# Chat Bot设计文档-Chat Server部分

|  |
| --- |
| Restricted |

|  |  |  |  |
| --- | --- | --- | --- |
| **修订历史** | | | |
| **日期** | **修改者** | **版本号** | **修改描述** |
| 2017/3/21 | Michael | 1.0.0 | 初始版本 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### 总体设计

如果站点开启了Chat-Bot 功能, Chat Server 要做如下处理:

1. Chat Server启动时,从数据库中读取Chat-Bot服务的地址, 并保存在内存变量中.

2. 接受聊天请求时判断是否要启用chat-Bot 聊天

3. Chat-Bot聊天过程中

a) 接收Visitor的消息, 调用Chat-Bot服务把处理结果返回给Visitor,

b) 在数据库中记录每个Question的处理结果(Report 需要使用)

4. 聊天结束后

c) 根据聊天的实际过程, 把Chat 的type 设置为Chat Bot Only Chat/Chat From Chat Bot To Agent.

#### 功能设计

##### 数据结构改动

1. ChatEngine 添加ChatBotServiceUrl属性

public string ChatBotServiceUrl;

ChatBotServiceUrl用于存储ChatBot 服务的URL, 在ChatEngine 初始化的时候赋值.

2.ActiveSite 添加ChatBotActionPendingQueue属性

public PendingQueue<PendingChatBotAction> ChatBotActionPendingQueue;

用于存储Chat-Bot 聊天过程中每个Question的处理结果, 最终队列中的数据会被写入数据库.

其中PendingChatBotAction包括如下属性:

public class PendingChatBotAction

{

public string GUID;

public int ChatId;

public string OriginalQuestion;

public int StandardQuestionId;

public EnumChatBotActionType ActionType;

public float MatchingScore;

public DateTime CreateTime;

}

其中的EnumChatBotActionType 包括如下内容

public enum EnumChatBotActionType

{

HighConfidenceAnwserAndHelpful = 0,

HighConfidenceAnwserAndNoHelpful = 1,

HighConfidenceAnwserAndNoRate = 2,

PossibleAnswer = 3,

NoAnswer = 4

#### }

3. EnumChatStatus 和 EnumVisitorStatus 增加新的枚举

enumChatWithChatBot

##### 程序及接口的改动

###### 2.2.1 ChatServerCore 启动时

1. ChatEngine.Start()中要对ChatBotServiceUrl进行初始化
2. CodePlan 因为增加了ChatBot 相关的配置信息, 相关的代码要进行更新.

###### 2.2.2 接受聊天时

ChatEngineWrap.RequestChat() 方法要做如下处理

1. 先获取CurrentVisitor.Codeplan.ChatbotIfEnable, 判断对应的CodePlan是否开启了ChatBot, 如果未开启处理流程不变.
2. 如果开启了ChatBot功能

a) 手动邀请或Chatbot转Chat With Agent 时不转ChatBot

b) 如果配置的是Only when queue length reaches, 但是Auto Accept 没有开启不转ChatBot

如果Auto Accept是开启的则需要判断队列中的