允许平台评论其他文章，默认允许
7. Platform\_Permission\_Reward   = 16,  //允许平台打赏其他文章，默认允许
8. Platform\_Permission\_Transfer = 32   //允许平台动用零钱转账，默认不允许
9. Platform\_Permission\_Post = 64, //允许平台发布文章，默认允许
10. };

更新说明：用户授予平台能代理用户做操作的权限，使用uint32\_t的标志位，方便之后做权限扩展。以上所有平台代理账户操作都需要账户授权，否则操作失效。

## 2. 新增作者授予文章权限

\\libraries\chain\include\graphene\chain\content\_object.hpp

1. **enum** Post\_Permission
2. {
3. Post\_Permission\_Forward = 1,   //文章允许转发，默认允许
4. Post\_Permission\_Liked   = 2,   //文章允许打分、点赞，默认允许
5. Post\_Permission\_Buyout  = 4,   //文章允许出售收益，默认允许
6. Post\_Permission\_Comment = 8,   //文章允许评论，默认允许
7. Post\_Permission\_Reward  = 16   //文章允许打赏，默认允许
8. };

更新说明：作者（外加平台）授予文章的权限，使用uint32\_t的标志位。所有对文章的相关操作，需要作者打开文章相关权限，否则操作实效。

## 3. 新增文章类型

\\libraries\chain\include\graphene\chain\protocol\content.hpp

1. **enum** Post\_Type
2. {
3. Post\_Type\_Post = 0,   //新文章
4. Post\_Type\_Comment = 1,   //评论
5. Post\_Type\_forward = 2,   //转发的文章
6. Post\_Type\_forward\_And\_Modify = 3,  //转发且修改的文章
8. Post\_Type\_Default
9. };

更新说明：添加了文章的类型，具体见上。

## 4. 扩展account\_auth\_platform\_operation

\\libraries\chain\include\graphene\chain\protocol\account.hpp

1. **struct** account\_auth\_platform\_operation : **public** base\_operation
2. {
3. **struct** extension\_parameter
4. {
5. optional<share\_type> limit\_for\_platform;  //扩展： 账户授予平台使用零钱的最大限额。当限额使用完了，所有平台动用账户零钱的操作将失败，直到账户提高平台限额。
6. optional<uint32\_t>   permission\_flags;    //扩展： 账户授予平台权限，权限按标志位设定，具体设定参见 Platform\_Auth\_Permission
7. optional<memo\_data>  memo;                //扩展： memo信息
8. };
10. **struct** fee\_parameters\_type
11. {
12. uint64\_t fee              = 1 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
13. uint64\_t min\_real\_fee     = 0;
14. uint16\_t min\_rf\_percent   = 0;
15. extensions\_type   extensions;
16. };
18. fee\_type                   fee;
19. account\_uid\_type           uid;
20. account\_uid\_type           platform;
22. optional< extension<extension\_parameter> > extensions;
23. };

更新说明：账户授权平台，在原来的account\_auth\_platform\_operation基础上，扩展添加了对授权平台动用零钱的限额，以及平台各类代理操作的限定。账户授予平台权限，会创建account\_auth\_platform\_object用于记录授权状态。在硬分叉之前，会新建account\_auth\_platform\_object，其它逻辑不变。在硬分叉之后，extensions扩展必须存在，用于创建或更新对平台的授权。

附加改动：1. account\_cancel\_auth\_platform\_operation 账户取消对平台操作，会删除建立的account\_auth\_platform\_object，删除对平台的多签。

2. account\_update\_auth\_operation，新增了暂停和恢复对平台授权的意义，在account\_auth\_platform\_operation授权平台后，可以通过去掉平台的多签，暂停对平台的授权，对应account\_auth\_platform\_object的is\_active为false。之后可也可以通过加回对平台的多签，来实现激活对平台的授权，对应account\_auth\_platform\_object的is\_active为true。

## 5. 扩展post\_operation

\\libraries\chain\include\graphene\chain\protocol\content.hpp

1. **class** Receiptor\_Parameter  //扩展：文章受益人参数
2. {
3. **public**:
4. uint16\_t       cur\_ratio;   //扩展: 当前的文章收益占比
5. **bool**           to\_buyout;   //扩展：收益是否出售，出售比例不得大于自己的当前占比，当该文章目前的受益人（包括平台）已经满了5人，必须全部出售自己的收益占比
6. uint16\_t       buyout\_ratio;  //扩展：收益出售比例，
7. share\_type     buyout\_price;  //扩展：收益出售价格
8. time\_point\_sec buyout\_expiration = time\_point\_sec::maximum();  //扩展：收益出售过期时间，当时间过期，该收益出售将无效
9. }
11. **struct** post\_operation : **public** base\_operation
12. {
13. **struct** ext
14. {
15. optional<uint8\_t>                                      post\_type = Post\_Type\_Post;   //扩展：文章类型，参见Post\_Type
16. optional<share\_type>                                   forward\_price;   //扩展：转发价
17. optional< map<account\_uid\_type, Receiptor\_Parameter> > receiptors;   //扩展：文章受益人列表，列表强制包括文章平台，且强制占比25%，不可修改，包括文章作者，至少占比25%。收益列表受益人数必须是[1,5]之间,当平台和作者为同一人时，可为1
18. optional<license\_lid\_type>                             license\_lid;   //扩展：文章许可id
19. optional<uint32\_t>                                     permission\_flags = 0xFF;   //扩展：文章权限，参见Post\_Permission
20. };
22. **struct** fee\_parameters\_type {
23. uint64\_t fee              = 1 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
24. uint32\_t price\_per\_kbyte  = 10 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
25. uint64\_t min\_real\_fee     = 0;
26. uint16\_t min\_rf\_percent   = 0;
27. extensions\_type   extensions;
28. };
30. fee\_type                     fee;
32. post\_pid\_type                post\_pid;
33. account\_uid\_type             platform = 0;
34. account\_uid\_type             poster = 0;
35. optional<account\_uid\_type>   origin\_poster;
36. optional<post\_pid\_type>      origin\_post\_pid;
37. optional<account\_uid\_type>   origin\_platform;
39. string                       hash\_value;
40. string                       extra\_data = "{}";
41. string                       title;
42. string                       body;
44. optional< extension<post\_operation::ext>> extensions;
45. };

更新说明：在原来的post\_operation基础上，添加了文章类型，文章转发价，文章许可，文章权限，以及文章受益人列表等扩展属性。

## 6. 扩展post\_update\_operation

\\libraries\chain\include\graphene\chain\protocol\content.hpp

1. **struct** post\_update\_operation : **public** base\_operation
2. {
3. **struct** ext
4. {
5. optional<share\_type>           forward\_price; //扩展：修改的文章转发价
6. optional<account\_uid\_type>     receiptor; //扩展：受益人账户id
7. optional<**bool**>                 to\_buyout; //扩展：是否出售收益
8. optional<uint16\_t>             buyout\_ratio; //扩展：出售收益比例，出售比例不得大于自己的当前占比。当该文章目前的受益人（包括平台）已经满了5人，必须全部出售自己的收益占比。
9. optional<share\_type>           buyout\_price; //扩展：出售收益价格
10. optional<time\_point\_sec>       buyout\_expiration; //扩展：出售挂单过期时间
11. optional<license\_lid\_type>     license\_lid; //扩展：文章许可id
12. optional<uint32\_t>             permission\_flags; //扩展：文章权限，参见Post\_Permission
13. };
14. **struct** fee\_parameters\_type {
15. uint64\_t fee              = 1 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
16. uint32\_t price\_per\_kbyte  = 10 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
17. uint64\_t min\_real\_fee     = 0;
18. uint16\_t min\_rf\_percent   = 0;
19. extensions\_type   extensions;
20. };
22. fee\_type                     fee;
24. account\_uid\_type             platform;
25. account\_uid\_type             poster;
26. post\_pid\_type                post\_pid;
28. optional< string >           hash\_value;
29. optional< string >           extra\_data;
30. optional< string >           title;
31. optional< string >           body;
33. optional< extension<post\_update\_operation::ext> > extensions;
34. };

更新说明：在原来的post\_update\_operation基础上，添加了文章转发价，文章许可id，文章权限，以及文章受益人的是否出售文章受益，出售收益比，出售价格，挂单有效时间等扩展属性。

## 7. 新增score\_create\_operation

\\libraries\chain\include\graphene\chain\protocol\content.hpp

1. **struct** score\_create\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = GRAPHENE\_BLOCKCHAIN\_PRECISION/100;
5. uint32\_t price\_per\_kbyte = 0;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;  //手续费
13. account\_uid\_type             from\_account\_uid;  //打分或点赞发起人账户id
14. account\_uid\_type             platform;   //文章所在平台
15. account\_uid\_type             poster;   //文章的作者
16. post\_pid\_type                post\_pid;   //文章的id
17. int8\_t                       score;   //文章打的分数，分数范围为-5~5
18. share\_type                   csaf;   //打分或点赞花费的积分
20. extensions\_type              extensions;  //扩展项
21. };

更新说明：新增score\_create\_operation，用于账户对文章进行打分或者点赞。文章需要开放打分权限，用户需要授权平台打分权限，否则操作失败。数据库底层新建score\_object来存储打分信息。每一个分配周期会参考打分信息来评定收益。超过最大的保存时间（由理事会参数决定）的，score\_object会被数据库删除。

## 8. 新增reward\_operation

\\libraries\chain\include\graphene\chain\protocol\content.hpp

1. **struct** reward\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = GRAPHENE\_BLOCKCHAIN\_PRECISION/10;
5. uint32\_t price\_per\_kbyte = 0;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;  //手续费
13. account\_uid\_type             from\_account\_uid; //打赏发起人账户id
14. account\_uid\_type             platform;   //文章所在平台
15. account\_uid\_type             poster;   //文章作者
16. post\_pid\_type                post\_pid;  //文章id
17. asset                        amount;   //打赏给文章的资产，可以是任意的资产
19. extensions\_type              extensions;  //扩展项
20. };

更新说明：新增reward\_operation，用于账户对文章进行打赏。文章需要开放打赏权限，否则失败。打赏的资产可以是任意链上资产（不限定为YOYO），打赏的资产由文章的受益人门按照收益比例平分，该收益直接结算，不进入收益分配环节。

## 9. 新增reward\_proxy\_operation

\\libraries\chain\include\graphene\chain\protocol\content.hpp

1. **struct** reward\_proxy\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = GRAPHENE\_BLOCKCHAIN\_PRECISION/10;
5. uint32\_t price\_per\_kbyte = 0;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;  //手续费
13. account\_uid\_type             from\_account\_uid; //代理打赏的账户id
14. account\_uid\_type             platform;   //文章所在平台
15. account\_uid\_type             poster;   //文章作者
16. post\_pid\_type                post\_pid;  //文章的id
17. share\_type                   amount;   //打上给文章的金额，代理打赏是平台代理用户操作，此处是动用账户的零钱，会使用账户授予平台的零钱限额
19. extensions\_type              extensions;  //扩展项
20. };

更新说明：新增reward\_proxy\_operation，用于平台代理账户来打赏文章。文章需要开放打赏权限，用户需要授权平台打赏权限，否则失败。该操作动用账户授予平台的零钱，消耗账户授予平台的额度，因此资产必须是YOYO，且平台必须是账户允许代理打赏的。打赏的YOYO由文章的受益人门按照收益比例平分，该收益直接结算，不进入收益分配环节。

## 10. 新增buyout\_operation

\\libraries\chain\include\graphene\chain\protocol\content.hpp

1. **struct** buyout\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = GRAPHENE\_BLOCKCHAIN\_PRECISION/10;
5. uint32\_t price\_per\_kbyte = 0;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;  //手续费
13. account\_uid\_type             from\_account\_uid; //买断文章受益的发起账户id
14. account\_uid\_type             platform;   //文章所在平台
15. account\_uid\_type             poster;   //文章的作者
16. post\_pid\_type                post\_pid;  //文章的id
17. account\_uid\_type             receiptor\_account\_uid; //要买断的受益人的id
19. extensions\_type              extensions;  //扩展项
20. };

更新说明：新增buyout\_operation，用于文章收益人对文章收益比的交易。文章需要开发买断权限，用户需要授权平台买断权限，否则失败。该操作动用账户零钱，消耗用户授予平台的额度。用于买断的YOYO会从发起账户的零钱转给收益卖出账户的零钱。

## 11. 新增license\_create\_operation

\\libraries\chain\include\graphene\chain\protocol\content.hpp

1. **struct** license\_create\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = GRAPHENE\_BLOCKCHAIN\_PRECISION/10;
5. uint32\_t price\_per\_kbyte = 1 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;  //手续费
13. license\_lid\_type             license\_lid;   //许可的id
14. account\_uid\_type             platform = 0;   //许可所在平台
15. uint8\_t                      type;   //许可类型，目前范围未定
16. string                       hash\_value;   //许可的hash
17. string                       extra\_data = "{}"; //许可的额外内容
18. string                       title;   //许可的标题
19. string                       body;   //许可的内容
21. extensions\_type              extensions;  //扩展项
22. };

更新说明：新增license\_create\_operation，用于平台用户新建文章许可。数据库底层新建license\_object来存储。许可在链上只可以新建，不可以修改。当文章许可声明需要修改的时候，需新建许可内容，并更新文章许可号。

## 12. 新增score\_object

\\libraries\chain\include\graphene\chain\content\_object.hpp

1. **class** score\_object : **public** graphene::db::abstract\_object<score\_object>
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id = implementation\_ids;
5. **static** **const** uint8\_t type\_id = impl\_score\_object\_type;
7. account\_uid\_type    from\_account\_uid; //打分或点赞的发起账户id
8. account\_uid\_type    platform;         //文章所在平台
9. account\_uid\_type    poster;           //文章作者
10. post\_pid\_type       post\_pid;         //文章的id
11. int8\_t              score;            //打分的分数，范围为-5~5
12. share\_type          csaf;             //花费的积分数
13. uint64\_t            period\_sequence;  //周期号
14. share\_type          profits;          //点赞收益
15. time\_point\_sec      create\_time;      //创建时间
16. };

更新说明：数据库新增score\_object用于存储点赞打分信息，每一个分配周期会参考打分信息来评定收益。超过最大的保存时间（由理事会参数决定）的，score\_object会被数据库删除。

## 13. 新增license\_object

\\libraries\chain\include\graphene\chain\content\_object.hpp

1. **class** license\_object : **public** graphene::db::abstract\_object<license\_object>
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id = implementation\_ids;
5. **static** **const** uint8\_t type\_id = impl\_license\_object\_type;
7. license\_lid\_type             license\_lid;  //许可的id
8. account\_uid\_type             platform;     //平台账户的id
9. uint8\_t                      license\_type; //许可类型，暂时没有预定具体类型
11. string                       hash\_value;   //许可的hash
12. string                       extra\_data;   //许可的额外内容
13. string                       title;   //许可的标题
14. string                       body;   //许可的内容
16. time\_point\_sec               create\_time;   //许可的创建时间
17. };

更新说明：数据库新增license\_object用于存储平台用户新建的许可信息。许可只能是平台来创建，且不可被修改。用户发布文章的时候，需要选择平台发布的相应的许可。

## 14. 新增active\_post\_object

\\libraries\chain\include\graphene\chain\content\_object.hpp

1. **class** active\_post\_object : **public** graphene::db::abstract\_object < active\_post\_object >
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id   = protocol\_ids;
5. **static** **const** uint8\_t type\_id    = active\_post\_object\_type;
7. **struct** receiptor\_detail //收益人的收益详细
8. {
9. share\_type  forward;                          //转发收益和
10. share\_type  post\_award;                       //来自内容资金池的收益
11. flat\_map<asset\_aid\_type, share\_type> rewards; //打赏收益和
12. };
14. account\_uid\_type                       platform;                  // 平台账号
15. account\_uid\_type                       poster;                    // 作者账号
16. post\_pid\_type                          post\_pid;                  // 同一账号同一平台下帖子唯一标识（pid）
17. share\_type                             total\_csaf;                // 文章总共积分
18. flat\_map<asset\_aid\_type, share\_type>   total\_rewards;             // 文章打赏
19. uint64\_t                               period\_sequence;           // 文章所属奖励周期数
21. **bool**                                   positive\_win;             //true: 正方胜(赞大于踩) false:反方胜（踩大于赞）
22. share\_type                             post\_award;               //文章的来自平台资金池(内容)的收益
23. share\_type                             forward\_award;            //文章的打赏收益和
24. flat\_map<account\_uid\_type, receiptor\_detail> receiptor\_details;  //收益人的收益表
25. }

更新说明：文章在获得点赞或打赏时创建此object（如同一评奖周期内已存在该object，修改属性即可），用于内容评奖周期到来时计算评奖所用。

数据库默认保存最近十个评奖周期内的object，新增命令行参数active-post-periods，用于设置数据库可保存多少期object。

## 15. 理事会参数扩展

1. **struct** content\_parameter\_extension\_type
2. {
3. uint32\_t   content\_award\_interval              = GRAPHENE\_DEFAULT\_CONTENT\_AWARD\_INTERVAL;                // 内容奖励周期，评奖周期。初始值为0，  0代表关闭，单位是秒。
4. uint32\_t   platform\_award\_interval             = GRAPHENE\_DEFAULT\_PLATFORM\_AWARD\_INTERVAL;               // 平台投票奖励周期。初始值为0 ，0代表关闭，单位是秒。
5. share\_type max\_csaf\_per\_approval               = GRAPHENE\_DEFAULT\_MAX\_CSAF\_PER\_APPROVAL;                 // 单次点赞消耗最大积分。初始值为1000积分 ，需要注意积分精度。
6. uint32\_t   approval\_expiration                 = GRAPHENE\_DEFAULT\_APPROVAL\_EXPIRATION;                   // 点赞过期时间, 同一篇文章多久可以点赞一次。初始值为1年  ，单位是秒，
7. share\_type min\_effective\_csaf                  = GRAPHENE\_DEFAULT\_MIN\_EFFECTIVE\_CSAF;                    // 文章参与评奖所需最小积分。初始值为100积分 ，注意积分精度
8. share\_type total\_content\_award\_amount          = GRAPHENE\_DEFAULT\_TOTAL\_CONTENT\_AWARD\_AMOUNT;            // 【内容资金池】总共文章奖励金额。初始值为0，一年量,每年给文章的总奖励。要考虑精度。
9. share\_type total\_platform\_content\_award\_amount = GRAPHENE\_DEFAULT\_TOTAL\_PLATFORM\_CONTENT\_AWARD\_AMOUNT;   // 【平台资金池（内容）】平台从文章中收益奖励金额。初始值为0，一年量  。
10. share\_type total\_platform\_voted\_award\_amount   = GRAPHENE\_DEFAULT\_TOTAL\_PLATFORM\_VOTED\_AWARD\_AMOUNT;     // 【平台资金池（投票）】总共平台投票奖励金额。初始值为0，一年量。平台投票激励
11. share\_type platform\_award\_min\_votes            = GRAPHENE\_DEFAULT\_PLATFORM\_AWARD\_MIN\_VOTES;              // 平台参与评奖所需最小投票数。初始值为10  ，需考虑精度
12. uint32\_t   platform\_award\_requested\_rank       = GRAPHENE\_DEFAULT\_PLATFORM\_AWARD\_REQUESTED\_RANK;         // 平台参与评奖所需投票排名数。初始值为100  ，即：前100名才有激励资格。
13. uint32\_t   platform\_award\_basic\_rate           = GRAPHENE\_DEFAULT\_PLATFORM\_AWARD\_BASIC\_RATE;             // 平台投票奖励中基础比例（平分给前100名平台）。初始值为20%
14. uint32\_t   casf\_modulus                        = GRAPHENE\_DEFAULT\_CASF\_MODULUS;                          // 积分绝对值+ （评分\*积分/5）\*系数；加权平均系数。初始值为200%
15. uint32\_t   post\_award\_expiration               = GRAPHENE\_DEFAULT\_POST\_AWARD\_EXPIRATION;                 // 文章评奖过期时间。初始值为1年  ，超过这个时间丧失评奖资格。单位是秒
16. uint32\_t   approval\_casf\_min\_weight            = GRAPHENE\_DEFAULT\_APPROVAL\_MIN\_CASF\_WEIGHT;              // 点赞积分最小权重。初始值为50%
17. uint32\_t   approval\_casf\_first\_rate            = GRAPHENE\_DEFAULT\_APPROVAL\_CASF\_FIRST\_RATE;              // 计算有效积分比例一。初始值为30%
18. uint32\_t   approval\_casf\_second\_rate           = GRAPHENE\_DEFAULT\_APPROVAL\_CASF\_SECOND\_RATE;             // 计算有效积分比例二。初始值为70%//有效积分比例二应该大于有效积分比例一
19. uint32\_t   receiptor\_award\_modulus             = GRAPHENE\_DEFAULT\_RECEIPTOR\_AWARD\_MODULUS;               // 反方胜时（踩大于赞）文章收益人奖励系数。初始值为80%
20. uint32\_t   disapprove\_award\_modulus            = GRAPHENE\_DEFAULT\_DISAPPROVE\_AWARD\_MODULUS;              // 反方胜时（踩大于赞）点踩者奖励系数。初始值为120%
21. uint32\_t   advertising\_confirmed\_fee\_rate      = GRAPHENE\_DEFAULT\_ADVERTISING\_CONFIRMED\_FEE\_RATE;        // 广告位交易确认成功手续费占比（所占交易金额的比例），初始值为0.1%
22. uint32\_t   advertising\_confirmed\_min\_fee       = GRAPHENE\_DEFAULT\_ADVERTISING\_CONFIRMED\_MIN\_FEE;         // 广告位交易确认成功最低手续费，初始值为20YOYO
23. uint32\_t custom\_vote\_effective\_time = GRAPHENE\_DEFAULT\_CUSTOM\_VOTE\_EFFECTIVE\_TIME; // 自定义投票有效时长，默认为1年；自定义投票过期时间减去创建时间不能超过该值
24. uint64\_t min\_witness\_block\_produce\_pledge = GRAPHENE\_DEFAULT\_MIN\_WITNESS\_BLOCK\_PRODUCE\_PLEDGE; //pos见证人出块所需最少的抵押资产
25. };

## 16. 扩展platform\_object

1. **class** platform\_object : **public** graphene::db::abstract\_object<platform\_object>
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id = protocol\_ids;
5. **static** **const** uint8\_t type\_id  = platform\_object\_type;
7. **struct** Platform\_Period\_Profits   //平台每周期详细收益
8. {
9. flat\_map<asset\_aid\_type, share\_type>   rewards\_profits;       //每周期文章打赏的收益和
10. share\_type                             foward\_profits    = 0; //每周期文章转发的收益和
11. share\_type                             post\_profits      = 0; //每周期来自内容资金池的收益(当作者与平台是同一账户时，还包含作者的收益)
12. share\_type                             post\_profits\_by\_platform = 0; //每周期来自内容资金池的收益（当作者与平台是同一账户时，只包含属于平台的25%收益）
13. share\_type                             platform\_profits  = 0; //每周期来自平台资金池(内容)的收益
14. };
16. account\_uid\_type owner = 0;
17. string name;
18. uint32\_t sequence;
20. **bool** is\_valid = **true**;
21. uint64\_t total\_votes = 0;
22. string url;
24. uint64\_t pledge = 0;
25. fc::time\_point\_sec  pledge\_last\_update;
26. uint64\_t            average\_pledge = 0;
27. fc::time\_point\_sec  average\_pledge\_last\_update;
28. uint32\_t            average\_pledge\_next\_update\_block;
30. map<time\_point\_sec, share\_type>        vote\_profits;   //平台投票收益,来自平台资金池(投票) ,map最大size为命令行参数active-post-periods 所设置值，默认为10
31. map<uint32\_t, Platform\_Period\_Profits> period\_profits; //平台每个周期所有的收益， ,map最大size为命令行参数active-post-periods 所设置值，默认为10
33. string extra\_data = "{}";
35. time\_point\_sec create\_time;
36. time\_point\_sec last\_update\_time;
37. };

更新说明：在platform\_object里扩展了平台的投票收益表，以及平台每个周期的其他收益表。详见注释。

## 17. 扩展dynamic\_global\_property\_object

1. **class** dynamic\_global\_property\_object : **public** abstract\_object<dynamic\_global\_property\_object>
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id = implementation\_ids;
5. **static** **const** uint8\_t type\_id  = impl\_dynamic\_global\_property\_object\_type;
7. uint32\_t                        head\_block\_number = 0;
8. block\_id\_type                   head\_block\_id;
9. time\_point\_sec                  time;
10. account\_uid\_type                current\_witness = 0;
11. time\_point\_sec                  next\_maintenance\_time;
12. time\_point\_sec                  last\_budget\_time;
13. share\_type                      witness\_budget;
14. share\_type                      budget\_pool;
15. share\_type                      total\_budget\_per\_block = 0;
16. uint32\_t                        next\_budget\_adjust\_block = 0;
17. uint32\_t                        next\_committee\_update\_block = 0;
18. committee\_proposal\_number\_type  next\_committee\_proposal\_number = 1;
19. uint32\_t                        accounts\_registered\_this\_interval = 0;
21. time\_point\_sec                  next\_content\_award\_time = time\_point\_sec(0);   //下一次内容奖励时间，默认为0；内容奖励开启时设置该时间，关闭时，设置为0
22. time\_point\_sec                  last\_content\_award\_time = time\_point\_sec(0);   //上一次内容奖励时间，默认为0；内容奖励开启时设置该时间，关闭时，设置为0
23. time\_point\_sec                  next\_platform\_voted\_award\_time = time\_point\_sec(0);   //下一次平台投票奖励时间，默认为0；平台投票奖励开启时设置该时间，关闭时，设置为0
24. time\_point\_sec                  last\_platform\_voted\_award\_time = time\_point\_sec(0);   //上一次平台投票奖励时间，默认为0；平台投票奖励开启时设置该时间，关闭时，设置为0
25. uint64\_t                        current\_active\_post\_sequence = 0;   //当前的评奖周期的编号即当前活跃文章周期数
27. **bool**                            content\_award\_enable = **false**;   //内容评奖开关，内容奖励开启时设置为true，关闭时设置为false;用于限制active\_post\_object的创建与修改，即当content\_award\_enable为false时，不能创建与修改active\_post\_object,反之即可

share\_type total\_witness\_pledge = 0; //所有见证人的抵押资产，用于计算硬分叉之后的见证人出块收益

1. bool reduce\_witness\_csaf = false; //硬分叉之后，见证人的抵押资产不再参与产生积分
2. uint16\_t by\_pledge\_witness\_pay\_per\_block = 0; //见证人抵押出块工资，每一轮见证人更新之后更新该数值，如果不存在抵押见证人出块，该值设为0

         uint32\_t           recently\_missed\_count = 0;

1. uint64\_t                current\_aslot = 0;

         fc::uint128\_t recent\_slots\_filled;

1. uint32\_t dynamic\_flags = 0;
3. uint32\_t last\_irreversible\_block\_num = 0;
5. **enum** dynamic\_flag\_bits
6. {
7. maintenance\_flag = 0x01
8. };
9. };
10. }}

## 18. 扩展post\_object

1. **class** post\_object : **public** graphene::db::abstract\_object<post\_object>
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id = protocol\_ids;
5. **static** **const** uint8\_t type\_id  = post\_object\_type;
7. account\_uid\_type             platform;
8. account\_uid\_type             poster;
9. post\_pid\_type                post\_pid;
10. optional<account\_uid\_type>   origin\_poster;
11. optional<post\_pid\_type>      origin\_post\_pid;
12. optional<account\_uid\_type>   origin\_platform;
14. string                       hash\_value;
15. string                       extra\_data;
16. string                       title;
17. string                       body;
19. time\_point\_sec create\_time;
20. time\_point\_sec last\_update\_time;
22. map<account\_uid\_type, Receiptor\_Parameter> receiptors;               //文章受益人列表
23. optional<share\_type>                       forward\_price;            //转发价格
24. optional<license\_lid\_type>                 license\_lid;              //许可的id
25. uint32\_t                                   permission\_flags = 0xFF;  //文章的权限
26. **bool**                                       score\_settlement = **false**; //是否结算点赞人收益
27. };

更新说明：在post\_object里扩展了受益人列表，转发价格，许可id等等。详见注释。

## 19. 新增advertising\_object

\\libraries\chain\include\graphene\chain\advertising \_object.hpp

1. **class** advertising\_object : **public** graphene::db::abstract\_object<advertising\_object>
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id = implementation\_ids;
5. **static** **const** uint8\_t type\_id = impl\_advertising\_object\_type;
7. advertising\_aid\_type       advertising\_aid;          //广告位id
8. account\_uid\_type           platform;                 //平台的账户id
9. **bool**                       on\_sell;                  //是否在售
10. uint32\_t                   unit\_time;                //出售的单位时间
11. share\_type                 unit\_price;               //出售的单位时间的价格
12. string                     description;              //描述
13. uint64\_t                   last\_order\_sequence = 0;  //广告订单自增序号
15. time\_point\_sec             publish\_time;             //发布时间
16. time\_point\_sec             last\_update\_time;         //更新时间
17. };

更新说明：新增advertising\_object用于平台的广告位的交易。仅允许平台来创建广告位，该object展示广告位的基本信息，包括在售状态，出售单元时间，单元价格等等。平台可以修改广告位的上架下架。

## 20. 新增advertising\_order\_object

1. **enum** advertising\_state
2. {
3. advertising\_undetermined = 0,  //订单待确定状态
4. advertising\_accepted     = 1,  //订单接受状态
5. advertising\_refused      = 2,  //订单拒绝状态
6. advertising\_ransom       = 3   //订单赎回状态
7. };
8. **class** advertising\_order\_object : **public** graphene::db::abstract\_object < advertising\_order\_object >
9. {
10. **public**:
11. **static** **const** uint8\_t space\_id = implementation\_ids;
12. **static** **const** uint8\_t type\_id = impl\_advertising\_order\_object\_type;
14. advertising\_order\_oid\_type  advertising\_order\_oid; //广告订单的id
15. account\_uid\_type            platform;              //广告所在平台的uid
16. advertising\_aid\_type        advertising\_aid;       //广告位的id
18. account\_uid\_type            user;                  //广告用户的uid
19. share\_type                  released\_balance;      //订单锁定金额
20. time\_point\_sec              start\_time;            //订单开始时间
21. time\_point\_sec              end\_time;              //订单结束时间
22. time\_point\_sec              buy\_request\_time;      //订单发起时间
23. advertising\_state           status;                //订单确认状态
24. time\_point\_sec              handle\_time;           //订单处理时间
26. optional<memo\_data>         memo;                  //memo信息
27. string                      extra\_data;            //额外信息
28. };

更新说明：新增advertising\_order\_object用于生成广告交易的订单，用户可以查询广告位的信息，发起自己购买请求，生成advertising\_order\_object，平台确定用户的订单，则结算订单，并退回和结算订单时间冲突的订单的锁定金额。若平台拒绝，则退回购买者的锁定资金。七天时间内，平台没有处理待定的订单，则用户可以赎回锁定在购买订单的资金。

## 21. 新增advertising\_create\_operation

[\\libraries\chain\include\graphene\chain\protocol\advertising.hpp](file:///\\libraries\chain\include\graphene\chain\protocol\advertising.hpp)

1. **struct** advertising\_create\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = GRAPHENE\_BLOCKCHAIN\_PRECISION/10;
5. uint32\_t price\_per\_kbyte = 1 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;             //手续费
13. advertising\_aid\_type         advertising\_aid; //广告位的id
14. account\_uid\_type             platform;        //平台的账户id
15. uint32\_t                     unit\_time;       //出售的单位时间
16. share\_type                   unit\_price;      //出售的单位时间的价格
17. string                       description;     //描述
19. extensions\_type              extensions;      //扩展项
20. };

更新说明：用于平台创建广告位，默认创建广告位，即上架广告位。广告位按照单位时间和单位价格来计费。需要平台active权限。

## 22. 新增advertising\_update\_operation

[\\libraries\chain\include\graphene\chain\protocol\advertising.hpp](file:///\\libraries\chain\include\graphene\chain\protocol\advertising.hpp)

1. **struct** advertising\_update\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = GRAPHENE\_BLOCKCHAIN\_PRECISION/10;
5. uint32\_t price\_per\_kbyte = 1 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;            //手续费
13. account\_uid\_type             platform;       //平台的账户id
14. advertising\_aid\_type         advertising\_aid; //广告位的id
16. optional<string>             description;    //描述
17. optional<share\_type>         unit\_price;     //单位时间
18. optional<uint32\_t>           unit\_time;      //单位时间的出售价格
19. optional<**bool**>               on\_sell;        //在售状态
21. extensions\_type              extensions;     //扩展项
22. };

更新说明：用于平台更新编辑广告位信息。如描述，价格，上架下架等等。上架，即开放购买，用户可以发购买请求到广告位的待定序列中；下架，即关闭购买，用户发起的购买请求不接受。需要平台active权限。

## 23. 新增advertising\_buy\_operation

[\\libraries\chain\include\graphene\chain\protocol\advertising.hpp](file:///\\libraries\chain\include\graphene\chain\protocol\advertising.hpp)

1. **struct** advertising\_buy\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = GRAPHENE\_BLOCKCHAIN\_PRECISION/10;
5. uint32\_t price\_per\_kbyte = 1\* GRAPHENE\_BLOCKCHAIN\_PRECISION;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;                    //手续费
12. advertising\_order\_oid\_type   advertising\_order\_oid;  //广告订单的id
13. account\_uid\_type             from\_account;           //购买账户的id
14. account\_uid\_type             platform;               //广告位所在平台的id
15. advertising\_aid\_type         advertising\_aid;        //广告位的id
16. time\_point\_sec               start\_time;             //开始生效时间
17. uint32\_t                     buy\_number;             //购买多少个单位时间广告
19. string                       extra\_data;             //额外信息
20. optional<memo\_data>          memo;                   //memo信息
22. extensions\_type              extensions;             //扩展项
23. };

更新说明：用于用户发起广告位的购买请求。需要广告位上架状态，并且购买时间和广告位的已成交的时间不冲突。购买发起成功，会锁定相应的购买金额到订单中，此处扣除的是用户的余额，需要用户的acive权限。

## 24. 新增advertising\_confirm\_operation

[\\libraries\chain\include\graphene\chain\protocol\advertising.hpp](file:///\\libraries\chain\include\graphene\chain\protocol\advertising.hpp)

1. **struct** advertising\_confirm\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = GRAPHENE\_BLOCKCHAIN\_PRECISION/10;
5. uint32\_t price\_per\_kbyte = 0;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;                  //手续费
13. account\_uid\_type             platform;             //平台的账户id
14. advertising\_aid\_type         advertising\_aid;      //广告位的id
15. advertising\_order\_oid\_type   advertising\_order\_oid;//订单的id
16. **bool**                         iscomfirm;            //是否确认订单
18. extensions\_type              extensions;           //扩展项
19. };

更新说明：平台账户确认或者拒绝购买订单。确认订单，结算订单，订单金额扣除千分之一手续费给资金池（若千分之一不足20YOYO，按20YOYO计算）；并退回和结算订单时间冲突的订单的锁定金额给购买用户。拒绝订单，订单金额返还给购买用户。需要平台的active权限。

1. **typedef** flat\_map<account\_uid\_type, share\_type> advertising\_confirm\_result;
3. **class** advertising\_confirm\_evaluator : **public** evaluator < advertising\_confirm\_evaluator >
4. {
5. **public**:
6. **typedef** advertising\_confirm\_operation operation\_type;
8. void\_result do\_evaluate(**const** operation\_type& op);
9. advertising\_confirm\_result do\_apply(**const** operation\_type& op);
11. **const** advertising\_object\* advertising\_obj = nullptr;
12. };

do\_apply接口返回值类型: map<返还账户ID， 返还金额>；

确认广告购买操作时，接受账户A订单，返回列表添加 { A， 0 }和｛　与A订单时间存在冲突的待确认订单账户, 购买时锁定的金额 ｝; 拒绝账户A订单，返回｛ A，购买时锁定的金额 ｝;

举例：账户A、B、C同时花费10000YOYO购买2019/4/1-2019/5/1时间段，D、F同时花费10000YOYO购买2019/6/1-2019/7/1时间段；

1. 如果确认A，那么返回{ { A，0 }，{ B，10000 }， { C, 10000 } };
2. 如果拒绝D，那么返回{ { D，10000 } }；

## 25. 新增advertising\_ransom\_operation

[\\libraries\chain\include\graphene\chain\protocol\advertising.hpp](file:///\\libraries\chain\include\graphene\chain\protocol\advertising.hpp)

1. **struct** advertising\_ransom\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = 0 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
5. uint32\_t price\_per\_kbyte = 10 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;                  //手续费
13. account\_uid\_type             from\_account;         //账户的id
14. account\_uid\_type             platform;             //广告位所在平台的id
15. advertising\_id\_type          advertising\_id;       //广告位的id
16. advertising\_order\_id\_type    advertising\_order\_id; //订单的id
18. extensions\_type              extensions;           //扩展项
19. };

更新说明：当用户发起的购买订单，七天时间内平台没有处理，用户可以自己发起赎回，返还未处理订单锁定的金额，将订单状态置为赎回状态除。需要用户的active权限。

## 26. 新增account\_auth\_platform\_object

1. **class** account\_auth\_platform\_object : **public** graphene::db::abstract\_object<account\_auth\_platform\_object>
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id = implementation\_ids;
5. **static** **const** uint8\_t type\_id = impl\_account\_auth\_platform\_object\_type;
7. account\_uid\_type  account;   //账户的id
8. account\_uid\_type  platform;  //平台的id
10. share\_type             max\_limit = 0;  //账户授权平台的最大使用额度
11. share\_type             cur\_used = 0;   //平台当前使用的额度
12. bool is\_active = true; //授权激活状态
13. optional<memo\_data>    memo;           //memo信息
14. uint32\_t      permission\_flags = account\_auth\_platform\_object::Platform\_Permission\_Forward |
15. account\_auth\_platform\_object::Platform\_Permission\_Liked |
16. account\_auth\_platform\_object::Platform\_Permission\_Buyout |
17. account\_auth\_platform\_object::Platform\_Permission\_Comment |
18. account\_auth\_platform\_object::Platform\_Permission\_Reward |
19. account\_auth\_platform\_object::Platform\_Permission\_Post |
20. account\_auth\_platform\_object::Platform\_Permission\_Transfer; //账户授予平台的权限
21. };

更新说明：新增account\_auth\_playform\_object用于记录账户授予平台的信息，包括最大使用额度，当前使用额度，操作权限等等。在account\_auth\_platform\_operation时，创建该oject。在account\_cancel\_auth\_platform\_operation时，删除该object. 可使用账号多签操作来激活或者暂停对平台的授权。

## 27. 新增custom\_vote\_object

1. **class** custom\_vote\_object : **public** graphene::db::abstract\_object<custom\_vote\_object>
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id = implementation\_ids;
5. **static** **const** uint8\_t type\_id = impl\_custom\_vote\_object\_type;
7. account\_uid\_type           custom\_vote\_creater;          //自定义投票创建者
8. custom\_vote\_vid\_type vote\_vid;                //自定义投票id
10. string                     title;                        //自定义投票主题
11. string                     description;                  //自定义投票描述
12. time\_point\_sec             vote\_expired\_time;            //自定义投票过期时间
14. asset\_aid\_type             vote\_asset\_id;                //用于自定义投票的资产id
15. share\_type                 required\_asset\_amount;        //账户投票所需最少资产数量
16. uint8\_t                    minimum\_selected\_items;       //投票至少选择数目
17. uint8\_t                    maximum\_selected\_items;       //投票至多选择数目
19. std::vector<string>        options;                      //自定义投票选项
20. std::vector<uint64\_t>      vote\_result;                  //每个投票选项得票数
21. };

更新说明：新增custom\_vote\_object用于记录自定义投票活动信息， 该信息的保留时长可由命令行参数custom-vote-remain-time设置；每个投票选项得票数，是依据投票人的资产实时更新，当投票人自定义投票资产余额变更时，投票人对应投票选项票数也对应变更。

## 28. 新增custom\_vote\_create\_operation

[\\libraries\chain\include\graphene\chain\protocol\custom\_vote.hpp](file:///\\libraries\chain\include\graphene\chain\protocol\custom_vote.hpp)

1. **struct** custom\_vote\_create\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = 1 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
5. uint32\_t price\_per\_kbyte = 1 \* GRAPHENE\_BLOCKCHAIN\_PRECISION;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                   fee;                      //手续费
13. account\_uid\_type           custom\_vote\_creater;      //自定义投票创建者
14. custom\_vote\_vid\_type vote\_vid;                //自定义投票id
15. string                     title;                    //自定义投票主题
16. string                     description;              //自定义投票描述
17. time\_point\_sec             vote\_expired\_time;        //自定义投票过期时间
18. asset\_aid\_type             vote\_asset\_id;            //用于自定义投票的资产id
19. share\_type                 required\_asset\_amount;    //账户投票所需最少资产数量
20. uint8\_t                    minimum\_selected\_items;   //投票至少选择数目
21. uint8\_t                    maximum\_selected\_items;   //投票至多选择数目
23. std::vector<string>        options;                  //自定义投票选项
24. extensions\_type            extensions;               //扩展项
25. };

更新说明：创建自定义投票活动操作，投票选项数目须大于1小于256， 投票至少选择数目须大于1，至多选择数目不能超过 （投票选项数目 – 1）， 投票有效时长不能超过理事会设定的有效时长（custom\_vote\_effective\_time），任何账户都可创建自定义投票，需要用户的active权限。

## 29. 新增cast\_custom\_vote\_object

1. **class** cast\_custom\_vote\_object : **public** graphene::db::abstract\_object<cast\_custom\_vote\_object>
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id = implementation\_ids;
5. **static** **const** uint8\_t type\_id = impl\_cast\_custom\_vote\_object\_type;
7. account\_uid\_type           voter;                //自定义投票账户
8. account\_uid\_type           custom\_vote\_creater;  //自定义投票创建者
9. custom\_vote\_vid\_type custom\_vote\_vid;      //自定义投票id
10. std::set<uint8\_t>          vote\_result;          //账户投票结果
11. time\_point\_sec             vote\_expired\_time;    //自定义投票过期时间
12. asset\_aid\_type             vote\_asset\_id;        //用于自定义投票的资产id
13. };

更新说明：新增cast\_custom\_vote\_object用于记录自定义投票活动投票信息， 该信息的保留时长可由命令行参数custom-vote-remain-time设置。

## 30. 新增custom\_vote\_cast\_operation

[\\libraries\chain\include\graphene\chain\protocol\custom\_vote.hpp](file:///\\libraries\chain\include\graphene\chain\protocol\custom_vote.hpp)

1. **struct** custom\_vote\_cast\_operation : **public** base\_operation
2. {
3. **struct** fee\_parameters\_type {
4. uint64\_t fee = GRAPHENE\_BLOCKCHAIN\_PRECISION/10;
5. uint32\_t price\_per\_kbyte = 0;
6. uint64\_t min\_real\_fee = 0;
7. uint16\_t min\_rf\_percent = 0;
8. extensions\_type   extensions;
9. };
11. fee\_type                     fee;                //手续费
13. account\_uid\_type             voter;              //自定义投票账户
14. account\_uid\_type           custom\_vote\_creater;  //自定义投票创建者
15. custom\_vote\_vid\_type custom\_vote\_vid;      //自定义投票id
16. std::set<uint8\_t>            vote\_result;        //账户投票结果
18. extensions\_type              extensions;         //扩展项
20. };

更新说明：针对自定义投票活动投票或者更新投票操作，选项得票数立即刷新，自定义投票活动创建者也可投票，需要用户的active权限。

如果该账户已经对投票主题投票，再一次进行投票时，定义为更新投票，更新投票结果可与之前投票结果相同

## 31. 扩展account\_object

[\\libraries\chain\include\graphene\chain\account\_object.hpp](file:///\\libraries\chain\include\graphene\chain\account_object.hpp)

1. **class** account\_object : **public** graphene::db::abstract\_object<account\_object>
2. {
3. **public**:
4. **static** **const** uint8\_t space\_id = protocol\_ids;
5. **static** **const** uint8\_t type\_id  = account\_object\_type;
7. time\_point\_sec membership\_expiration\_date;
8. uint32\_t         referrer\_by\_platform = 0;  //if referrered by platform, == platform.sequence
10. account\_uid\_type registrar;
11. account\_uid\_type referrer;
12. account\_uid\_type lifetime\_referrer;
14. uint16\_t network\_fee\_percentage = GRAPHENE\_DEFAULT\_NETWORK\_PERCENT\_OF\_FEE;
15. uint16\_t lifetime\_referrer\_fee\_percentage = 0;
16. uint16\_t referrer\_rewards\_percentage = 0;
18. account\_uid\_type uid = 0;
19. string name;
21. authority owner;
22. authority active;
23. authority secondary;
25. public\_key\_type       memo\_key;
26. account\_reg\_info      reg\_info;
28. **bool** can\_post         = **true**;
29. **bool** can\_reply        = **false**;
30. **bool** can\_rate         = **false**;
32. **bool** is\_full\_member   = **false**;
33. **bool** is\_registrar     = **false**;
34. **bool** is\_admin         = **false**;
36. time\_point\_sec        create\_time;
37. time\_point\_sec        last\_update\_time;
39. string active\_data    = "{}";
40. string secondary\_data = "{}";
42. account\_statistics\_id\_type             statistics;
43. flat\_set<account\_uid\_type>             whitelisting\_accounts;
44. set<account\_uid\_type>                  whitelisted\_accounts;
45. set<account\_uid\_type>                  blacklisted\_accounts;

48. flat\_set<account\_uid\_type>             blacklisting\_accounts;
49. optional< flat\_set<asset\_aid\_type> >   allowed\_assets;
51. };

更新说明：account\_object 添加参数referrer\_by\_platform，表示账户的创建由平台引荐。用于账户给平台投票的时候，会默认给引荐的平台投票。详见平台计票调整。若平台被用户取消（将平台的抵押更新为0），引荐的平台失效，那么用户不会默认给引荐的平台投票。

# Database\_api 接口修改整理

## 1. 新增获取点赞打分

1. score\_object get\_score(**const** account\_uid\_type platform,     //文章所在平台
2. **const** account\_uid\_type poster\_uid,   //文章作者
3. **const** post\_pid\_type    post\_pid,     //文章id
4. **const** account\_uid\_type from\_account  //点赞打分人的账户id
5. )**const**;

返回 score\_object的详细信息。

示例：*get\_score platform\_account1 test\_account1 1 test\_account2*

1. vector<score\_object> list\_scores(**const** account\_uid\_type platform,              //文章所在平台
2. **const** account\_uid\_type poster\_uid,            //文章作者
3. **const** post\_pid\_type    post\_pid,              //文章id
4. **const** object\_id\_type   lower\_bound\_score,     //起始score的id
5. **const** uint32\_t         limit,                 //展示数目限制
6. **const** **bool**             list\_cur\_period = **true** //是否只展示当前收益分配周期 true:只展示当前收益周期的点赞；false：展示所有点赞。
7. )**const**;

返回该文章所有的点赞信息。

示例：*list\_scores platform\_account1 test\_account1 1 2.16.6 100 true*

## 2. 新增获取许可信息

1. license\_object get\_license(**const** account\_uid\_type platform,    //平台的账户id
2. **const** license\_lid\_type license\_lid  //许可的id
3. )**const**;

返回license\_object的详细信息

示例：*get\_license platform\_account1 1*

1. vector<license\_object> list\_licenses(**const** account\_uid\_type platform,           //平台账户的id
2. **const** object\_id\_type lower\_bound\_license,  //起始license的id
3. **const** uint32\_t limit                       //展示数目限制
4. )**const**;

返回该平台下所有的许可信息。

示例：*list\_licenses platform\_account1 2.17.3 100*

## 3. 新增获取文章收益

1. vector<active\_post\_object> get\_post\_profits\_detail(**const** uint32\_t         begin\_period, //起始周期号
2. **const** uint32\_t         end\_period,   //截至周期号
3. **const** account\_uid\_type platform,     //文章所在平台
4. **const** account\_uid\_type poster,       //文章的作者
5. **const** post\_pid\_type    post\_pid      //文章的id
6. )**const**;

返回从起始周期到截至周期的文章收益详情vector<active\_post\_object>。

示例：*get\_post\_profits\_detail 1 5 platform\_account1 auther\_account 1*

## 4. 新增获取平台收益

1. **struct** Platform\_Period\_Profit\_Detail  //平台周期收益详情
2. {
3. uint32\_t                               cur\_period;       //当前周期号
4. account\_uid\_type                       platform\_account; //平台所有人账户
5. string                                 platform\_name;    //平台名称
7. flat\_map<asset\_aid\_type, share\_type>   rewards\_profits;      //每周期文章打赏的收益和
8. share\_type                             foward\_profits = 0;   //每周期文章转发的收益和
9. share\_type                             post\_profits = 0;     ///每周期来自内容资金池的收益(当作者与平台是同一账户时，还包含作者的收益)
10. share\_type                             post\_profits\_by\_platform = 0; //每周期来自内容资金池的收益（当作者与平台是同一账户时，只包含属于平台的25%收益）
11. share\_type                             platform\_profits = 0; //每周期来自平台资金池(内容)的收益
13. vector< active\_post\_object >            active\_post\_pids;     //所有有收益的active\_objects
14. };
16. vector<Platform\_Period\_Profit\_Detail> get\_platform\_profits\_detail(**const** uint32\_t         begin\_period,       //起始周期号
17. **const** uint32\_t         end\_period,         //截至周期号
18. **const** account\_uid\_type platform            //平台的所有人
19. **const** uint32\_t         lower\_bound\_index,  //起始active序号
20. uint32\_t               limit               //显示数目限制
21. )**const**;

返回从起始周期到截至周期的平台收益详情，vector<Platform\_Period\_Profit\_Detail>。

示例：*get\_platform\_profits\_detail 1 5 platform\_account1 1 100*

## 5. 新增获取作者收益

1. **struct** Poster\_Period\_Profit\_Detail  //作者周期收益详情
2. {
3. uint32\_t                               cur\_period;           //当前周期号
4. account\_uid\_type                       poster\_account;       //作者
6. share\_type                             total\_forward = 0;    //每周期文章转发的收益和
7. flat\_map<asset\_aid\_type, share\_type>   total\_rewards;        //每周期文章打赏的收益和
8. share\_type                             total\_post\_award = 0; //每周期来自内容资金池的收益
10. vector< active\_post\_object >            active\_post\_pids;     //所有有收益的active\_objects
11. };
13. vector<Poster\_Period\_Profit\_Detail> get\_poster\_profits\_detail(**const** uint32\_t         begin\_period,       //起始周期号
14. **const** uint32\_t         end\_period,         //截至周期号
15. **const** account\_uid\_type poster              //作者
16. **const** uint32\_t         lower\_bound\_index,  //起始active序号
17. uint32\_t               limit               //显示数目限制
18. )**const**;

返回从起始周期到截至周期的作者收益详情，vector<Poster\_Period\_Profit\_Detail>。

示例：*get\_poster\_profits\_detail 1 5 auther\_account 1 100*

## 6. 新增获取点赞收益

1. share\_type get\_score\_profit(account\_uid\_type account, //账户的id
2. uint32\_t         period   //收益分配的周期号
3. )**const**;

返回某一周期的某账户所有的点赞收益。

示例：*get\_score\_profit test\_account1 5*

## 7. 新增获取广告位

1. optional<advertising\_object> get\_advertising(**const** account\_uid\_type platform,            //平台的账户id
2. **const** advertising\_aid\_type advertising\_aid  //起始广告的id
3. )**const**;
5. vector<advertising\_object> list\_advertisings(**const**  string         platform,                //平台的账户id
6. **const**  object\_id\_type lower\_bound\_advertising, //起始广告的id
7. uint32\_t        limit                    //展示的限制数目
8. );

新增获取某一广告get\_advertising，列出平台所有的广告位list\_advertisings。

示例：  *get\_advertising platform\_account 5 list\_advertisings platform\_account 5 100*

## 8. 新增获取广告订单

1. vector<advertising\_order\_object> list\_advertising\_orders\_by\_purchaser(**const** account\_uid\_type  purchaser,                     //广告购买者id
2. **const** object\_id\_type    lower\_bound\_advertising\_order, //起始广告订单id
3. uint32\_t                limit)**const**;                   //展示数目限制
5. vector<advertising\_order\_object> list\_advertising\_orders\_by\_ads\_aid(**const** account\_uid\_type           platform,                     //广告所在平台id
6. **const** advertising\_aid\_type       id,                           //广告位id
7. **const** advertising\_order\_oid\_type lower\_bound\_advertising\_order,//起始广告订单id
8. uint32\_t                         limit)**const**;                  //展示数目限制

新增获取广告位订单

示例：*list\_advertising\_orders\_by\_purchaser test\_account 10 100 list\_advertising\_orders\_by\_ads\_id platform 2 30 100*

## 9. 新增获取自定义投票对象

1. vector<custom\_vote\_object> lookup\_custom\_votes(**const** account\_uid\_type  creater,                //创建投票的账户id
2. **const** custom\_vote\_vid\_type  lower\_bound\_custom\_vote,  //起始投票对象的id
3. uint32\_t  limit                         //展示数目限制
4. )**const**;
6. vector<custom\_vote\_object> list\_custom\_votes(**const** account\_uid\_type lowerbound, //起始的账户id
7. uint32\_t limit                     //展示数目限制
8. )**const**;

新增获取自定义投票对象信息

示例：*lookup\_custom\_votes test\_account 1 10 list\_custom\_votes test\_account 10*

## 10. 新增获取自定义投票详情

1. vector<cast\_custom\_vote\_object> list\_cast\_custom\_votes\_by\_id(const account\_uid\_type creater //创建投票的账户id
2. custom\_vote\_vid\_type  vote\_id,                      //投票对象的id
3. **const** object\_id\_type  lower\_bound\_cast\_custom\_vote, //起始投票详情对象的id
4. uint32\_t limit                                      //展示数目限制
5. )**const**;
7. vector<cast\_custom\_vote\_object> list\_cast\_custom\_votes\_by\_voter(account\_uid\_type  voter,                            //投票人的账户id
8. **const** object\_id\_type  lower\_bound\_cast\_custom\_vote,  //起始投票对象的id
9. uint32\_t  limit                              //展示数目限制
10. )**const**;

新增获取自定义投票的详情

示例：*list\_cast\_custom\_votes\_by\_id test\_account 1 10 list\_cast\_custom\_votes\_by\_voter test\_account 2.21.6 10*

## 11. 新增获取账户授权平台详情

1. vector<account\_auth\_platform\_object> list\_account\_auth\_platform\_by\_platform(**const** account\_uid\_type platform,            //平台的账户id
2. **const** account\_uid\_type lower\_bound\_account, //起始用户的账户id
3. **const** uint32\_t limit                        //展示数目限制
4. )**const**;
6. vector<account\_auth\_platform\_object> list\_account\_auth\_platform\_by\_account(**const** account\_uid\_type account,              //用户的账户id
7. **const** account\_uid\_type lower\_bound\_platform, //起始平台的账户id
8. **const** uint32\_t limit                         //展示数目限制
9. )**const**;

新增查询账户授权了多少平台，以及查询平台有多少用户授权

示例：*list\_account\_auth\_platform\_by\_platform platform\_account test\_account 10 list\_account\_auth\_platform\_by\_account test\_account platform\_account 10*

## 12. 新增获取平台授权账户数量

1. uint64\_t get\_account\_auth\_platform\_count(**const** account\_uid\_type platform)**const**;

新增查询平台下授权该平台的账户数量

示例：*get\_account\_auth\_platform\_count platform\_account*

## 13. 新增获取账户点赞收益详情

1. vector<score\_object> get\_scores\_by\_uid(**const** account\_uid\_type  scorer,         //点赞者
2. **const** uint32\_t period,                  //周期号
3. **const** object\_id\_type lower\_bound\_score, //起始打分id
4. **const** uint32\_t limit)**const**;             //展示数目限制

新增查询账户查询某个周期内的点赞详细收益

示例：*get\_scores\_by\_uid 33013 12 2.20.1 100*

## 14. 新增获取文章数目

1. uint64\_t get\_posts\_count(optional<account\_uid\_type> platform,        //平台
2. optional<account\_uid\_type> poster)**const**;    //作者

新增查询发表文章数目接口

1. 平台为null,作者为null： 返回所有平台下的所有文章总数；
2. 平台为null,作者不为null时：抛出异常，不允许存在这种情况；
3. 平台不为null, 作者不为null时：返回该平台下该作者发表的所有文章数目；
4. 平台不为null, 作者为null时：返回该平台下所有作者发表的所有文章数目；

示例：

*1.get\_posts\_count null null*

*2.get\_posts\_count 33013 33034*

*3.get\_posts\_count 33013 null*

# Wallet接口修改整理

## 1. 新增transfer\_extension

1. signed\_transaction transfer\_extension(string from,                //转出人的id或者名字
2. string to,                  //转入人的id或者名字
3. string amount,              //交易数额
4. string asset\_symbol,        //交易的资产
5. string memo,                //交易的备注
6. **bool** isfrom\_balance = **true**, //是否从余额转出。true，从余额转出；false，从零钱转出
7. **bool** isto\_balance = **true**,   //是否转到余额。true，转到余额；false，转到零钱
8. **bool** csaf\_fee = **true**,       //使用积分付手续费。true，积分付手续费；false，余额付手续费
9. **bool** broadcast = **false**      //是否广播
10. );

更新说明：transfer\_extension 用于支持账户之间的余额、零钱的相互交易。

示例：*transfer\_extension test\_account1 test\_account1 5000 YOYO "" true false true true*

## 2. 扩展account\_auth\_platform

1. signed\_transaction account\_auth\_platform(string account,              //账户的id或名字
2. string platform\_owner,       //平台的账户id或名字
3. string memo,                 //memo信息
4. string limit\_for\_platform,   //授予平台零钱的使用限额
5. uint32\_t permission\_flags =  //授予平台的操作权限
6. account\_auth\_platform\_object::Platform\_Permission\_Forward |  //转帖权
7. account\_auth\_platform\_object::Platform\_Permission\_Liked |    //点赞权
8. account\_auth\_platform\_object::Platform\_Permission\_Buyout |   //买断权
9. account\_auth\_platform\_object::Platform\_Permission\_Comment |  //评论权
10. account\_auth\_platform\_object::Platform\_Permission\_Reward |   //打赏权
11. account\_auth\_platform\_object::Platform\_Permission\_Post,      //发贴权
12. **bool** csaf\_fee  = **true**,       //使用积分付手续费。true，积分付手续费；false，余额付手续费
13. **bool** broadcast = **false**       //是否广播
14. );

更新说明：相对于原来的account\_auth\_platform增加了，账户授予平台的零钱使用限额，以及授予平台的各类操作权限。

示例：*account\_auth\_platform test\_account8 platform\_account1 memo 10000 255 true true*

## 3. 新增create\_post

1. **struct** post\_create\_ext
2. {
3. uint8\_t                                           post\_type = post\_operation::Post\_Type\_Post; //文章类型
4. optional<std::string>                             forward\_price;                              //转发价格
5. optional< map<account\_uid\_type, receiptor\_ext> >  receiptors;                                 //受益人列表
6. optional<license\_lid\_type>                        license\_lid;                                //文章许可id
7. uint32\_t                                          permission\_flags =                          //文章开放的权限
8. post\_object::Post\_Permission\_Forward |  //文章允许转发
9. post\_object::Post\_Permission\_Liked |    //文章允许点赞
10. post\_object::Post\_Permission\_Buyout |   //文章允许买断
11. post\_object::Post\_Permission\_Comment |  //文章允许评论
12. post\_object::Post\_Permission\_Reward;    //文章允许打赏
13. };
15. signed\_transaction create\_post(string           platform,               //文章发布的平台账户的id或名字
16. string           poster,                 //文章作者的账户id或者名字
17. string           hash\_value,             //文章hash
18. string           title,                  //文章标题
19. string           body,                   //文章内容
20. string           extra\_data,             //文章额外内容
21. string           origin\_platform = "",   //文章的原平台
22. string           origin\_poster = "",     //文章的原作者
23. string           origin\_post\_pid = "",   //文章的原id
24. post\_create\_ext  ext = post\_create\_ext(),//扩展结构，参见上文结构体
25. **bool** csaf\_fee  = **true**,                   //使用积分付手续费。true，积分付手续费；false，余额付手续费
26. **bool** broadcast = **false**                   //是否广播
27. );

更新说明：新增create\_post用于创建文章，回复文章，转发文章等等。

示例：*create\_post platform\_account1 test\_account1 987456321 postname postbody postextry "" "" "" {post\_type :0, forward\_price :null, receiptors :null, license\_lid :1, permission\_flags :255} true true*

## 4. 新增update\_post

1. **struct** post\_update\_ext
2. {
3. optional<std::string>       forward\_price;     //更新转发价，需要作者签名
4. optional<std::string>       receiptor;         //更新的目标受益人，收益人相关内容更新，需要该收益人签名
5. optional<**bool**>              to\_buyout;         //更新是否出售收益，需要目标收益人签名
6. optional<uint16\_t>          buyout\_ratio;      //更新出售收益比率，需要目标收益人签名
7. optional<std::string>       buyout\_price;      //更新出售收益价格，需要目标收益人签名
8. optional<time\_point\_sec>    buyout\_expiration; //更新出售收益过期时间，需要目标收益人签名
9. optional<license\_lid\_type>  license\_lid;       //更新文章许可id ，需要作者签名
10. optional<uint32\_t>          permission\_flags;  //更新文章开放权限 ，需要作者签名
11. };
13. signed\_transaction update\_post(string           platform,              //文章所在平台
14. string           poster,                //文章作者
15. string           post\_pid,              //文章的id
16. string           hash\_value = "",       //文章的hash
17. string           title = "",            //文章的标题
18. string           body = "",             //文章的内容
19. string           extra\_data = "",       //文章的额外内容
20. post\_update\_ext ext = post\_update\_ext(),//扩展结构，参见上文结构体
21. **bool** csaf\_fee  = **true**,                  //使用积分付手续费。true，积分付手续费；false，余额付手续费
22. **bool** broadcast = **false**                  //是否广播
23. );

更新说明：新增update\_post，用于更新文章内容，开发权限，许可id，转发价以及受益人出售收益的挂单。

示例：*update\_post platform\_account1 test\_account1 2 "" "" "" "" {forward\_price : 500, receiptor : test1, to\_buyout : true, buyout\_ratio : 3000, buyout\_price : 300, buyout\_expiration :* *1546272000, license\_lid : 1, permission\_flags : null} true true*

## 5. 新增score\_a\_post

1. signed\_transaction score\_a\_post(string           from\_account, //点赞打分发起人的账户id或者名字
2. string           platform,     //文章所在平台
3. string           poster,       //文章作者
4. post\_pid\_type    post\_pid,     //文章的id
5. int8\_t           score,        //打的分数，范围-5~5
6. string           csaf,         //花费的积分数
7. **bool** csaf\_fee  = **true**,         //使用积分付手续费。true，积分付手续费；false，余额付手续费
8. **bool** broadcast = **false**         //是否广播
9. );

更新说明：新增score\_a\_post用于账户对文章帖子点赞打分，打分范围为-5~5，需要花费账户积分。

示例：*score\_a\_post test\_account2 platform\_account1 test\_account1 2 5 10 true true*

## 6. 新增reward\_post

1. signed\_transaction reward\_post(string           from\_account, //打赏人的账户id或者名字
2. string           platform,     //文章所在平台
3. string           poster,       //文章作者的账户id或者名字
4. post\_pid\_type    post\_pid,     //文章的id
5. string           amount,       //打赏的金额
6. string           asset\_symbol, //打赏的资产名
7. **bool** csaf\_fee  = **true**,         //使用积分付手续费。true，积分付手续费；false，余额付手续费
8. **bool** broadcast = **false**         //是否广播
9. );

更新说明：新增reward\_post用于个人账户对文章的打赏，打赏可以是链上任意资产。

示例：*reward\_post test\_account3 platform\_account1 test\_account1 2 1000 YOYO true true*

## 7. 新增reward\_post\_proxy\_by\_platform

1. signed\_transaction reward\_post\_proxy\_by\_platform(string           from\_account, //打赏人的账户id或者名字
2. string           platform,     //文章所在平台
3. string           poster,       //文章作者的账户id或者名字
4. post\_pid\_type    post\_pid,     //文章的id
5. string           amount,       //打赏的金额，动用的是授权给平台的零钱，必须是YOYO
6. **bool** csaf\_fee  = **true**,         //使用积分付手续费。true，积分付手续费；false，余额付手续费
7. **bool** broadcast = **false**         //是否广播
8. );

更新说明：新增reward\_post\_proxy\_by\_platform用于平台代理账户对文章进行打赏。动用的是用户授权给平台的零钱，需要相应的授权并消耗授权的零钱额度，打赏资产必须是YOYO。

示例：*reward\_post\_proxy\_by\_platform test\_account3 platform\_account1 test\_account1 2 1000 true true*

## 8. 新增buyout\_post

1. signed\_transaction buyout\_post(string           from\_account,     //买断发起人的账户id或者名字
2. string           platform,         //文章所在平台
3. string           poster,           //文章作者的账户id或者名字
4. post\_pid\_type    post\_pid,         //文章的id
5. string           receiptor\_account,//买断的受益人的账户id或者名字
6. **bool** csaf\_fee  = **true**,             //使用积分付手续费。true，积分付手续费；false，余额付手续费
7. **bool** broadcast = **false**             //是否广播
8. );

更新说明：新增buyout\_post用于买断文章受益人的收益比。其中买断的受益人receiptor\_account，必须先完成出售文章受益的有效挂单，参见update\_post操作。

示例：*buyout\_post test\_account3 platform\_account1 test\_account1 2 test\_account1 true true*

## 9. 新增create\_license

1. signed\_transaction create\_license(string           platform,     //创建的平台的账户id或者名字
2. uint8\_t          license\_type, //许可的类型
3. string           hash\_value,   //许可的hash
4. string           title,        //许可的标题
5. string           body,         //许可的内容
6. string           extra\_data,   //许可的额外内容
7. **bool** csaf\_fee  = **true**,         //使用积分付手续费。true，积分付手续费；false，余额付手续费
8. **bool** broadcast = **false**         //是否广播
9. );

更新说明：新增create\_license用于平台账户创建许可。仅有平台账户才可以创建许可，且许可不可修改。

示例：*create\_license platform\_account1 1 123456789 license\_one licensebody licenseextra true true*

## 10. 新增account\_manage

1. **struct** opt
2. {
3. optional< **bool** > can\_post;  //允许发帖
4. optional< **bool** > can\_reply; //允许回复
5. optional< **bool** > can\_rate;  //允许点赞
6. };
8. signed\_transaction account\_manage(string executor,                       //管理员账户的id或名字
9. string account,                        //管理的账户
10. account\_manage\_operation::opt options, //扩展结构，参见上文结构体
11. **bool** csaf\_fee  = **true**,                 //使用积分付手续费。true，积分付手续费；false，余额付手续费
12. **bool** broadcast = **false**                 //是否广播
13. );

更新说明：新增account\_manage用于账户的管理。account\_manage\_operation仅账户注册者可以管理账户的发帖，回复，点赞权。

示例：*account\_manage init test\_account1 {can\_post : null, can\_reply : true, can\_rate : true} true true*

## 11. 新增get\_post

1. post\_object get\_post(string platform\_owner, //文章发布的平台账户的id或名字
2. string poster\_uid,     //文章作者的账户id或者名字
3. string post\_pid        //文章的id
4. );

更新说明：新增get\_post用于post\_object的查询。

示例：*get\_post platform\_account1 test\_account1 1*

## 12. 新增get\_score

1. score\_object get\_score(string platform,     //文章所在平台
2. string poster\_uid,   //文章作者的账户id或者名字
3. string post\_pid,     //文章的id
4. string from\_account  //点赞打分的账户id或者名字
5. );

更新说明：新增get\_score用于score\_object的查询。

示例：*get\_score platform\_account1 test\_account1 1 test\_account2*

## 13. 新增get\_license

1. license\_object get\_license(string platform,   //创建许可的平台的账户id或者名字
2. string license\_lid //许可的id
3. );

更新说明：新增get\_license用于license\_object的查询。

示例：*get\_license platform\_account1 1*

## 14. 新增get\_global\_properties\_extensions

1. get\_global\_properties\_extensions();

更新说明：新增get\_global\_properties\_extensions用于扩展理事会参数查询

示例：*get\_global\_properties\_extensions*

## 15. 更新committee\_proposal\_create扩展参数示例

理事会扩展参数提案创建示例：

*committee\_proposal\_create init1 [[3 , {content\_award\_interval : 300,platform\_award\_interval : 300,max\_csaf\_per\_approval : 1000,approval\_expiration : 31536000,min\_effective\_csaf : 10,total\_content\_award\_amount : 10000000000000,total\_platform\_content\_award\_amount : 10000000000000,total\_platform\_voted\_award\_amount : 10000000000000,platform\_award\_min\_votes : 1,platform\_award\_requested\_rank : 100, platform\_award\_basic\_rate : 3000, casf\_modulus : 30000, post\_award\_expiration : 62000000, approval\_casf\_min\_weight : 4000, approval\_casf\_first\_rate : 2000, approval\_casf\_second\_rate : 8000, receiptor\_award\_modulus : 9000, disapprove\_award\_modulus : 11000}]] 200 1 200 200 true true*

## 16. 新增get\_posts\_by\_platform\_poster

1. vector<post\_object> get\_posts\_by\_platform\_poster(string           platform\_owner,   //文章发布的平台账户的id或名字
2. optional<string> poster,           //文章作者的账户id或者名字
3. uint32\_t         begin\_time\_range, //开始时间
4. uint32\_t         end\_time\_range,   //结束时间
5. object\_id\_type   lower\_bound\_post, //起始文章id
6. uint32\_t         limit             //展示数目限制
7. );

更新说明：新增get\_posts\_by\_platform\_poster，用于通过平台作者查询所有时间限制内发布的文章。当poster为空的时候，查询平台下时间限制内发布的文章。

示例：*get\_posts\_by\_platform\_poster platform\_account1 test\_account1 1546272000 1547026795 2.7.9 100*

## 17. 新增list\_scores

1. vector<score\_object> list\_scores(string         platform,                 //文章所在平台的账户id或者名字
2. string         poster\_uid,               //文章作者的账户id或者名字
3. string         post\_pid,                 //文章的id
4. object\_id\_type lower\_bound\_score,        //起始点赞的id
5. uint32\_t       limit,                    //展示数目限制
6. **bool**           list\_cur\_period = **true**    //是否仅展示当前收益分配周期的点赞。true：只展示当前周期点赞；false：展示所有点赞。
7. );

更新说明：新增list\_scores，用于通过文章来查询所有的当前收益分配周期或者所有的点赞信息。

示例：*list\_scores platform\_account1 test\_account1 1 2.16.2 100 true*

## 18. 新增list\_licenses

1. vector<license\_object> list\_licenses(string  platform,  //平台的账户id或名字
2. object\_id\_type lower\_bound\_license //起始license的id
3. uint32\_t  limit    //展示数目限制
4. );

更新说明：新增list\_licenses，用于通过平台查询所有该平台发布的许可信息。

示例：*list\_licenses platform\_account1 2.17.6 100*

## 19. 新增get\_account\_statistics

1. account\_statistics\_object get\_account\_statistics(string account // 账户的id或者名称
2. );

更新说明：新增get\_account\_statistics，用于查看账户的统计信息。

示例：*get\_account\_statistics test\_account1*

## 20. 新增get\_post\_profits\_detail

1. vector<active\_post\_object> get\_post\_profits\_detail(uint32\_t         begin\_period,       //起始周期号
2. uint32\_t         end\_period,         //截至周期号
3. string           platform,           //文章所在平台的账户id或名字
4. string           poster,             //文章的作者的账户id或名字
5. string           post\_pid            //文章的id
6. );

更新说明：查询从起始周期到截至周期的文章的所有收益详情。

示例：*get\_post\_profits\_detail 1 5 platform\_account1 auther\_account 1*

## 21. 新增get\_platform\_profits\_detail

1. vector<Platform\_Period\_Profit\_Detail> get\_platform\_profits\_detail(uint32\_t         begin\_period,  //起始周期号
2. uint32\_t         end\_period,    //截至周期号
3. string           platform,       //平台的账户id或名字
4. uint32\_t lower\_bound\_index, //起始active序号
5. uint32\_t limit //显示数目
6. );

更新说明：查询从起始周期到截至周期的平台的所有收益详情。

示例：*get\_platform\_profits\_detail 1 5 platform\_account1 1 100*

## 22. 新增get\_poster\_profits\_detail

1. vector<Poster\_Period\_Profit\_Detail> get\_poster\_profits\_detail(uint32\_t         begin\_period,  //起始周期号

                                                              uint32\_t         end\_period,    //截至周期号

1. string           poster         //作者的账户id或名字

                                                              uint32\_t lower\_bound\_index, //起始active序号

1. uint32\_t limit //显示数目
2. );

更新说明：查询从起始周期到截至周期的作者的所有收益详情。

示例：*get\_poster\_profits\_detail 1 5 auther\_account 10 100*

## 23. 新增get\_score\_profit

1. share\_type get\_score\_profit(string    account, //账户的名称或者id
2. uint32\_t  period   //收益分配的周期号
3. );

更新说明：查询某账户在某收益周期的点赞总收入

示例：*get\_score\_profit test\_account1 1*

## 24. 新增create\_advertising

1. signed\_transaction create\_advertising(string           platform,     //平台账户的id或者名字
2. string           description,  //描述
3. string           unit\_price,   //单位时间的价格
4. uint32\_t         unit\_time,    //单位时间
5. **bool** csaf\_fee = **true**,          //使用积分付手续费。true，积分付手续费；false，余额付手续费
6. **bool** broadcast = **false**         //是否广播
7. );

更新说明：平台创建新的广告位。

示例：*create\_advertising platform\_account ad\_desc 100 86400 true true*

## 25. 新增update\_advertising

1. signed\_transaction update\_advertising(string                     platform,       //平台账户的id或者名字
2. advertising\_aid\_type       advertising\_aid, //广告的id
3. optional<string>           description,    //描述
4. optional<string>           unit\_price,     //单位时间的价格
5. optional<uint32\_t>         unit\_time,      //单位时间
6. optional<**bool**>             on\_sell,        //出售状态
7. **bool** csaf\_fee = **true**,                      //使用积分付手续费。true，积分付手续费；false，余额付手续费
8. **bool** broadcast = **false**                     //是否广播
9. );

更新说明：平台编辑广告位信息，上架、下架广告位。

示例：*update\_advertising platform\_account 1 ad\_update 200 86400 null true true*

## 26. 新增buy\_advertising

1. signed\_transaction buy\_advertising(string               account,          //账户的id或名字
2. string               platform,         //平台账户的id或者名字
3. advertising\_aid\_type advertising\_aid,   //广告的id
4. uint32\_t             start\_time,       //开始时间
5. uint32\_t             buy\_number,       //购买数目
6. string               extra\_data,       //额外信息
7. string               memo,             //memo信息
8. **bool**                 csaf\_fee = **true**,  //使用积分付手续费。true，积分付手续费；false，余额付手续费
9. **bool**                 broadcast = **false** //是否广播
10. );

更新说明：用户发起购买广告位的请求。

示例：*buy\_advertising test\_account platform\_account 1 1551974400 30 extramessage null true true*

## 27. 新增confirm\_advertising

1. signed\_transaction confirm\_advertising(string          platform,            //平台账户的id或者名字
2. advertising\_aid\_type  advertising\_aid,  //广告的id
3. advertising\_order\_oid\_type advertising\_order\_oid, //广告订单的id
4. **bool**            comfirm,             //是否确认
5. **bool**            csaf\_fee = **true**,     //使用积分付手续费。true，积分付手续费；false，余额付手续费
6. **bool**            broadcast = **false**    //是否广播
7. );

更新说明：平台同意用户的购买请求，或者拒绝用户的购买请求。

示例：*confirm\_advertising platform\_account 1 1 true true true*

## 28. 新增ransom\_advertising

1. signed\_transaction ransom\_advertising(string            platform,         //平台账户的id或者名字
2. string            from\_account,     //用户的id或名字
3. advertising\_aid\_type    advertising\_aid,   //广告的id
4. advertising\_order\_oid\_type advertising\_order\_oid,  //订单序号
5. **bool** csaf\_fee = **true**,              //使用积分付手续费。true，积分付手续费；false，余额付手续费
6. **bool** broadcast = **false**             //是否广播
7. );

更新说明：平台七天内未处理购买请求，用户之后可以赎回锁定在订单里的金额。

示例：*ransom\_advertising platform\_account test\_account 1 1 true true*

## 29. 新增list\_advertisings

1. vector<advertising\_object> list\_advertisings(string         platform,                //平台的账户id
2. object\_id\_type lower\_bound\_advertising, //起始广告的id
3. uint32\_t       limit                    //展示的限制数目
4. );

新增获取某一广告get\_advertising，列出平台所有的广告位list\_advertisings。

示例： *list\_advertisings platform\_account 2.18.5 1000*

## 30. 新增create\_custom\_vote

1. signed\_transaction create\_custom\_vote(string           create\_account,          //创建账户的id
2. string           title,                   //自定义投票标题
3. string           description,             //自定义投票描述
4. uint32\_t         expired\_time,            //自定义投票过期时间
5. asset\_aid\_type   asset\_id,                //投票有效资产类型
6. share\_type       required\_amount,         //投票需要最低资产数
7. uint8\_t          minimum\_selected\_items,  //最小选择数
8. uint8\_t          maximum\_selected\_items,  //最大选择数
9. vector<string>   options,                 //选项详情
10. **bool**             csaf\_fee = **true**,         //使用积分付手续费。true，积分付手续费；false，余额付手续费
11. **bool**             broadcast = **false**);      //是否广播

新增发起自定义投票，支持多选，可设置最低拥有资产来限制投票资格。

示例：*create\_custom\_vote title desc 1552617830 1 1000 1 1 [1, 2, 3] true true*

## 31. 新增cast\_custom\_vote

1. signed\_transaction cast\_custom\_vote(string                 voter,                //投票人的账户id
2. string custom\_vote\_creater,      //投票对象的创建账户
3. custom\_vote\_vid\_type custom\_vote\_vid,       //投票对象的id
4. set<uint8\_t>          vote\_result,          //投的选项，和投票对象的选项对应
5. **bool**                  csaf\_fee = **true**,      //使用积分付手续费。true，积分付手续费；false，余额付手续费
6. **bool**                  broadcast = **false**);   //是否广播

新增自定义投票，用于用户对自定义的投票进行投票。需要用户拥有投票设定的最低的资产。

示例：*cast\_custom\_vote test\_account creater\_account 1 [1] true true*

## 32. 新增list\_custom\_votes

1. vector<custom\_vote\_object> list\_custom\_votes(**const** account\_uid\_type lowerbound,  //起始账户id
2. uint32\_t               limit        //展示数目限制
3. );

新增依据创建账户开始查看自定义投票，用于用户投票前浏览所有投票。

示例：*list\_custom\_votes 33018 10*

## 33. 新增lookup\_custom\_votes

1. vector<custom\_vote\_object> lookup\_custom\_votes(string           creater,                  //创建账户的id
2. **const** custom\_vote\_vid\_type  lower\_bound\_custom\_vote,  //起始投票对象的id
3. uint32\_t         limit                     //展示数目限制
4. );

新增查看某账户创建的所有自定义投票。

示例：*lookup\_custom\_votes 22069 1 10*

## 34. 新增list\_cast\_custom\_votes\_by\_id

1. vector<cast\_custom\_vote\_object> list\_cast\_custom\_votes\_by\_id(**const** string  creater,   //创建投票的账户id
2. **const** custom\_vote\_vid\_type  vote\_vid,     //投票对象的id
3. **const** object\_id\_type  lower\_bound\_cast\_custom\_vote,   //起始投票详情对象的id
4. uint32\_t  limit //展示数目限制
5. );

新增查看投票对象的自定义投票详情

示例：*list\_cast\_custom\_votes\_by\_id create\_account 1 2.21.3 10*

## 35. 新增list\_cast\_custom\_votes\_by\_voter

1. vector<cast\_custom\_vote\_object> list\_cast\_custom\_votes\_by\_voter(string         voter,                        //投票人的账户id
2. object\_id\_type lower\_bound\_cast\_custom\_vote, //起始投票详情对象的id
3. uint32\_t       limit                         //展示数目限制
4. );

新增查看账户的自定义投票详情

示例：*list\_cast\_custom\_votes\_by\_voter 22389 2.21.7 10*

## 36. 新增list\_account\_auth\_platform\_by\_platform

1. vector<account\_auth\_platform\_object> list\_account\_auth\_platform\_by\_platform(string   platform,             //平台的账户id
2. string   lower\_bound\_account,  //起始账户的id
3. uint32\_t limit = 100           //展示数目限制
4. );

新增查看平台下有哪些账户授权，及其授权详情

示例：*list\_account\_auth\_platform\_by\_platform platform\_account test\_account 10*

## 37. 新增list\_account\_auth\_platform\_by\_account

1. vector<account\_auth\_platform\_object> list\_account\_auth\_platform\_by\_account(string   account,              //用户的账户id
2. string   lower\_bound\_platform, //起始平台的账户id
3. uint32\_t limit = 100           //展示数目限制
4. );

新增查看账户授权了哪些平台，及其授权详情

示例：*list\_account\_auth\_platform\_by\_account test\_account platform\_account 10*

## 38. 新增get\_account\_auth\_platform\_count

1. uint64\_t get\_account\_auth\_platform\_count(**const** string platform)**const**;

新增查询平台下授权该平台的账户数量

示例：*get\_account\_auth\_platform\_count platform\_account*

## 39. 新增get\_scores\_by\_uid

1. vector<score\_object> get\_scores\_by\_uid(**const** string  scorer,         //点赞者
2. **const** uint32\_t period,                  //周期号
3. **const** object\_id\_type lower\_bound\_score, //起始打分id
4. **const** uint32\_t limit)**const**;             //展示数目限制

新增查询账户查询某个周期内的点赞详细收益

示例：*get\_scores\_by\_uid test1 12 2.20.1 100*

## 14. 新增get\_posts\_count

1. uint64\_t get\_posts\_count(optional<string> platform,       //平台
2. optional<string> poster);        //作者

新增查询发表文章数目接口

1. 平台为null,作者为null： 返回所有平台下的所有文章总数；
2. 平台为null,作者不为null时：抛出异常，不允许存在这种情况；
3. 平台不为null, 作者不为null时：返回该平台下该作者发表的所有文章数目；
4. 平台不为null, 作者为null时：返回该平台下所有作者发表的所有文章数目；

示例：

*1.get\_posts\_count null null*

*2.get\_posts\_count account\_platform1 test1*

*3.get\_posts\_count account\_platform1 null*

# 插件改动整理

## 1. 新增插件non\_consensus\_plugin

更新说明：将不参与共识的部分功能移到该插件，以提高出块人，验证人的执行效率。目前，将自定义投票即custom\_vote\_cast\_operation和cast\_custom\_vote\_object相关内容改到插件执行。需要使用自定义投票功能的节点可以依据插件启用说明来启用该功能。

插件启用说明：启动一个节点具备witness，account\_history，non\_consensus 功能。命令行--plugins="witness account\_history non\_consensus" --non\_consensus\_indexs=customer\_vote

其中non\_consensus\_indexs=customer\_vote表示启动非共识插件的自定义投票功能。

# 命令行选项

## 新增active-post-periods

更新说明： 新增active-post-periods命令行参数，用于设置数据库保存最近多少内容评奖周期的活跃文章对象，默认保存最近十期。

示例：*--active-post-periods=20*

## 2. 新增enable-check-invariants

更新说明： 新增enable-check-invariants命令行参数，apply block时检查所有账户余额，零钱，积分，投票等的开关；如果设置该选项表示开启检查，未设置则检查关闭。

示例：*--enable-check-invariants*

## 3. 新增advertising-remain-time

更新说明： 新增advertising-remain-time命令行参数，表示广告订单对象（advertising\_order\_object）在数据库中保留时长（已确认订单从广告订单到期时间算起， 已拒绝或者已赎回订单从订单处理时间算起， 待确认订单对象不删除）, 超过时长清除该对象，默认保留一年；假如广告订单购买时间为2019/03/14 – 2019/10/14，保留时长为1年，那么2020/10/14清除该广告订单对象。

示例：*--advertising-remain-time=31536000*

## 4. 新增custom-vote-remain-time

更新说明： 新增custom-vote-remain-time命令行参数，表示自定义投票活动对象（custom\_vote\_object）在数据库中保留时长（从自定义投票活动过期时间算起），超过时长清除该对象，并且清除所有针对该自定义投票活动产生的投票对象（cast\_custom\_vote\_object），默认保留一年；假如自定义投票活动从 2019/03/14开始，到2019/10/14终止投票，保留时长为1年，那么2020/10/14清除自定义投票活动对象和所有的针对自定义投票活动产生的投票对象。

调节custom-vote-remain-time时长可提高出块节点与验证节点的效率，保留时长越短，效率越高；

示例：*--custom-vote-remain-time=31536000*

# 评奖规则

## 平台资金池（内容）

可参与平台资金池（内容）评奖条件：

平台在评奖周期内有评分行为(即该平台内文章的打分消耗的积分总数大于0)

平台资金池（内容）分配：

根据可参与评奖平台的所有积分数（评奖周期内）占所有平台的积分数总和（评奖周期内）比例分配

## 平台资金池（投票）

可参与平台资金池（投票）评奖条件：

1. 平台投票数不小于平台参与评奖所需最小投票数（理事会参数）；
2. 平台投票数排名在平台参与评奖所需排名数（理事会参数）之内；

平台资金池（投票）分配：

平台资金池的分配基础比例（理事会参数）平均分配给可参与评奖平台；  
剩余根据可参与评奖平台的投票数占所有可参与评奖平台的投票数总和比例分配；

## 3. 内容资金池

参考以下流程图（单独文件\yoyow-core\tests\文章奖池流程图）

有效文章1

有效文章2

有效文章N

计算M = M1+M2+M3+…+Mn；根据每篇有效文章的Mi占M比例分配文章奖池

25%

75%

该文章第一次参与分配？

不分配

否

计算评分者i有效积分Si;

计算所有评分者有效积分和：S = S1+S2+S3+…+Sn

是

根据评分者i有效积分Si与总有效积分S占比计算收益Yi

活跃文章i

该文章总积分 >= 文章参与评奖最小积分（理事会参数）

不参与评奖

Mi >= 0 ?

计算公式1：Mi = 该文章积分总数 + ∑（评分\*消耗积分\*系数（理事会参数）/5）

不参与评奖

否

否

是

是

文章收入：按文章收益列表占比分配

（先分配账户，剩余分配给平台）

计算公式2：

前30%（理事会参数）的积分：100%权重；

30%~70%（理事会参数）：权重逐渐由100%降至50%（理事会参数）；后30%：固定50%

评分者i增加收益Yi

评分者i收益增加Yi\*反正者收益比例（理事会参数）

该篇文章是否赞大于踩即∑（评分\*消耗积分\*系数（理事会参数）/5 ）>= 0 ?

是

评分者i是否点赞即评分 >= 0 ?

是

否

否

该篇文章是否赞大于踩即∑（评分\*消耗积分\*系数（理事会参数）/5 ）>= 0 ?

是

否

文章收入\*文章受益人奖励系数（理事会参数）：按文章收益列表占比分配

（先分配账户，剩余分配给平台）

均匀分布计算，分三部分：

第一部分（前30%）：100%\*该部分的积分；

第二部分（30%-70%）: 先计算平均权重，再计算该部分的有效积分，即：平均权重\*第二部分的积分；

第三部分（后70%）：50%\*该部分的积分；

举例：评分者A、B、C先后都花费1000积分点赞；

则：  
A的有效积分为900+100\*97% = 997  
B的有效积分为1000\*75%=750  
C的有效积分为100\*52%+900\*50%=502

总共有效积分为997+750+502

## 4. 补充说明

### 1.全局资金储备池如果小于平台资金池【内容】与内容资金池之和，则不发放平台资金池【内容】与内容资金池奖励；

### 2.全局资金储备池如果小于平台资金池【投票】，则不发放平台资金池【投票】奖励；

# 功能调整

## 1. 积分公式调整

硬分叉之前，见证人抵押的ＹＯＹＯ资产包含在积分公式的有效资产里，可以产生积分。硬分叉之后，积分公式需要的有效资产将减去见证人抵押的那一部分，即见证人抵押的ＹＯＹＯ不再参与产生积分。

## 2. 平台计票调整

硬分叉之前，由平台引荐注册的账户不会默认直接给平台投票。硬分叉之后调整为由平台注册的用户，在平台投票的时候该账户会默认给创建的平台投票。

## 3. 抵押出块工资

HARDFORK\_0\_4\_TIME前，旷工出块收益由理事会参数决定（by\_pledge\_witness\_pay\_per\_block）；

HARDFORK\_0\_4\_TIME后，根据见证人抵押计算旷工收益（\yoyow-core\tests\经济模型-公式版），公式详情如下：

ｘ：当前抵押（千万）

ｙ：纯旷工利率

ｚ：每年支付矿工YOYO数（万）

|  |  |  |
| --- | --- | --- |
| 当前抵押（千万） | 纯矿工利率 | 每年支付矿工YOYO数（万） |
| < 1 | 25.00% | \ |
| 1 | 19.53% | 195.32318 |
| 2 | 18.08% | 361.54128 |
| 3 | 16.75% | 502.43778 |
| 4 | 15.54% | 621.54368 |
| 5 | 14.44% | 722.1375 |
| 6 | 13.45% | 807.24528 |
| 7 | 12.57% | 879.64058 |
| 8 | 11.77% | 941.84448 |
| 9 | 11.07% | 996.12558 |
| 10 | 10.45% | 1044.5 |
| 11 | 9.90% | 1088.73138 |
| 12 | 9.42% | 1130.33088 |
| 13 | 9.00% | 1170.55718 |
| 14 | 8.65% | 1210.41648 |
| 15 | 8.34% | 1250.6625 |
| 16 | 8.07% | 1291.79648 |
| 17 | 7.85% | 1334.06718 |
| 18 | 7.65% | 1377.47088 |
| 19 | 7.48% | 1421.75138 |
| 20 | 7.33% | 1466.4 |
| 21 | 7.19% | 1510.65558 |
| 22 | 7.06% | 1553.50448 |
| 23 | 6.93% | 1593.68058 |
| 24 | 6.79% | 1629.66528 |
| 25 | 6.64% | 1659.6875 |
| 26 | 6.47% | 1681.72368 |
| 27 | 6.27% | 1693.49778 |
| 28 | 6.04% | 1692.48128 |
| 29 | 5.78% | 1675.89318 |
| 30 | 5.47% | 1640.7 |
| 31 | 5.11% | 1583.61578 |
| 32 | 4.69% | 1501.10208 |
| 33 |  | 1501.10208 |
| 34 |  | 1501.10208 |
| 35 |  | 1501.10208 |
| 36 |  | 1501.10208 |
| 37 |  | 1501.10208 |
| 38 |  | 1501.10208 |
| 39 |  | 1501.10208 |
| 40 |  | 1501.10208 |
| 41 |  | 1501.10208 |
| 42 |  | 1501.10208 |
| 43 |  | 1501.10208 |
| 44 |  | 1501.10208 |
| 45 |  | 1501.10208 |
| 46 |  | 1501.10208 |
| 47 |  | 1501.10208 |
| 48 |  | 1501.10208 |
| 49 |  | 1501.10208 |
| 50 |  | 1501.10208 |
| 51 |  | 1501.10208 |
| 52 |  | 1501.10208 |
| 53 |  | 1501.10208 |
| 54 |  | 1501.10208 |
| 55 |  | 1501.10208 |
| 56 |  | 1501.10208 |
| 57 |  | 1501.10208 |
| 58 |  | 1501.10208 |
| 59 |  | 1501.10208 |
| 60 |  | 1501.10208 |
| 61 |  | 1501.10208 |
| 62 |  | 1501.10208 |
| 63 |  | 1501.10208 |
| 64 |  | 1501.10208 |
| >64 |  |  |

## 4. 见证人抵押出块调整

硬分叉之后，Pos节点抵押超过50万YOYO才可以出块，并且获取出块收益