**ABOUT MPT-ETH**

THE MPT-ETH contract is the only DAPP application to establish the ETH public chain in the world at present. It is also the only revenue platform of contract system in the world. It solves the trust relationship between investors and platforms. It uses non-codifiable electronic contracts to completely establish Re-public chain for revenue and output. This will be a new revolution in the global block chain financial market!

**ETH Contract Profit Source**

1. Futures contracts

Participate in futures trading with ETH and get a fixed return

2. Community contracts

When we invest in ecology, we can't all use it as a node, because the profit of the node is not enough for us to share. At this time, most of it will be used for carry trade.

For example, there are 10,000 ETH when 248 USDT is thrown, that is, 2.48 million USDT; when 218 USDT is taken back, 2.48 million\_228 yuan = 11.3 million ETH, one throw, one suck, net earnings of 130 ETH, and every day we are given ETH, as long as the market rises and falls, there must be a profit margin, once there is a profit margin, there must be a profit margin.

And our advantage is: with a lot of chips, you can smash, you can pull, between smash and pull, you can naturally earn the number of ETH.

3. Super Node Bonus contracts

We invest in global nodes, nodes can get ET annual additional issuance quota, such as 1 billion issuance, 5% (50 million) per year, which is allocated to these nodes 50 million.

**Contract revenue**

Members register, go to the home page to make an appointment, pay 20% of the amount of the contract at the time of the appointment, wait for the signing of the contract after the appointment is successful, the time period for signing the contract is 13:00-13:10, there will be a countdown in this period, after the countdown, there will be a signing button, Click the signing button to sign the contract, pay the remaining 80% of the contract amount. Then, the contract period dividends x ETH per day, 5 days principal plus interest and enters the membership account. At this time, the members make an appointment again. They get a dynamic award within 10 days. If they don't make an appointment after 10 days, they stop the dynamic award. The principal plus interest on the platform account can enter the Imtoken purse and complete the appearance operation.

**Share revenue**

**Five days output is allocated statically (static income part is allocated commission, the bonus can be taken repeatedly). Successful appointment triggers the payment of team dynamic bonus to the above people. To promote revenue, the following people do not make an appointment, do not issue! Promotion revenue of 0.2 ETH can be clicked to withdraw, in the upper right corner of the promotional revenue function to make a withdrawal button, Click to withdraw revenue to the platform account wallet, unified every night at 11:00 ETH, automatic ETH to IMtoken wallet.**

**Drill Members: 100% of first-generation revenue from successful signing (direct push 0, team 0)**

**Second drill members: direct push 3 people, team 10 people first generation revenue 100%, second generation revenue 40%.**

**Three Drilling Members: Direct Push 3 Second Drilling Members with first generation revenue of 100%, second generation revenue of 40%.**

**Third-generation earnings 20%**

**Four Drill Members: Direct Drill Four Second Drill Members First Generation Income 100%, Second Generation Income 40%.**

**The third generation earnings are 20%, and the fourth generation earnings are 2%.**

**Five Drilling Members: Five Drilling Members get 100% first-generation revenue and 40% second-generation revenue.**

**The third generation earnings are 20%, and the fourth generation earnings are 2%.**

**Unlimited generation 2%**

**Six drill members: I five drills, direct push a five drill first generation revenue 100%, the second generation revenue 40%.**

**The third generation earnings are 20%, and the fourth generation earnings are 2%.**

**Unlimited generation 2%**

**Six Drill Members: You can get an additional 50 ETH reward (manual) and enjoy 1% of the global dividend \* weighted average dividend every day (reward, dividend parameters can be adjusted)**

**Five Drill Members: Can get an additional 20 ETH reward, (parameter adjustable) (manual delivery)**

**Attachment:**

**Global dividend refers to the sum of all the static benefits of a platform day (weighted average of 1%, adjustable parameters)**

**Weighted averaging means, for example, when six members reach six stars, the total profit per day is 100 \* 1% / 6 and is distributed to six people on average.**

**开源代码**

**public static void StaticBonus()**

**{ //工厂注入**

**SysDbFactory dbFactory = new SysDbFactory();**

**SysDBTool sysDbTool = new SysDBTool(dbFactory);**

**UserService userService = new UserService(dbFactory);**

**BonusDetailService bonusDetailService = new BonusDetailService(dbFactory);**

**UserWalletLogService userWalletLogService = new UserWalletLogService(dbFactory);**

**InverstmentService inverstmentService = new InverstmentService(dbFactory);**

**UserWalletService userWalletService = new UserWalletService(dbFactory);**

**var getBonusReleaseList = inverstmentService.List(x => x.Status == (int)Data.Enum.InverstmentStatus.paid || x.Status == (int)Data.Enum.InverstmentStatus.effect).ToList();//找到签约成功的单**

**if (getBonusReleaseList.Count > 0)**

**{**

**List<Data.BonusDetail> BonusDetailList = new List<Data.BonusDetail>();**

**List<Data.UserWalletLog> UserWalletLogList = new List<Data.UserWalletLog>();**

**var cacheSysParam = JN.Services.Manager.CacheHelp.GetSysParamsList();//参数缓存集合**

**List<int> uIDs = getBonusReleaseList.Select(x => x.UID).Distinct().ToList();//签约成功的订单用户ID集合**

**StringBuilder walletsql = new StringBuilder();**

**StringBuilder invermentsql = new StringBuilder();**

**var parm1101 = cacheSysParam.Single(x => x.ID == 1101);//静态奖励参数**

**decimal bonus = 0;//初始化奖金金额**

**int ReleasedDay = 0;// 初始化已释放天数**

**decimal AddupInterest = 0;//初始化静态奖金累计收益**

**DateTime SettlementTime = new DateTime();//初始化时间**

**int status = 0;//初始化订单状态**

**foreach (var item in getBonusReleaseList)**

**{ //把用户信息添加到字典**

**Dictionary<int, Data.User> UserListDic = userService.List(x => uIDs.Contains(x.ID)).ToList().ToDictionary(d => d.ID, d => d);**

**if (item.AddupDay < item.StaticIssueDay)//应释放天数大于已释放天数**

**{**

**var InverstmentModel = cacheSysParam.Single(x => x.ID == item.InvestmentMode);//投资项目**

**bonus = (item.StaticBonus ?? 0);**

**if (bonus > 0)//如果奖金大于0**

**{**

**var onUser = userService.Single(item.UID);**

**if (onUser == null) continue;//如此会员不存在,跳过该会员**

**//奖金结算**

**changeWalletNoCommitAddupDic(onUser, UserListDic, bonus, parm1101.ID, parm1101.Name, "来自合约为：" + item.InvestmentNo + "的" + parm1101.Name + "：" + InverstmentModel.Value3 + "\*" + (item.Quantity ?? 0), 2002, cacheSysParam, ref BonusDetailList, ref UserWalletLogList, ref walletsql, true, true);**

**ReleasedDay = (item.AddupDay ?? 0) + 1; //奖金天数累加**

**AddupInterest = (item.AddupInterest ?? 0) + bonus;//奖金累加**

**if (ReleasedDay >= item.StaticIssueDay)//如已得奖天数大于等于应得奖天数**

**{**

**//返还预约金与合约**

**changeWalletNoCommitAddupDic(onUser, UserListDic, (item.Quantity ?? 0), 0, "", "合约：" + item.InvestmentNo + "完成，返还预约金与合约金金额：" + item.Quantity, 2002, cacheSysParam, ref BonusDetailList, ref UserWalletLogList, ref walletsql);**

**//订单分红结束（订单失效时间）**

**SettlementTime = item.SettlementEndTime ?? DateTime.Now;**

**//该项目已完成,sql更新数据**

**invermentsql.AppendFormat("update [Inverstment] set Status={0},AddupDay={1},AddupInterest={2},SettlementEndTime='{3}' where UID={4} and ID={5}", (int)Data.Enum.InverstmentStatus.completed, ReleasedDay, AddupInterest, SettlementTime, item.UID, item.ID);**

**if (onUser.IsNew == true)//完成一轮成老会员，并更新数据**

**{**

**invermentsql.AppendFormat("update [User] set IsNew=0 where ID={0}", onUser.ID);**

**}**

**}**

**else**

**{**

**//生效时间**

**SettlementTime = item.SettlementTime ?? DateTime.Now;**

**//订单状态**

**status = item.Status == (int)Data.Enum.InverstmentStatus.paid ? (int)Data.Enum.InverstmentStatus.effect : item.Status;**

**//提交数据，更新此订单**

**invermentsql.AppendFormat("update [Inverstment] set AddupDay={0},AddupInterest={1},SettlementTime='{2}',Status={3} where UID={4} and ID={5}", ReleasedDay, AddupInterest, SettlementTime, status, item.UID, item.ID);**

**}**

**}**

**}**

**}**

**using (System.Transactions.TransactionScope ts = new System.Transactions.TransactionScope())//事物提交，数据更新**

**{**

**DbParameter[] s = new System.Data.Common.DbParameter[] { };**

**if (BonusDetailList.Count > 0)**

**{**

**bonusDetailService.BulkInsert(BonusDetailList);**

**}**

**if (UserWalletLogList.Count > 0)**

**{**

**userWalletLogService.BulkInsert(UserWalletLogList);**

**}**

**if (walletsql.Length > 0)**

**{**

**sysDbTool.ExecuteSQL(walletsql.ToString(), s);**

**walletsql.Clear();**

**}**

**if (invermentsql.Length > 0)**

**{**

**sysDbTool.ExecuteSQL(invermentsql.ToString(), s);**

**invermentsql.Clear();**

**}**

**ts.Complete();**

**}**

**dbFactory.Dispose();//释放工厂**

**}**

**}**

**#endregion**