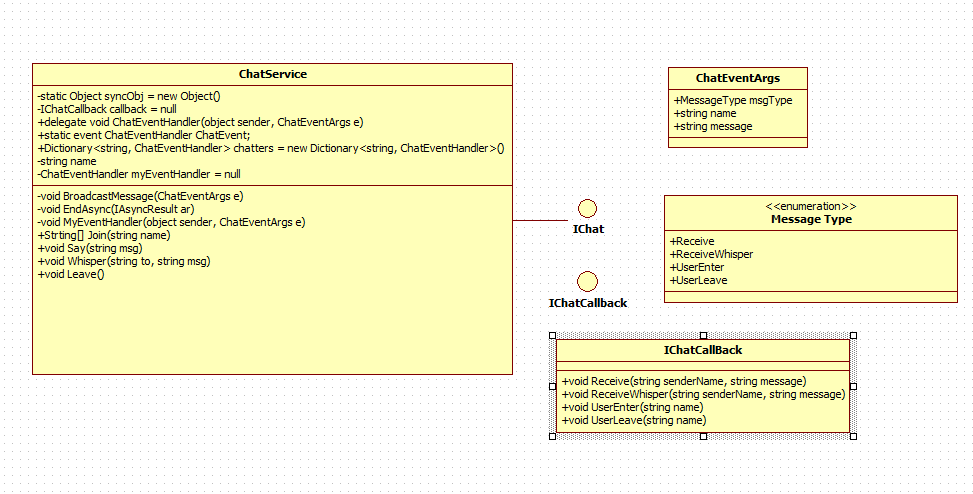
주제 : 채팅 서비스 구현

1. 인터페이스 설계



Void BroadcastMessage(ChatEventars e)

|  |  |  |
| --- | --- | --- |
| 리턴 | Void | 없음 |
| 인자 | ChatEventars e | 현재 클라이언트들에게 이벤트를 전달 |
| 기능 | 임시 이벤트 변수 temp에 ChatEvent 대입  temp가 null이 아닐경우 Temp의 전체 클라이언트에게 이벤트 전달  (EndAsync 호출) | |

Void EndAsync(IAsyncResult ar)

|  |  |  |
| --- | --- | --- |
| 리턴 | Void | 없음 |
| 인자 | IAsyncRsult | 비동기 작업의 상태를 받는 인자 |
| 기능 | EndAsync | ChatEventHandler 변수 생성후  인자의 비동기 호출이 호출된 델리게이트 개체 대입 |

void MyEventHandler (object sender, ChatEventars e);

|  |  |  |
| --- | --- | --- |
| 리턴 | void |  |
| 인자 | Object sender  CharEventars e | 샌더, 이벤트 |
| 기능 | MyEventHandler | 인자로 들어온 이벤트 스위칭  상황에 맞게 클라이언트에게 전달할 IChatCallback 메서드 호출 |

String[] Join(string name);

|  |  |  |
| --- | --- | --- |
| 리턴 | String | 사용자 리스트 리턴 |
| 인자 | String name | 접속한 사람의 이름 |
| 기능 | Join | 이름과 이벤트를 추가  사용자에게 보내 줄 채널을 설정.  UserEnter 라는 이벤트를 전달  델리게이터 추가, 사용자리스트를 보냄. |

Void say(string msg)

|  |  |  |
| --- | --- | --- |
| 리턴 | Void |  |
| 인자 | String | 사용자가 입력한 메세지 |
| 기능 | say | ChatEventArgs 객체를 생성하고 리시브 메시지 타입, 사용자의 이름, 메시지내용을 대입 BroadcastMessage 함수를 호출 |

Void whisper(string to, string msg)

|  |  |  |
| --- | --- | --- |
| 리턴 | void |  |
| 인자 | String to string msg | 귓속말을 보낼 대상의 이름 메시지 |
| 기능 | whisper | ChatEventArgs 객체를 생성, 귓속말 메시지 타입, 사용자의 이름, 메시지내용을 대입.  Dictionary에 포함된 사용자중 귓속말을 보낼 대상을 찾아 delegate ChatEventHandler를 이용해 BeginInvoke. |

Void Leave()

|  |  |  |
| --- | --- | --- |
| 리턴 | void |  |
| 인자 |  |  |
| 기능 | Leave | 접속자 목록에서 대상 삭제  델리게이터 감소  ChatEventArgs 객체 생성  사용자의 이름, 메시지 타입 대입  BroadcastMessage 호출 |