# 原理

1. ioloop是tornado的核心
2. ioloop封装了操作管理epoll的工作
3. 当ioloop实例启动时，ioloop将服务器监听的socket添加到epoll容器中，然后循环等待epoll返回可处理的socket
4. 当有客户端发起连接后，ioloop从epoll容器中拿到了服务器监听的socket，并调用服务器处理该监听socket的方法，接收连接请求，并将新的与客户端连接的socket添加到epoll容器中，然后，循环等待epoll返回可处理的socket
5. 当客户端发来请求数据后，ioloop从epoll中拿到了接收数据的socket，并调用服务器处理实例处理该传输的 socket方法，从socket中读取出http报文，解析后通过apllaction的实例方法，进行路由分发，实例化具体的requestHandler，执行其中的http方法，生成相应数据并打包报文写入缓冲区中
6. 当与客户端对应的socket可写时，ioloop从中拿到对应可写的socket，将缓冲区中对应的响应报文数据，写入到socket传回客户端，完成请求处理。

tornado本质上是一个模式，非async函数不要做阻塞操作

tornado使用同步代码会阻塞程序，典型time.sleep()