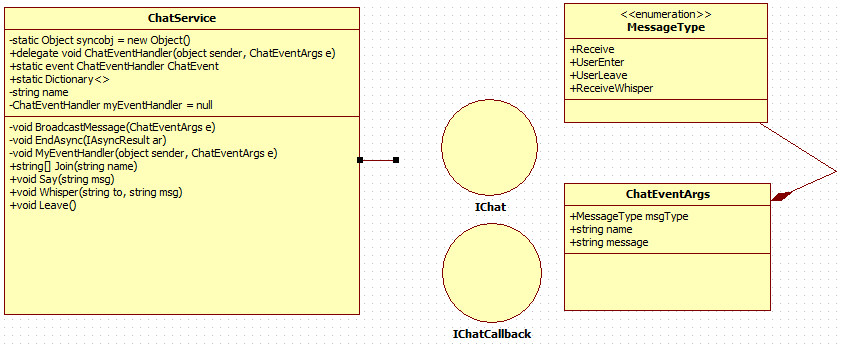
[WCF를 이용한 채팅 프로그램]

1. 제목 : ChatService

2. 설계



3. 설명

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Private | 함수 | 리턴값 | | 인자 | 기능 |
|  | BroadcastMessage | Void | ChatEventArgs e | | 임시 이벤트 변수 temp에 ChatEvent 대입.  temp가 null이 아닐경우 Temp의 모든 클라이언트에게 이벤트 전달(EndAsync 호출) |
|  | EndAsync | Void | IAsyncResult ar | | ChatEventHandler 변수 생성.  인자의 비동기 호출이 호출된  델리게이트 개체 대입. |
|  | MyEventHandler | Void | Object sender,  ChatEventArgs e | | 인자로 들어온 이벤트 스위칭.  상황에 맞게 클라이언트에게  전달할 IChatCallback 메서드 호출. |

|  |  |  |
| --- | --- | --- |
| Interface | IChat | IChatCallback |
| 기능 | 메시지관련 Contract Interface  (클라이언트->서버) | 클라이언트에 콜백할  CallBack Contract  (서버->클라이언트) |
|  | Join문 | Receive문 |
|  | Say문 | ReceiveWhisper문 |
|  | Whisper문 | UserEnter문 |
|  | Leave문 | UserLeave문 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IChat | 함수명 | 리턴값 | 인자 | 기능 |
|  | Join | String[] | String name | 이름이 기존 채터에 존재 검색.  이름, 이벤트 추가.  사용자에게 보내줄 채널 설정.  UserEnter라는 이벤트 전달.  델리게이터 추가.  사용자 리스트를 보내줌.  (예외 : 사용자가 이미 사용하는 이름인 경우에는 null값 리턴) |
|  | Say | Void | String msg | ChatEventArgs 객체 생성.  이름과 메시지와  메시지타입(enum) 대입형성.  브로드캐스트메시지로 호출. |
|  | Whisper | Void | String to, string msg | ChatEventArgs 객체 생성.  이름과 메시지와  메시지타입(enum) 대입형성.  채터 목록에서 검색후 EndAsync 호출 |
|  | Leave | void |  | 채터목록에서 대상 삭제.  델리게이터 감소.  ChatEventArgs 객체 생성.  이름과 메시지와  메시지타입(enum) 대입형성.  브로드캐스트메시지로 호출. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IChatCallback | 함수명 | 리턴값 | 인자 | 기능 |
|  | Receive | Void | String sendername,  String message | 특정사용자와 특정메세지를 모든 클라이언트에게 보내준다 |
|  | ReceiveWhisper | void | String sendername,  String message | 특정사용자와 특정메세지를 지정한 클라이언트 보내준다 |
|  | UserEnter | Void | String name | 클라이언트가 자신이 입력한 이름으로 접속하여 채터 목록에 올라간다. |
|  | UserLeave | void | String name | 접속중이던 사용자가 나가고 채터목록에서 삭제된뒤, 남아있는 클라이언트들에게 귓속말로 알려짐. |