# libatsredis介绍

## 结构体：

typedef struct TrainsInfo

{

char group[30]; //车站组代码

int trainsno; //车组号

int srvno; //服务号

int direction; //方向（上行为0，下行为1）

int state; //状态

int arrivet; //本站到达时间（时间戳）

int departuret; //本站出发时间

}TrainsInfo;

## 动态库类：

#ifdef \_\_unix

class CTrainDataOpt

#else

class DLLFUN CTrainDataOpt

#endif

{

private:

redisContext\* \_conn\_write;

struct timeval tvs;

int use\_tvs;

char \_conn\_write\_ip[128];

int \_conn\_write\_port;

char \_conn\_write\_pw[128];

int reply\_num;

bool CreatConnectWrite(const char\* ip, int port);

bool CloseConnectWrite();

int ReconnectWrite();

bool SetTimeout\_Redis(long sec, long usec);

public:

CTrainDataOpt();

~CTrainDataOpt();

//connect,timeout...

bool SetSeverAddr(char \* ip, int port);//将连接redis的ip号和端口号保存，并创建连接；

//function

int PutTrainsInfo(TrainsInfo ATS\_INFO);//将结构体中的数据存入redis中，PutTrainsInfo接口中有重连功能，当连接中断时会根据SetSeverAddr设置的ip和端口号进行一次重连；

};

#endif

## 使用方式：

例如：

先创建连接：

CTrainDataOpt \* redis\_ats;

redis\_ats = new CTrainDataOpt;

redis\_ats->SetSeverAddr("127.0.0.1",6379);

结构体存入值：

TrainsInfo redisdata;

memset(&redisdata, 0, sizeof(TrainsInfo));

strcpy(redisdata.group,"AAA");

redisdata.trainsno = 1001;

redisdata.srvno = 1001;

redisdata.direction = 0;

redisdata.state = 1;

redisdata.arrivet = 1603696018;

redisdata.departuret = 1603696068;

存入redis（返回值0为存入失败，1为成功，-1为redis没有连上或阻塞）

int ret = redis\_ats->PutTrainsInfo(redisdata);