院方PACS/AI

院方PACS/AI

API

Orthanc

UI

在case介面中，通過socket通知UI進行資料更新

在case介面中，通過socket通知UI進行資料更新

上傳Dicom至PACS中

一筆資料合併成功通知進行PACS上傳

多筆資料等待User手動合併，通知API後進行上傳通知

返回結果

返回合併結果

透過MQ通知Orthanc進行手動合併accNum tag更新

只有一筆資料，透過MQ通知Orthanc進行accNum tag更新

User於取回最新case資料中，手動進行合併，合併確認後，通知API更新資料

返回上傳結果

成功:通知AI執行分析

返回合併結果

返回accNum資料

向醫院查詢accNum資料

UI

Orthanc

UI

在case介面中，通過socket通知UI進行資料更新

是否自動合併，或提供使用者可選Mapping資訊

儲存各Case詳細Dicom資訊

根據成功與否新增Case資料

上傳批次資訊

上傳Dicom影像

回應上傳是否成功

批次資料儲存

[POST] http:/{IP}:{port}/tourCar

Request body:

{

    "job": "測試-002", //批次名稱(會是唯一值)

    "time": "2025/07/07-12: 00: 00", //上傳時間

    "name": "七月", //資料夾名稱

    "series": 5, //Dicom影像數量

    "files": [

{

    "caseName": "caseName2", //case名稱，dicom的檔案名稱，包含身分證名稱等資訊

    "upload": **true**, //是否上傳Orthanc成功

    "patientId": "1234", //Dicom tag中的病人ID(0010,0020 (PatientID))，根據這項值，查詢該系列accNumber

    "studyId": "1.2.840.113817.20240501115500.1200140438396.85189808942",  //Dicom tag中的0020,000d (StudyInstanceUID)，用來生成UUID向Orthanc做查詢

    "seriesId": "1.2.392.200036.9116.2.6.1.3268.2054920286.1579063148.386285", //Dicom tag中0020,000e (SeriesInstanceUID)

    "instancesId": "1.2.392.200036.9116",  //Dicom tag中0008,0018 (SOPInstanceUID)

}

]

}

Response:

成功

{

    "codeStatus": 200, //200成功 404失敗

    "message": "Add new TourCar Record job: 測試-002, 七月", //Log紀錄訊息

    "result": {

        "job": "測試-002",

        "name": "七月",

        "series": 5,

        "time": "2025-07-07T04:00:00.000Z",

        "id": "1",

        "status": "Pending"

    }

}

失敗(job名稱不能重複)

{

    "codeStatus": 404,

    "message": "duplicate job name."

}

取得批次資料

[GET] http:/{IP}:{port}/tourCar

Response:

{

    "codeStatus": 200,

    "result": [

        {

            "id": "1",

            "job": "測試-002",

            "name": "七月",

            "series": 5,

            "status": "Pending",

            "time": "2025-08-01T08:56:06.000Z"

        }

    ]

}

刪除批次資料

[DELETE] http:/{IP}:{port}/tourCar

Request body:

{

    "job": "測試-002"

}

//會刪除job名稱"測試-002"的這筆資料

Response:

成功  
{ "codeStatus": 200, message: "Delete new TourCar Record job: 測試-002. " }

失敗  
{ "codeStatus": 404, message: "{錯誤訊息}" }

新增Case資料在哪個批次下，根據最後job名稱做對應

[POST] http://{IP}:{port}/tourCarCase/{job名稱}

Ex: <http://localhost:8081/tourCarCase/>批次名字  
Request Body:

{

    "caseName": "caseName2", //case名稱，dicom的檔案名稱，包含身分證名稱等資訊

    "upload": **true**, //是否上傳Orthanc成功

    "patientId": "1234", //Dicom tag中的病人ID(0010,0020 (PatientID))，根據這項值，查詢該系列accNumber

    "studyId": "1.2.840.113817.20240501115500.1200140438396.85189808942",  //Dicom tag中的0020,000d (StudyInstanceUID)，用來生成UUID向Orthanc做查詢

    "seriesId": "1.2.392.200036.9116.2.6.1.3268.2054920286.1579063148.386285", //Dicom tag中0020,000e (SeriesInstanceUID)

    "instancesId": "1.2.392.200036.9116",  //Dicom tag中0008,0018 (SOPInstanceUID)

}

Response:

成功

{

    "codeStatus": 200,

    "message": "Add new TourCarCase Record job: 測試-002,caseName: caseName3"

}

失敗 重複caseName

{

    "codeStatus": 404,

    "message": "duplicate caseName."

}

失敗 沒有該筆批次資訊

{

    "codeStatus": 404,

    "message": "Not found this job."

}

根據批次名稱取得Case資料  
[GET] http://{IP}:{port}/tourCarCase/{job名稱}

Response

{

    "codeStatus": 200,

    "result": [

        {

            "id": "1",

            "map\_job": "測試-003",

            "patientId": "1234",

            "seriesId": "1.2.392.200036.9116.2.6.1.3268.2054920286.1579063148.386285",

            "caseName": "caseName",

            "series": 5,

            "status": "Pending",

            "upload": 1, // 0為失敗 1為成功

            "mapping": **null**, //完成mapping後的accNumber

            "postAI": 0, // 0為失敗 1為成功

            "postPACS": 0, // 0為失敗 1為成功

            "accNumbers": "News\_caseName" //(目前為假資料)對應的accNumber資料，可能會有多筆，會需要user於介面中選擇綁定

        }

    ]

}

根據caseName名稱更新Case資料  
[PUT] http://{IP}:{port}/tourCarCase

Request body:

{

    "caseName": "caseName3", //更新caseName3的 資料

    "mapping": "Test\_caseName2" //user於介面中選擇綁定的accessNumber名稱

}

Response:

成功

{

    "codeStatus": 200,

    "message": "Update TourCarCase Record caseName: caseName3.",

    "result": {

        "id": "1",

        "patientId": "1234",

        "seriesId": "1.2.392.200036.9116.2.6.1.3268.2054920286.1579063148.386285",

        "map\_job": "測試-002",

        "caseName": "caseName3",

        "series": 0,

        "status": "Pending",

        "upload": 1,

        "mapping": "Test\_caseName1",

        "postAI": 0,

        "postPACS": 0

    }

}

失敗 沒找到該筆caseName

{

    "codeStatus": 404,

    "message": "Not found this caseName caseName13."

}

根據caseName名稱retry Post to PACS  
[POST] http://{IP}:{port} /tourCarCase/{caseName}/retryPACS

Request body:

{}

Response:

成功

{

    "codeStatus": 200,

    "message": "Retry post To PACS caseName: caseName3.",

    "result": {

        "id": "2",

        "patientId": "1234",

        "studyId": "1.2.840.113817.20240501115500.1200140438396.85189808942",

        "seriesId": "1.2.392.200036.9116.2.6.1.3268.2054920286.1579063148.386285",

        "instancesUUId": "9dc27eb9-8d1e1a32-91bef8dd-55f5f291-f4bfb1e3",

        "map\_job": "測試-002",

        "caseName": "caseName3",

        "series": 5,

        "status": "Pending",

        "upload": 1,

        "mapping": **null**,

        "postAI": 1,

        "postPACS": 2

    }

}

根據caseName名稱retry Post to PACS  
[POST] http://{IP}:{port} /tourCarCase/{caseName}/retryAI

Request body:

{}

Response:

成功

{

    "codeStatus": 200,

    "message": "Retry AI caseName: caseName3.",

    "result": {

        "id": "2",

        "patientId": "1234",

        "studyId": "1.2.840.113817.20240501115500.1200140438396.85189808942",

        "seriesId": "1.2.392.200036.9116.2.6.1.3268.2054920286.1579063148.386285",

        "instancesUUId": "9dc27eb9-8d1e1a32-91bef8dd-55f5f291-f4bfb1e3",

        "map\_job": "測試-002",

        "caseName": "caseName3",

        "series": 5,

        "status": "Pending",

        "upload": 1,

        "mapping": **null**,

        "postAI": 2,

        "postPACS": 2

    }

}

即時更新

目前因為未實際串接院方及Minipacs，暫時以模擬方式透過socket.io送出通知

透過

npm install socket.io-client

                    socket.on("hamster", (data) => {

                        console.log("??:", data);

                        //呼叫進行更新

                    });

目前使用:  
[PUT] /tourCarCase (更新mapping)

[POST] /tourCarCase/:case/retryPACS (模擬retry送給PACS)

[POST] /tourCarCase/:case/retryAI (模擬retry送給AI分析)

會透過socket.io send資訊在hamster這個頻道上