注：以下所有请求使用POST方式，成功code为200，失败为4000，参数错误为3000

[http://127.0.0.1:8888/contract/getTokenList](http://127.0.0.1:8888/contract/getTokenDecimals)

Des : 获取TokenLIst

Args:

* token : Token地址

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "list":list  
   }  
  }

<http://127.0.0.1:8888/contract/getTokenDecimals>

Des : 获取Token的小数位数

Args:

* token : Token地址

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "decimals":dec  
   }  
  }

[http://127.0.0.1:8888/contract/getPairAddress](http://127.0.0.1:8888/contract/getTokenDecimals)

Des : 获取tokenA&tokenB配对合约Pair地址

Args:

* tokenA : TokenA地址
* tokenB : TokenB地址

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "address":addr  
   }  
  }

<http://127.0.0.1:8888/contract/getPairAddressETH>

Des : 获取token&ETH配对合约Pair地址

Args:

* token : Token地址

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "address":addr  
   }  
  }

[http://127.0.0.1:8888/contract/getReserves](http://127.0.0.1:8888/contract/getTokenDecimals)

Des : 获取tokenA&tokenB的储备量

Args:

* tokenA : TokenA地址
* tokenB : TokenB地址

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "reserveA":r.reserveA,  
   "reserveB":r.reserveB  
   }  
  }

[http://127.0.0.1:8888/contract/getReservesETH](http://127.0.0.1:8888/contract/getTokenDecimals)

Des : 获取token&ETH的储备量

Args:

* token : Token地址

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "reserveA":r.reserveA,  
   "reserveB":r.reserveB  
   }  
  }

<http://127.0.0.1:8888/contract/quote>

Des : 获取tokenA对tokenB的对价

Args:

* amountADesired: A的期望数量
* reserveA: A的储备量
* reserveB: B的储备量

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "b":b  
   }  
  }

<http://127.0.0.1:8888/contract/factory>

Des : 获取工厂合约地址

Args:

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "address":addr  
   }  
  }

<http://127.0.0.1:8888/contract/WETH>

Des : 获取WETH合约地址

Args:

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "address":addr  
   }  
  }

[http://127.0.0.1:8888/contract/getAmountsIn](http://127.0.0.1:8888/contract/quote)

Des : 根据输出求输入

Args:

* amountOut: 输出数量
* path: 兑换路径数组

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "amounts":as  
   }  
  }

[http://127.0.0.1:8888/contract/getAmountsOut](http://127.0.0.1:8888/contract/quote)

Des : 根据输入求输出

Args:

* amountIn: 输入数量
* path: 兑换路径数组

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "amounts":as  
   }  
  }

[http://127.0.0.1:8888/contract/getTotalLiquidity](http://127.0.0.1:8888/contract/quote)

Des : 获取总流动性数量

Args:

* pair: 配对合约地址

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "total":total  
   }  
  }

[http://127.0.0.1:8888/contract/getPersonalLiquidity](http://127.0.0.1:8888/contract/quote)

Des : 获取个人所拥有的流动性数量

Args:

* pair: 配对合约地址
* owner:个人用户地址

Return:

* {  
   "code": 200,  
   "msg":"成功",  
   "data": {  
   "personal":personal  
   }  
  }

[http://127.0.0.1:8888/contract/addLiquidity](http://127.0.0.1:8888/contract/quote)

Des : tokenA&tokenB添加流动性

Args:

* tokenA: tokenA地址
* tokenB: tokenB地址
* amountADesired: A的期望数量
* amountBDesired：B的期望数量
* amountAMin: A的最小值
* amountBMin: B最小值
* to: 流动性接收地址
* timeout: 超时时间(ms)
* userID: 用户ID

Return:

* {"code": 200, "msg":"成功", "data": {"tx":tx}}

<http://127.0.0.1:8888/contract/addLiquidityETH>

Des : token&ETH添加流动性

Args:

* token: token地址
* amountTokenDesired: token的期望数量
* amountETHDesired： ETH的期望数量
* amountTokenMin: token的最小值
* amountETHMin: ETH最小值
* to: 流动性接收地址
* timeout: 超时时间(ms)
* userID: 用户ID

Return:

* {"code": 200, "msg":"成功", "data": {"tx":tx}}

[http://127.0.0.1:8888/contract/swapExactTokensForTokens](http://127.0.0.1:8888/contract/addLiquidityETH)

Des :用精确数量的 tokenA换尽可能多的tokenB

Args:

* amountIn： tokenA的精确数量
* amountOutMin: tokenB的最小值
* path: 兑换路径数组
* to: 接收地址
* timeout: 超时时间(ms)
* userID: 用户ID

Return:

* {"code": 200, "msg":"成功", "data": {"tx":tx}}

[http://127.0.0.1:8888/contract/swapTokensForExactTokens](http://127.0.0.1:8888/contract/addLiquidityETH)

Des :用尽可能少的 tokenA换精确数量的tokenB

Args:

* amountOut： tokenB的精确数量
* amountInMax: tokenA的最大值
* path: 兑换路径数组
* to: 接收地址
* timeout: 超时时间(ms)
* userID: 用户ID

Return:

* {"code": 200, "msg":"成功", "data": {"tx":tx}}

[http://127.0.0.1:8888/contract/swapTokensForExactETH](http://127.0.0.1:8888/contract/addLiquidityETH)

Des :用尽可能少的 token换精确数量的ETH

Args:

* amountOut： ETH的精确数量
* amountInMax: token的最大值
* path: 兑换路径数组
* to: 接收地址
* timeout: 超时时间(ms)
* userID: 用户ID

Return:

* {"code": 200, "msg":"成功", "data": {"tx":tx}}

[http://127.0.0.1:8888/contract/swapExactTokensForETH](http://127.0.0.1:8888/contract/addLiquidityETH)

Des :用精确数量的 token换尽可能多的ETH

Args:

* amountIn： token的精确数量
* amountOutMin: ETH的最小值
* path: 兑换路径数组
* to: 接收地址
* timeout: 超时时间(ms)
* userID: 用户ID

Return:

* {"code": 200, "msg":"成功", "data": {"tx":tx}}

[http://127.0.0.1:8888/contract/swapETHForExactTokens](http://127.0.0.1:8888/contract/addLiquidityETH)

Des :用尽可能少的ETH换尽可能多的token

Args:

* amountIn： ETH的数量
* amountOut: token的精确数量
* path: 兑换路径数组
* to: 接收地址
* timeout: 超时时间(ms)
* userID: 用户ID

Return:

* {"code": 200, "msg":"成功", "data": {"tx":tx}}

[http://127.0.0.1:8888/contract/removeLiquidity](http://127.0.0.1:8888/contract/quote)

Des : tokenA&tokenB移除流动性

Args:

* tokenA: tokenA地址
* tokenB: tokenB地址
* liquidity: 要移除的流动性数量
* amountAMin: A的最小值
* amountBMin: B的最小值
* to: 接收地址
* timeout: 超时时间(ms)
* userID: 用户ID

Return:

* {"code": 200, "msg":"成功", "data": {"tx":tx}}

[http://127.0.0.1:8888/contract/removeLiquidityETH](http://127.0.0.1:8888/contract/addLiquidityETH)

Des : token&ETH移除流动性

Args:

* token: token地址
* liquidity: 要移除的流动性数量
* amountTokenMin: token的最小值
* amountETHMin: ETH最小值
* to: 流动性接收地址
* timeout: 超时时间(ms)
* userID: 用户ID

Return:

* {"code": 200, "msg":"成功", "data": {"tx":tx}}