《미들웨어》채팅시스템

버전：2.6

# 시스템묘사

채팅시스템은 종합메세지, 전서버메시지, 길드메시지, 사적메시지를 포함한다. 채팅메시지는 시간순으로 표시된다. 유저의발신메시지는 민감메시지필터링에걸러질수있다. 내용문자열은 제한이있다. 각발송간의 시간제한이있다.

# 시스템설계

ChatService 기본클래스를 사용해서 채팅메시지를 조작하고, 발송,송신, 금칙어 필터링을 수행한다.

1. **메시지발송**

Send메서드：모든서버를향해 메시지를 전송함，전섭의 캐쉬에 보존됨 ChatCacheSet내부에；

SendWhisper메서드: 유저에게 귓속말 메시지를 보냄

Receive메서드：유저와 이미받은버전ID를 근거로 ChatCacheSet 에서 걸러낸다.

# 사용실례

1. ChatService 기본클래스를 상속해서 DemoChatService 클래스를 새로만든다.

|  |
| --- |
| public class DemoChatService : ChatService  {  private readonly GameUser \_user;  private const int MsgTimeOut = 0;//分钟  private const int IntervalSend = 10;  private SensitiveWordService \_wordServer;  public DemoChatService()  : this(new GameUser())  {  }  public DemoChatService(GameUser user)  : base(user.UserId.ToInt())  {  \_wordServer = new SensitiveWordService();  this.\_user = user;  }  public static bool IsAllow(GameUser user, ChatType chatType)  {  if (user != null)  {  if (chatType == ChatType.World && (DateTime.Now - user.Property.ChatDate).TotalSeconds >= IntervalSend)  {  return true;  }  else if (chatType != ChatType.World)  {  return true;  }  }  return false;  }  public void SystemSend(ChatType chatType, string content)  {  var chat = new ChatData();  chat .Version = NextVersion  Send(chat);  }  public void Send(ChatType chatType, string content)  {  var chat = new ChatData();  chat .Version = NextVersion  \_user.Property.ChatDate = DateTime.Now;  Send(chat);  }  public void SendWhisper(GameUser toUser, string content)  {  if (\_user == null || toUser == null)  {  throw new Exception("发送人或接收人为空值");  }  \_user.Property.ChatDate = DateTime.Now;    var chat = new ChatData();  chat .Version = NextVersion  SendWhisper(toUser.UserId.ToInt(), chat);  }  protected override List<ChatMessage> GetRange(List<ChatMessage> msgList)  {  if (msgList.Count > 50)  {  int pageCount;  return msgList.GetPaging(1, 50, out pageCount);  }  return msgList;  }  protected override bool HasReceive(ChatMessage message)  {  var m = message as ChatData;  return m != null && m.Version > \_user.Property.ChatVesion; //&& m.SendDate.AddMinutes(MsgTimeOut) > DateTime.Now;  }  protected override string FilterMessage(string message)  {  return \_wordServer.Filter(message);  }  protected override void WriteLog(ChatMessage message)  {    }  } |

1. 처리발송（Python스크립트）

|  |
| --- |
| def takeAction(urlParam, parent):  user = parent.Current.User;  chatService =DemoChatService(user)  chatService.Send(ChatType.Whisper,chatinfo.Id, chatinfo.Content)  return |

1. 처리수신（Python스크립트）

|  |
| --- |
| def takeAction(urlParam, parent):  user = parent.Current.User;  chatService = DemoChatService(user)  actionResult.ChatMaxNum = chatService.CurrVersion  chatList = chatService.Receive()  return |