串點移動DEMO - 社群後台

API路徑前綴:https://localhost:5001/api/

備註

- 只有 Archive = 0 的資料會被搜尋出來
- 使用者只能操作自己的貼文與留言內容
- 因 DEMO API 未實作 Token 之類的身分驗證機制，故暫以建構函式中的 userId 模擬從 Middleware 解出來的使用者資訊，userId 以信任資訊無誤的前提，在controller中進行操作
- 異動資料時，Admin_Id欄位會記錄下該次資料異動者的User_Id，同時將UpdateTime更新為當下
- 刪除資料時，不會真的進行 DELETE動作，而是將 Archive 狀態調為1
- PUT、DELETE API，QueryString 與 Request.Body 中皆有 Id 參數，是為了 Double check 前端所帶參數的正確性

新增貼文POST:post

Request

{

“Content”: string,

“PrivacyType_Id”: int

}

Response

{

“StatusCode”: 0

“Message”: 成功

“Data”: null

}

PrivacyType = 1，為公開發佈，所有人可見
PrivacyType = 2，為私人發佈，不會出現在公開貼文列表中

修改貼文PUT:post/:id

Request

{

“Id”: int,

“Content”: string,

“PrivacyType_Id”: int

}

Response

{

“StatusCode”: 0

“Message”: 成功

“Data”: null

}

刪除貼文DELETE:post/:id

Request

{

“Id”: int

}

Response

{

“StatusCode”: 0

“Message”: 成功

“Data”: null

}

新增留言POST:comment

Request

{

“Post_Id”: int

“Parent_Comment_Id”: int

“Content”: string

}

Response

{

“StatusCode”: 0

“Message”: 成功

“Data”: null

}

父留言(文章留言): Parent_Comment_Id = 0
子留言(回覆他人的文章留言): Parent_Comment_Id = 回覆對象的Comment_Id

修改留言PUT:comment/:id

Request

{

“Id”: int,

“Content”: string

}

Response

{

“StatusCode”: 0

“Message”: 成功

“Data”: null

}

删除留言DELETE:comment/:id

Request

{

“Id”: int

}

Response

{

“StatusCode”: 0

“Message”: 成功

“Data”: null

}