Ⅰ. 接口编写

/web/start/all

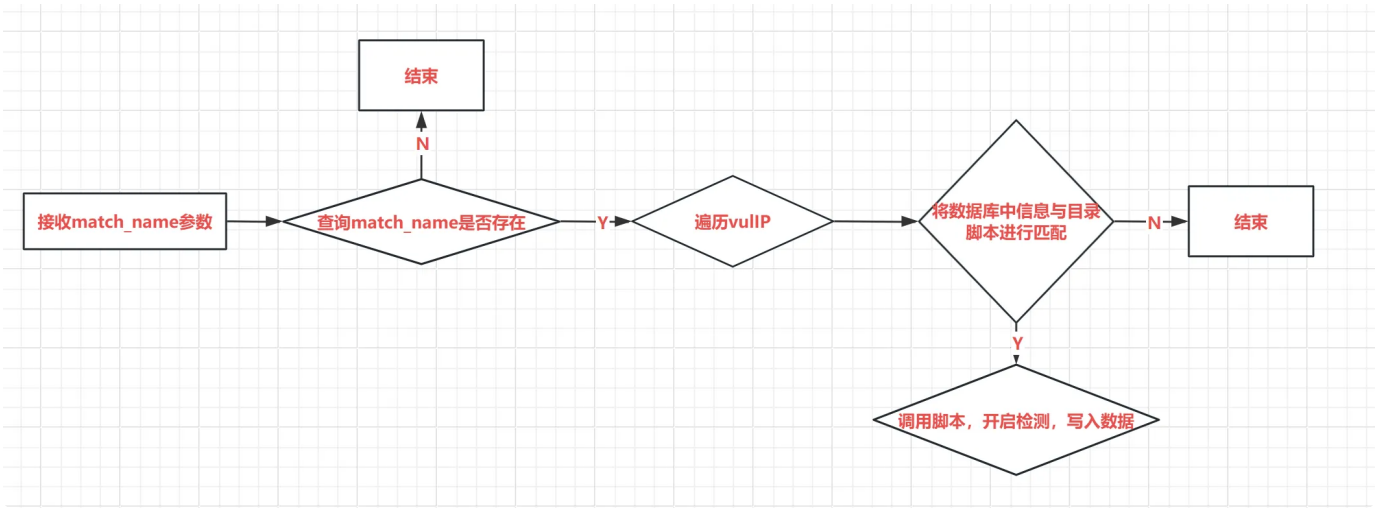
/web/start/all

接口说明：批量开启检测脚本

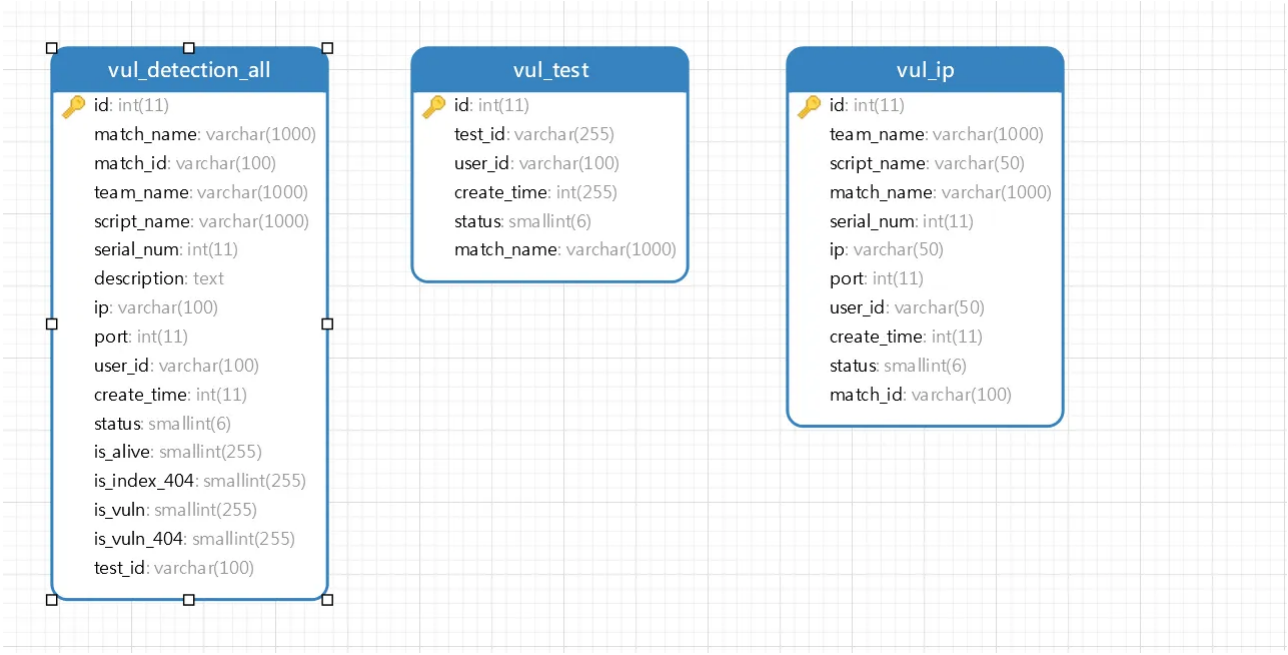
提交数据结构：

```
{
    "match_name": "测试-7questions"
}
```

编写脚本的流程：



数据建模：



在创建一次查询的时候，会将此次查询的的记录写入vul_test,将查询的具体结果写入vul_detection_all

vul_test建模

```python
class VulTest(Base):
    __tablename__ = 'vul_test'
    id = Column(Integer, primary_key=True)
    test_id = Column(String)
    match_name = Column(String)
    user_id = Column(String)

    def to_dict(self):...

    @staticmethod
    def add(vul_test:dict):
        with db.auto_commit():
            vulTest = VulTest()
            vulTest.test_id = vul_test.get('test_id')
            vulTest.match_name = vul_test.get('match_name')
            vulTest.user_id = vul_test.get('user_id')
            db.session.add(vulTest)
```

vul_detection_all建模

```python
class VulDetectionAll(Base):
    __tablename__ = 'vul_detection_all'
    id = Column(Integer, primary_key=True)
    team_name = Column(String)
    match_name = Column(String)
    match_id = Column(String)
    script_name = Column(String)
    serial_num = Column(Integer)
    ip = Column(String)
    port = Column(Integer)
    description =Column(String)
    user_id = Column(String)
    is_alive = Column(SmallInteger)
    is_index_404 = Column(SmallInteger)
    is_vuln = Column(SmallInteger)
    is_vuln_404 = Column(SmallInteger)
    test_id = Column(String)

    def to_dict(self):...

    @staticmethod
    def add(vul_detect_all:dict):
        with db.auto_commit():
            vulDetectionAll = VulDetectionAll()
            vulDetectionAll.team_name = vul_detect_all.get('team_name')
            vulDetectionAll.match_name = vul_detect_all.get('match_name')
            vulDetectionAll.match_id = vul_detect_all.get('match_id')
            vulDetectionAll.script_name = vul_detect_all.get('script_name')
            vulDetectionAll.serial_num = vul_detect_all.get('serial_num')
            vulDetectionAll.ip = vul_detect_all.get('ip')
            vulDetectionAll.port = vul_detect_all.get('port')
            vulDetectionAll.description = vul_detect_all.get('description')
            vulDetectionAll.is_alive = vul_detect_all.get('is_alive')
            vulDetectionAll.is_index_404 = vul_detect_all.get('is_index_404')
            vulDetectionAll.is_vuln = vul_detect_all.get('is_vuln')
            vulDetectionAll.is_vuln_404 = vul_detect_all.get('is_vuln_404')
            vulDetectionAll.test_id = vul_detect_all.get('test_id')
            db.session.add(vulDetectionAll)
```

视图函数：

```python
@app.route('/web/start/all', methods=['POST'])
def web_start_all():
    data = request.json
    match_name = data.get('match_name')
    vulIps = VulIp.query.filter_by(match_name=match_name).all()
    if vulIps is None:
        return {'code':200, 'status':1, 'msg':'参数有误'}
    filenames = get_all_scripts(app)
    script_names = list(set([vulIp.script_name for vulIp in vulIps]))
    if filenames != script_names:
        return {'code':200, 'status':1, 'msg':'参数有误'}
    test_id = uuid.uuid4()
    result = []
    for vulIp in vulIps:
        vulMsg = VulMsg.query.filter_by(script_name=vulIp.script_name).first()
        vul_point_id = vulMsg.vul_point_id
        ip = vulIp.ip
        port = vulIp.port
        url = generate_url(ip, port)
        script_name = vulIp.script_name
        serial_num = vulIp.serial_num
        vulPointMsg = VulPointMsg.query.filter_by(vul_point_id=vul_point_id, serial_num=serial_num).first()
        print(vulPointMsg)
        if vulPointMsg is None:
            return {'code':200, 'status':1, 'msg':'参数有误'}
        func = importlib.import_module(app.config.get('SCRIPTS_PATH') + script_name)
        if hasattr(func, app.config.get('FUNC_NAME')):
            numTuple = bool_to_num(getattr(func, app.config.get('FUNC_NAME'))(url, str(serial_num)))
            data = {
                    'script_name': script_name,
                    'serial_num': serial_num,
                    'is_alive': numTuple.is_alive,
                    'is_index_404': numTuple.is_index_404,
                    'is_vuln': numTuple.is_vuln,
                    'is_vuln_404':numTuple.is_vuln_404,
                    'description': vulPointMsg.description,
                    'ip': ip,
                    'port': port,
                    'match_name':vulIp.match_name,
                    'match_id':vulIp.match_id,
                    'test_id':test_id,
                    'team_name':vulIp.team_name
                }
            VulDetectionAll.add(data)
            result.append(data)
    else:
        VulTest.add({'match_name':match_name, 'test_id':test_id})
        return {'code':200, 'status':1, 'data':result}
```