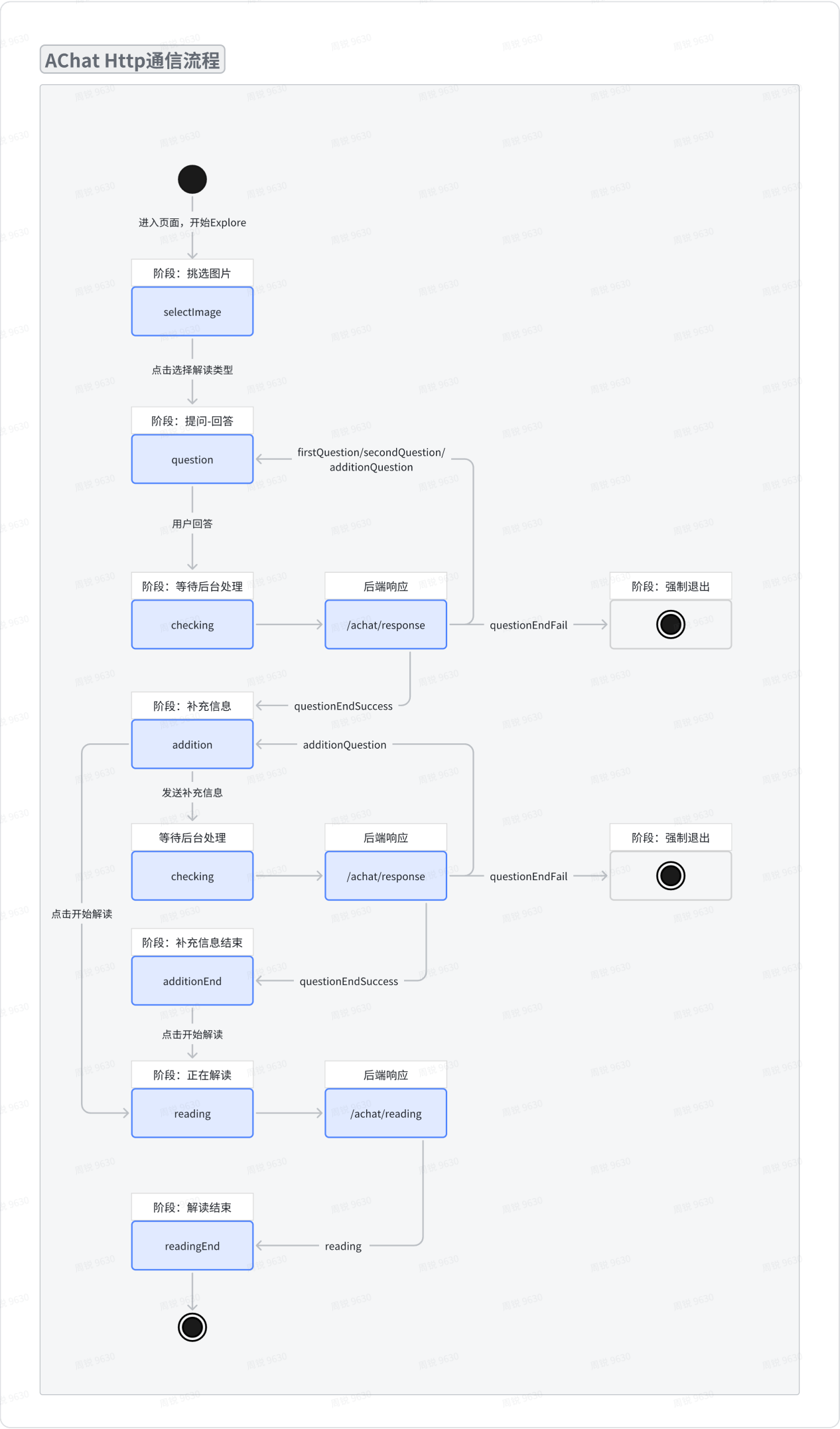


Spirittrue AChat通信流程与注意事项



AChat通信流程

后台完成解读后，把发送引导”到“返回…



陈彩霞 4月22日 10:44

这个 id 应该在 GuideStart 的时候就分配了，后续的每一条消息都标记上这个 id。只是没有 readingStart 就结束的话后台可删除前面的记录（删不删看后台需求，反正客户端不看）



陈彩霞 4月22日 10:47

这个不需要，客户端通过 http 跟后台通信，不会有过期请求一说



周锐 4月22日 10:48

这个有效性是指用户内容是否有效，不是请求过期



陈彩霞 4月22日 10:50

明白了，这个可以，看 ai 那边能不能做



陈彩霞 4月22日 10:50

请求期间，客户端应该是 loading 状态，不许再输入

通信流程较为严格，需要考虑信令发送或…



陈彩霞 4月22日 10:46

这整个流程是 http 吧，那客户端就是监听请求的结果，比聊天消息那种回执简单一些

1. 客户端不需要考虑消息次数，和后台保持“一问一答”，通过“AChatQuestionEnd”和

- 2. 后台完成解读后，把发送引导”到“返回解读”的全流程作为整体存储,为其分配一个Id
- 3. 各消息字段需要附加的信息
 - a. AChatGuideStart => 选中图片链接，选择解读类型
 - b. AChatQuestion => 后台问答文本，是否是最后一条提问，用户上一次回答是否有效
 - c. AChatQuestionResponse => 用户输入文本
 - d. AChatReadingStart =>
 - e. AChatReadingEnd => AI解读内容, success: 允许进入下一步， fail: 多次识别均无效
- 4. 用户如果在AChatReadingStart前退出该页面/强退App,此次AChat服务不算次数，客户无感知
- 5. 客户端向后台发送消息之后，后台需要检测用户发送内容是否为有效内容，期间存在延迟，用户无感知
- 6. 通信流程较为严格，需要考虑信令发送或接收失败的情况，进行重试或让用户重发（类语音消息，点击感叹号可重发）。

A-Chat UI展示阶段定义

```
1 class AChatActivity(Enum):
2     # 查寻解读历史记录
3     history
4     # 挑选图片
5     selectImage
6     # 机器人提问
7     question
8     # 后台正在处理用户消息
9     checking
10    # 用户补充信息
11    addition
12    # 后台正在解读
13    reading
14    # 解读结束
15    readingEnd
16
```

A-Chat 通信间消息类型定义

```
1 class AChatMessageType(Enum):
2     # 机器人提出第一个问题
3     firstQuestion = 0
4     # 机器人提出第二个问题
5     secondQuestion = 1
6     # 校验失败，机器人请求用户重说
7     additionQuestion = 2
8     # 用户回应机器人问题
9     response = 3
10    # 机器人提问结束，进入下一步
11    questionEndSuccess = 4
12    # 机器人提问结束，用户多次回答无效，强制结束流程
13    questionEndFail = 5
14    # 解读消息
15    reading = 6
16    # 打招呼消息
17    greet = 7
18    # 图片消息
19    image = 8
```

```
20 # 解读类型消息
21 readingKind = 9
22 # 用户选择解读类型
23 selectReadingKind = 10
24 # 推荐顾问
25 recommendStar = 11
26 # 机器人正在输入
27 composing = 12
```

API定义

1. Get v2/achat/switch //用户进入或离开

```
1  ### 参数
2  {
3      'type': int # 0: 离开, 1:进入, 2:登录app
4  }
5  ### 返回, 当type==1时存在返回
6  {[
7      {
8          'content':str, # 文本内容
9          'role': str, # 角色类型: enum['assistant','user']
10         'type': int # 消息类型, 7, 打招呼
11     },
12     {
13         'content': str, # 文本
14         'role': str, # 角色类型: enum['assistant','user']
15         'type': int # 消息类型, 8, 图片信息的文本
16     },
17     {
18         'content': str, # 文本
19         'role': str, # 角色类型: enum['assistant','user']
20         'type': int # 消息类型, 9, 选择解读类型的文本
21     }
22 ]}]
23
24
```

2. Post v2/achat/response //用户回答问题

```
1  ### 参数
2  {
3      'text': String, # 用户发送文本
4      'id': int # 当前achat服务id
5  }
6  ### 返回
7  {
8      'content': String, # 机器人发送文本
9      'role': String, # 角色类型: enum['assistant','user']
10     'type': int # 消息类型,此处取值可能为0, 1, 2, 4, 5
11 }
12
13
```

3. Post /v2/achat/reading //开始解读

```
1  ### 参数
2  {
3      'id': int # 当前achat服务id
4  }
5  ### 返回
6  {
7      'content': str, # 机器人解读文本
8      'role': str, # 角色类型: enum['assistant', 'user']
9      'type': int # 消息类型, 此处取值为6
10 }
```

4. Get /v2/achat/history //获取用户所有A-Chat历史记录

```
1  ### 参数
2  {
3      'page': int, # 第几页
4      'countPerPage': int # 每页多少个
5  }
6  ### 返回
7  {
8      'id': int, # 当前achat记录的id
9      'kind': str, # 选择解读类型(全大写)
10     'createTime': int, # 记录解读时间
11     'images':
12     [{
13         image: str,
14         thumbnail: str # 缩略图url
15     }]
16 }
```

5. Get /v2/achat/one //获取用户某一次A-Chat历史记录

```
1  ### 参数
2  {
3      'id': int # 当前achat记录的id
4  }
5  ### 返回
6  {
7      'id': String, # 当前achat记录的id
8      'createTime': int, # 记录解读时间
9      'messages': # 消息列表
10     [{
11         'content': String, # 文本
12         'role': String, # 角色类型['assistant', 'user']
13         'type': int, # 消息类型
14         'recommendStars': [UserInfoModel] # 推荐顾问列表, 当消息类型为7时需要
15         'kind': str, # 选择解读类型, 当消息类型为9时需要
16         'images': # 图片信息, 当消息类型为8时需要
17         [{
18             image: str,
19             thumbnail: str # 缩略图url
20         }]
21     }]
22 }
23 }
```

6. Get v2/achat/images //获取/刷新三张塔罗图片

```
1  ### 参数
2
3  ### 返回
4  {
5      'images': # 图片信息
6      [{
7          'image': str,
8          'thumbnail': str # 缩略图url
9      }]
10 }
```

7. Get /user/me

```
1  ### 增加字段
2
3  # A-Chat解析剩余次数
4  aiChatChance: int
5
6  # A-Chat离开聊天室剩余次数
7  aiLeaveChance: int
8
9  # 用户有A-Chat解析历史记录
10 hasAIChatHistory: bool
11
12 # A-Chat下一次可用时间的时间戳
13 aiNextAvailableTime: int
```

8. Get /v2/recommend/achat/stars

```
1  ### 参数
2
3  ### 返回
4  {
5      'images': # 图片信息
6      [{
7          'image': str,
8          'thumbnail': str # 缩略图url
9      }]
10 }
```

9. Get /v2/achat/create

```
1  ### 参数
2  {
3      'images':
4      [{
5          'image': str,
6          'thumbnail': str # 缩略图url
7      }]
8      'kind': str, # 选择解读类型
9  }
10 ### 返回
```

```
11 {
12     'id': int, # 当前achat记录的id
13     'createTime': int, # 记录解读时间
14     'messages': # 消息列表
15     [{
16         'content': str, # 文本
17         'role': str, # 角色类型['assistant', 'user']
18         'type': int # 消息类型
19     }]
20 }
```