**检 测 报 告**

**Test Report**

**报告编号（Report No.）：**

**产品名称(Product Name)： patahub**

**型 号(Model/Type) ：**

**委 托 方(Client) ：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **检测报告** | | | | | |
| 样品名称：  版 本 号：  样品来源：现场测试  样品检测日期：2022年4月25日-2021年4月28  日 | | | 制造商：  制造商地址： | | |
| 测试单位 |  | | | | |
| 测试地点 |  | | | | |
| 检测依据：  《XX系统XXX模块的XXX代码》 | | | | | |
| 检测方法同标准或技术规范的偏离:  无 | | | | | |
| 检测结论： | | | | | |
| 测试人员 | |  | | 日 期 |  |
| 审核 | |  | | 日 期 |  |
| 签发 | |  | | 日 期 |  |
| 测试小组：    年 月 日 | | | | | |

**测试内容及测试结果**

1、待测代码1：查询操作

1.1 程序描述

此段代码支持根据rcd项的rcd编号、paper编号、dataset编号、codeset编号、result编号综合查询rcd列表

#RCD列表

@app.get("/rcd")

async def get\_rcd\_list(

    rcd\_id: Optional[int] = None,

    result\_id: Optional[int] = None,

    codeset\_id: Optional[int] = None,

    dataset\_id: Optional[int] = None,

    paper\_id: Optional[int] = None ,

    user\_id: Optional[int] = None

):

    rcd = dbop.dbGetRCDList(db, rcd\_id=rcd\_id, result\_id=result\_id, codeset\_id=codeset\_id, dataset\_id=dataset\_id, paper\_id=paper\_id, user\_id=user\_id)

    return rcd

# 综合查询RCD列表

def dbGetRCDList(db, rcd\_id=None, result\_id=None, codeset\_id=None,dataset\_id=None ,paper\_id=None):

    exec = """SELECT \* FROM rcds WHERE true """

    if rcd\_id:

        exec += """and (rcd\_id = %d)"""%(rcd\_id)

    if paper\_id:

        exec += """and (paper\_id = %d)"""%(paper\_id)

    if result\_id:

        exec += """and (result\_id = %d) """%(result\_id)

    if dataset\_id:

        exec += """and (dataset\_id = %d)"""%(dataset\_id)

    if codeset\_id:

        exec += """and (codeset\_id = %d)"""%(codeset\_id)

    result = db.execute(exec)

    return [{'rcd\_id': row[0], 'result\_id': row[1], 'codeset\_id':row[2], 'code\_link':row[3], 'dataset\_id':row[4], 'data\_link':row[5],'paper\_id':row[6]} for row in result]

1.2 程序流程图（一个测试用例可能走过多个子路线）

返回检索结果

不存在

存在

不存在

存在

不存在

不存在

存在

不存在

存在

存在

Codeset\_id

Dataset\_id

Rcd\_id

Paper\_id

Result\_id

输入参数

开始

1.3 分支覆盖测试用例

设计足够多的测试用例，使得被测程序中的每个判断的“真”、“假”分支至少被执行一次。

response = client.get(

            "/rcd",

            params={

                "rcd\_id": 1

            }

        )

        response = client.get(

            "/rcd",

            params={

                "codeset\_id": 1,

                "paper\_id": 1

            }

        )

       response = client.get(

            "/rcd",

            params={

                "dataset\_id": 1,

                "result\_id": 1

            }

        )

        response = client.get(

            "/rcd",

            params={

            }

        )

以上测试应都能正常返回，若查不到对应的rcd则返回空列表，若不带任何参数则缺省返回所有rcd。

2、待测代码2：新增和更新操作

2.1 程序描述

此段程序根据paper\_id的有无，判断新增论文或更新论文信息。

class Paper(BaseModel):

    user\_id: int # 论文的用户

    paper\_name: Optional[str] # 论文标题

    paper\_link: Optional[str] # 论文链接

    paper\_id: Optional[int] # 论文id

    paper\_abstract: Optional[str] # 论文摘要

#新增/更新论文

@app.post("/paper", response\_model=Paper)

async def post\_paper(

    paper\_in: Paper

):

    if not paper\_in.paper\_id:

        # 新建

        if (not paper\_in.paper\_name) or (not paper\_in.paper\_link):

            raise HTTPException(status\_code=422)

        newid = dbop.dbInsertPaper(db, paper\_in.paper\_name, paper\_in.paper\_link, paper\_in.user\_id)

        paper\_in.paper\_id = newid

        return paper\_in

    else:

        # 更新

        dbop.dbUpdatePaper(db, paper\_in.paper\_id, paper\_name=paper\_in.paper\_name, paper\_link=paper\_in.paper\_link)

# 新增论文

def dbInsertPaper(db, title, link, user\_id):

    curresult = db.execute("SELECT paper\_id FROM papers;")

    mxid = 0

    for row in curresult:

        mxid = max(mxid, row[0])

    mxid = mxid + 1

    db.execute("""INSERT INTO papers VALUES

                (?,?,?,?);""", (mxid, title, link, user\_id))

    return mxid

# 更新papers表的paper\_name,paper\_link

def dbUpdatePaper(db,paper\_id,paper\_name=None,paper\_link=None):

    if paper\_name:

        db.execute("""update papers SET  paper\_name=? WHERE paper\_id = ? ;""",(paper\_name,paper\_id))

    if paper\_link:

        db.execute("""update papers SET  paper\_link=? WHERE paper\_id = ? ;""",(paper\_link,paper\_id))

2.2 程序流程图

开始

User\_id存在？

不存在

存在

报错

Paper\_id存在？

存在 不存在

name, link, abstract存在？

Paper\_name, paper\_link, user\_id存在?

都不存在

存在 都存在 否则

修改数据库中id为paper\_id的论文信息

找到数据库中id为paper\_id的论文，然后啥也不做

报错

通过paper\_name，paper\_link等新建paper，并赋id

2.3 分支覆盖测试用例

设计足够多的测试用例，使得被测程序中的每个判断的“真”、“假”分支至少被执行一次。

        response = client.post(

            "/paper",

            json={

                "user\_id": 3,

                "paper\_name": "p4",

                "paper\_link": "pl4"

            }

        )

        response = client.post(

            "/paper",

            json={

                "user\_id": 2,

                "paper\_id": 3,

                "paper\_name": "pp3",

                "paper\_link": "ppl3"

            }

        )

        response = client.post(

            "/paper",

            json={

                "user\_id": 2,

                "paper\_id": 3,

            }

        )

        response = client.post(

            "/paper",

            json={

                "paper\_id": 3,

            }

        )

        response = client.post(

            "/paper",

            json={

                "user\_id": 3,

                "paper\_name": "p4"

            }

        )

以上用例，第一个用例为新增paper项，第二、三个用例为修改paper项，最后两例会报错。

3、待测代码3：删除操作

3.1 程序描述

此段程序通过dataset\_id指定删除数据集

class Dataset(BaseModel):

    user\_id: int #数据集的用户

    dataset\_name: Optional[str] # 数据集名称

    dataset\_link: Optional[str] # 数据集链接

    dataset\_id: Optional[int] # 数据集id

#删除数据集

@app.delete("/dataset")

async def delete\_dataset(

    dataset\_id: int

):

    if len(dbop.dbGetDatasetList(db, dataset\_id)) == 0:

        raise HTTPException(status\_code=404, detail="dataset id not found")

    dbop.dbDeleteDataset(db, dataset\_id)

    dbop.dbDeleteRCD(db, dataset\_id=dataset\_id)

    return dataset\_id

# 删除数据集

def dbDeleteDataset(db, dataset\_id):

    result = db.execute("""DELETE FROM datasets WHERE dataset\_id = ?;""", (dataset\_id,))

    return result

3.2 程序流程图

通过dataset\_id查询对应的dataset

开始

不存在

存在

返回404错误

删除dataset和对应rcd项

3.3 分支覆盖测试用例

设计足够多的测试用例，使得被测程序中的每个判断的“真”、“假”分支至少被执行一次。

        response = client.delete(

            "/dataset",

            params={

                "dataset\_id": 2

            }

        )

        assert response.status\_code == 200

        response = client.delete(

            "/dataset",

            params={

                "dataset\_id": 10000

            }

        )

        assert response.status\_code == 404

        assert response.json() == {

            "detail": "dataset id not found"

        }

以上用例中，第一个用例正常返回，数据库应已删除dataset\_id为2的数据集；第二个用例报错，因为尚不存在编号为10000的数据集。