채팅 서비스 구현

1. 인터페이스 설계



1. 세부기능 설명
2. 인터페이스 이름

IChat

1. 기능

|  |  |  |  |
| --- | --- | --- | --- |
| 함수명 | 리턴값 | 인자 | 설명 |
| BroadcastMessage | void | ChatEventArgs e | 1. ChatEvent를 ChatEventHandler타입의temp에 대입  2.temp가 null이 아닐경우 temp의 모든 클라이언트에 이벤트를 전달(EndAsync에 이벤트 전달, 비동기호출) |
| EndAsync | void | IAsyncResult ar | 1.전달받은 비동기호출이 호출된 이벤트를 ChatEventHandler타입의 d변수에 대입  2.에러시 이벤트에서 비동기 호출이 호출된 이벤트 종료 |
| MyEventHandler | void | object sender, ChatEventArgs e | 1.이벤트의 메시지 타입에따라 클라이언트에 전송  2.에러가 발생했을 경우 Leave |
| Join | string[] | string name | 1.이름이 기존 채터에 있는지 검색한다.  2.이름과 이벤트를 추가한다.  3.사용자에게 보내 줄 채널을 설정한다.  4.UserEnter 라는 이벤트를 전달한다  5.델리게이터 추가  6.사용자리스트를 보내준다.  7.같은 이름이 없을 경우에는 사용자 리스트를 반환한다  8이미 사용자가 사용하고 있는 이름일 경우 리턴 |
| Say | void | string msg | 1.메시지타입 Receive와 name, msg를 ChatEventArgs의 변수 e에 대입  2.BroadcastMessage에 e 전달 |
| Whisper | void | string to, string msg | 1.메시지타입 ReceiveWhisper와 name, msg를 ChatEventArgs의 변수 e에 대입  2.BroadcastMessage에 e 전달  3.귓속말 대상을 받아 클라이언트에 이벤트를 전달(EndAsync에 이벤트 전달, 비동기호출) |
| Leave | void |  | 1.name 이 null 값이면 리턴  2.아니면 name을 제거  3.ChatEvent 에서 myEventHandler 제거  4.메시지타입 UserLeave 와 name을 ChatEventArgs의 변수 e에 대입  2.BroadcastMessage에 e 전달 |