```
#ifndef MSNetLoopRunnableHEADER
#define MSNetLoopRunnableHEADER

#include <MSNet/MSNetLoop.H>
#include <string>
#include <future>
#include <memory>


namespace fusionbase {

class MSNetLoopRunnable
{
public:
  MSNetLoopRunnable() = default;
  explicit MSNetLoopRunnable(const MSNetLoop& netLoop_):
    _netLoop(netLoop_)
  {
  }
   virtual ~MSNetLoopRunnable(){}

   void setLaunchMode(std::launch launchMode_)
   {
      _launchMode = launchMode_;
   }
   std::launch getLaunchMode() const
   {
     return _launchMode;
   }
   virtual std::string getId() const{ return "";}
   void operator()()
   {
    _netLoop.loop();
   }
protected:
  MSNetLoop     _netLoop;
  std::launch   _launchMode{std::launch::async};
};

typedef std::shared_ptr<MSNetLoopRunnable> MSNetLoopRunnablePtr;

};// namespace fusionbase



#e

#ifndef MSNetLoopsRunnerHEADER
#define MSNetLoopsRunnerHEADER

#include <fusionbase/MSNetLoopRunnable.H>
#include <MSLog/MSLog.H>
#include <vector>


namespace fusionbase {
/*
 * Adding more than 1 runnable with std::launch::deferred most likely mistake.
 * It will block on wait for first runnable
 */
class MSNetLoopsRunner
{
public:
  void add(MSNetLoopRunnablePtr runnable_)
  {
    _runnables.emplace_back(std::move(runnable_));
```

```
67      }
68      void run()
69      {
70        const size_t size = _runnables.size();
71        std::vector<std::future<void> > futures(size);
72        for(size_t pos =0; pos < size; ++pos)// run all
73        {
74          futures.at(pos) = std::async(_runnables.at(pos)->getLaunchMode(),
          *(_runnables.at(pos)) );
75        }
76
77        for(size_t pos =0; pos < size; ++pos)
78        {
79          try
80          {
81            futures.at(pos).wait();
82          }
83          catch(const std::exception& e_)
84          {
85            MSlogError <<"Exception "<< e_.what() << " while running runnable " <<
            _runnables.at(pos)->getId()  << send;;
86          }
87        }
88
89      }
90
91
92
93    private:
94      std::vector<MSNetLoopRunnablePtr> _runnables;
95    };
96
97
98
99
100   };// namespace fusionbase
101
102
103
104   #endif // MSNetLoopsRunnerHEADER
```