SSL server 代码流程

```
// 1. 初始化SSL 环境 context
 /* SSL 库初始化 */
 SSL_library_init();
 /* 载入所有 SSL 算法 */
 OpenSSL_add_all_algorithms();
 /* 载入所有 SSL 错误消息 */
 SSL_load_error_strings();
 /* 以 SSL V2 和 V3 标准兼容方式产生一个 SSL_CTX ，即 SSL Content Text */
 ctx = SSL_CTX_new(SSLv23_server_method());
  if(!ctx)
  {
  ERR_print_errors_fp(stdout);
  return -1;
  }

// 服务端 创建一个socket 监听
// 2 .socket 创建
fd = socket(AF_INET,SOCK_STREAM,0);
// 3 .bind socket 绑定服务端地址
  bzero(&saddr, sizeof(saddr));
  saddr.sin_family = PF_INET;
  saddr.sin_port = htons(s->port);
  saddr.sin_addr.s_addr = s->ip.size() <= 0 ? INADDR_ANY : inet_addr(s->ip.c_str());

  if(bind(s->fd,(struct sockaddr *)&saddr,sizeof(struct sockaddr)) < 0)
  {
  perror(" Bind :");
  return -1;
  }
// 4 .listen 开始监听
  if(listen(s->fd,max_client) < 0)
  {
  perror(" Listen :");
  return -2;
  }
// 5 .accept 等待客户端连接
  bzero(&saddr, sizeof(saddr)
  client_fd = accept(s->fd,(struct sockaddr *)&saddr,&len);
```

```
 if(client_fd < 0
 {
 perror(" Accpet :");
 return -1;
}

// 6. 客户端连接上来，创建一个ssl 会话。
 /* 基于 ctx 产生一个新的 SSL */
 if(!sl && s->ctx)
 sl = SSL_new(s->ctx);

 /* 将连接用户的 socket 加入到 SSL */
 if(client->fd < 0)
 return -1;

 SSL_set_fd(sl,client->fd);

 client->ssl = sl;
 clients.push_back(client);

 cout << " SSL_accept ... \n";
 /* 建立 SSL 连接 主要是握手操作*/
 if ((ret = SSL_accept(sl)) == -1)
 {
 // perror("accept");
 cout << "SSL_accept failed,ret " << ret << endl;
 Close();
 return -1;
 }
// 7. 开始SSL 通信
len = SSL_read(ssl, tempBuf, BUFSIZ);
retLen = SSL_write(ssl, pBuf + sendLen, bufLen);
```