

header	body	備註
add_friend_request	<pre>{ selfs_ID:(string), user_name:(string), profile_picture: { file_type:(string), binary:(string) }, profile_context:(string) latest_edit_time 這個暫時先刪掉 }</pre>	這裡的 latest_edit_time 是專門放，送出邀請的時間
add_friend_reply	<pre>{ selfs_ID:(string), response:(boolean) }</pre>	
update_personal_data	<pre>{ personal_data_list: [ID:(string), user_name:(string), profile_picture: { file_type:(string), binary:(string) }, profile_context:(string)], background_picture: { file_type:(string), binary:(string) }, }</pre>	ID 就是使用者的 ID

	<pre> latest_edit_time:(string)] } </pre>	
post_article	<pre> { article_list: [{ latest_edit_time:(string), article_ID:(string), owner_ID:(string), parent_ID:(string), root_article_ID:(string), content:(string), image_content: [{ file_type:(string), binary:(string) }], like_list:[(string)uuid list], position_tag:[(string)], friend_tag:[(string)], deletion:(boolean) }] } </pre>	<p>接收方需要檢查該文章是否存在 如果存在就直接複寫</p> <p>//如果要 deletion == true 就把其他的欄位設成""</p>

	}	
like_up	{ article_ID:(string), self_ID:(string), latest_edit_time:(string) }	latest_edit_time 要更新到 article 的 latest_edit_time
sync1_request	{ data_list: [{ ID:(string), latest_edit_time:(string), current_ID_type:(string) }] }	當 A 跟 B 要求新的資料的時候 A 會傳這個給 B(順便包自己有的) 這裡的 ID 可以是個人資料、文章、貼文的 sync1_request 跟 sync1_response 在同一條連線。 current_ID_type : personal_data, article, comment
sync1_response	{ data_list: [{ ID:(string), latest_edit_time:(string), current_ID_type:(string) }] }	B 收到之後會把新的文章 ID 跟 time 給你
sync2_request	{ data_list: [A 向不同的擁有者要資料

	<pre>{ ID:(string) current_ID_type:(string) }</pre>	<p>//然後資料擁有者 會再向 A 發 post_article 或 update_personal_data</p>
--	---	--

範例傳輸內容

```
{
  "header": "add_friend_request",
  "body": {
    "self_ID": "dc7bafa6-391c-473d-b70b-4e92c26f9e04",
    "user_name": "testName",
    "profile_picture": "binaryImage",
    "profile_context": "你好我是菜雞",
    "last_edit_time": "2019.11.29.24.00",

    "latest_edit_time": "2019.11.29.23.59"
  }
}
```