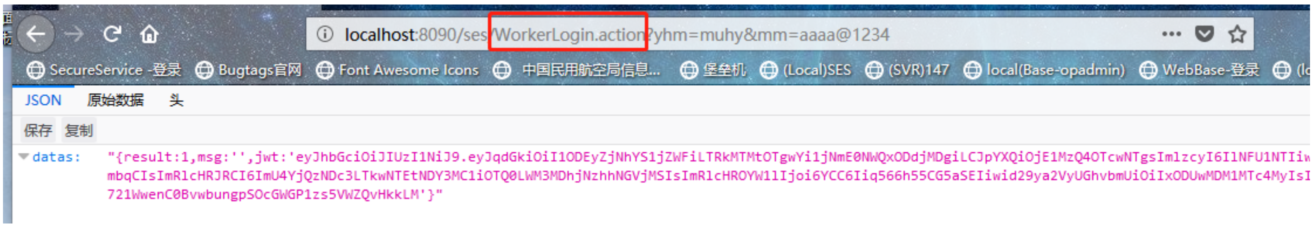
统一鉴权接口

登录

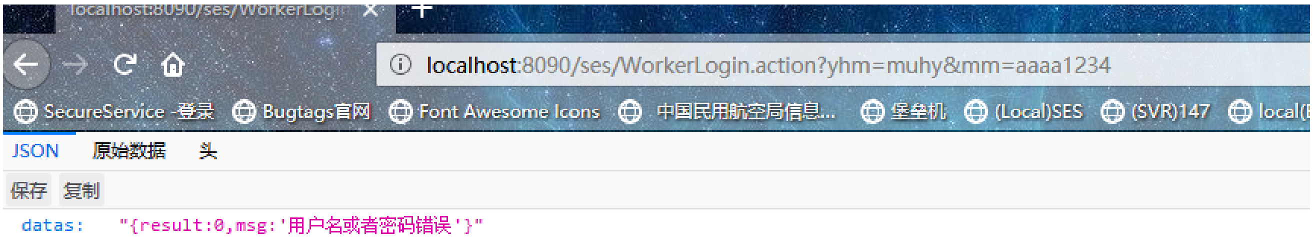
基于《行政执法接口说明.docx》，增加数据结构中若干数据段，支持法规库用户登录管理需求。

主要调整的内容：



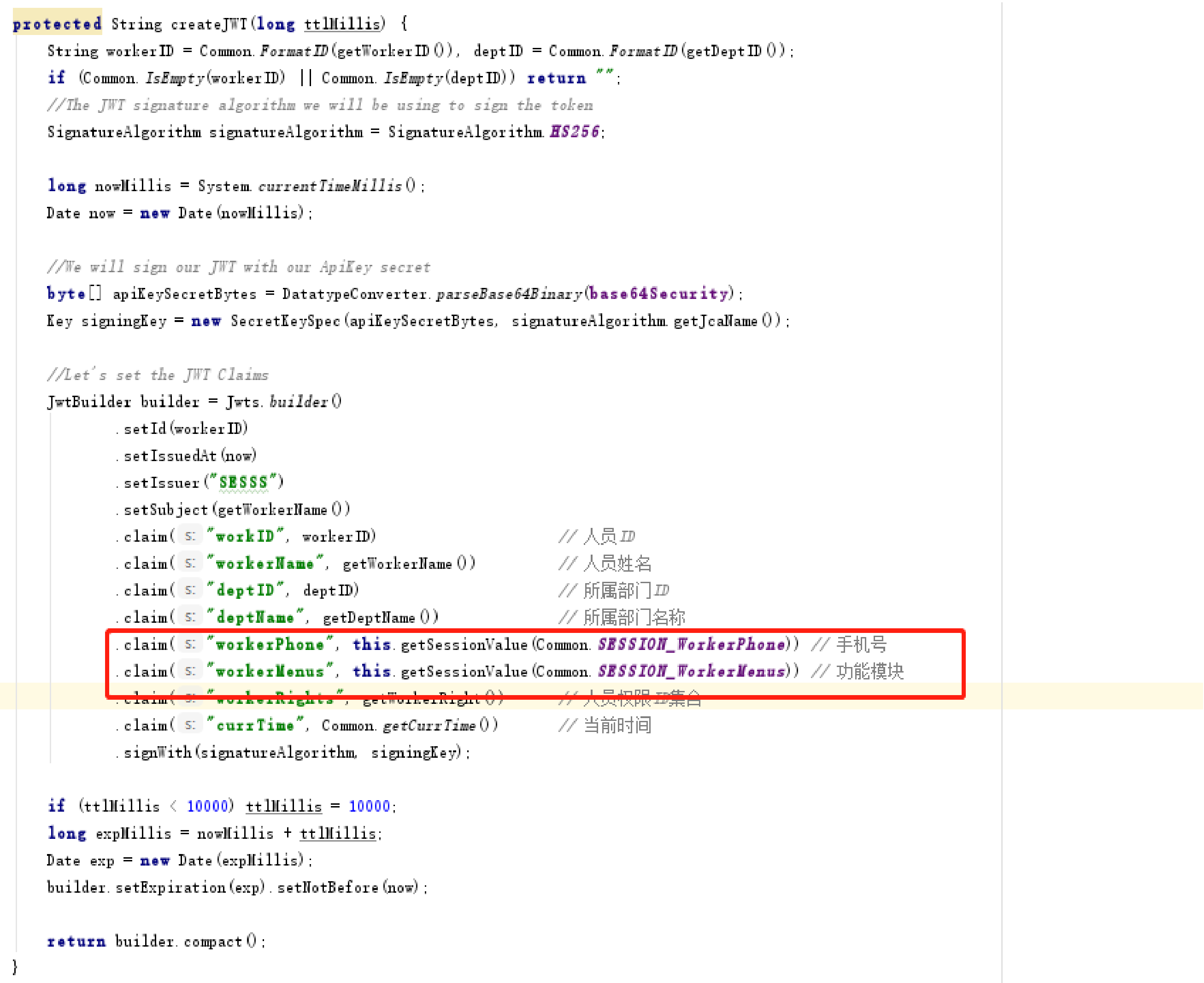
```
datas:   "{result:1,msg:'',jwt:'eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiI1ODEyZjNhYS1jZWFiLTRkMTMtOTgwYi1jNmE0NWQxODdjMDgiLCJpYXQiOjE1MzQ4OTcwNTgsImlzcyI6IlNFU1NTIiw
         mbqCIsImRlcHRJRCI6ImU4YjQzNDc3LTkwNTEtNDY3MC1iOTQ0LWM3MDhjNzhhNGVjMSIsImRlcHROYW1lIjoi6YCC6Iiq566h55CG5aSEIiwid29ya2VyUGhvbmUiOiIxODUwMDM1MTc4MyIsI
         721WwenC0BvwbungpSOcGWGP1zs5VWZQvHkkLM'}"
```

返回一个json，如果 result=1，那说明登录成功，会返回jwt ，否则，result=0，没有jwt。



```
datas:   "{result:0,msg:'用户名或者密码错误'}"
```

生成的token中增加了两个字段。

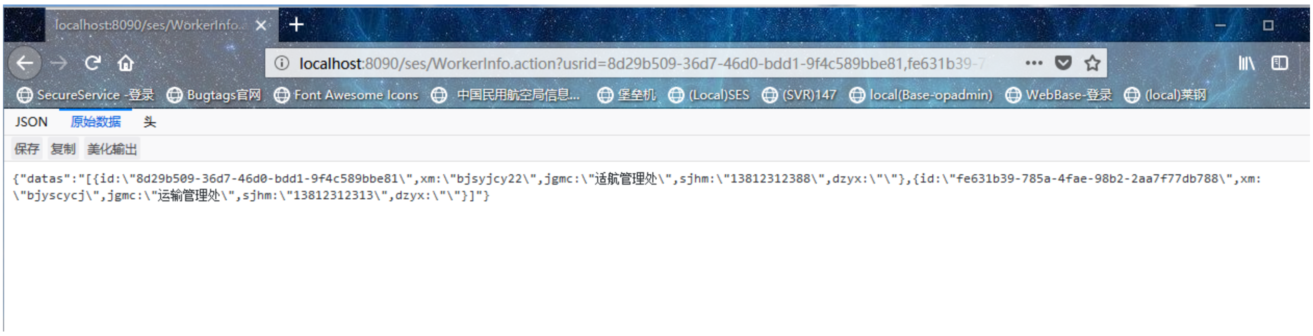


```
protected String createJWT(long ttlMillis) {
    String workerID = Common.FormatID(getWorkerID()), deptID = Common.FormatID(getDeptID());
    if (Common.IsEmpty(workerID) || Common.IsEmpty(deptID)) return "";
    //The JWT signature algorithm we will be using to sign the token
    SignatureAlgorithm signatureAlgorithm = SignatureAlgorithm.HS256;

    long nowMillis = System.currentTimeMillis();
    Date now = new Date(nowMillis);

    //We will sign our JWT with our ApiKey secret
    byte[] apiKeySecretBytes = DatatypeConverter.parseBase64Binary(base64Security);
    Key signingKey = new SecretKeySpec(apiKeySecretBytes, signatureAlgorithm.getJcaName());

    //Let's set the JWT Claims
    JwtBuilder builder = Jwts.builder()
            .setId(workerID)
            .setIssuedAt(now)
            .setIssuer("SESSS")
            .setSubject(getWorkerName())
            .claim( s: "workID", workerID)                // 人员ID
            .claim( s: "workerName", getWorkerName())     // 人员姓名
            .claim( s: "deptID", deptID)                  // 所属部门ID
            .claim( s: "deptName", getDeptName())         // 所属部门名称
            .claim( s: "workerPhone", this.getSessionValue(Common.SESSION_WorkerPhone)) // 手机号
            .claim( s: "workerMenus", this.getSessionValue(Common.SESSION_WorkerMenus)) // 功能模块
            .claim( s: "workerRights", getWorkerRight())  // 人员权限ID集合
            .claim( s: "currTime", Common.getCurrTime())  // 当前时间
            .signWith(signatureAlgorithm, signingKey);

    if (ttlMillis < 10000) ttlMillis = 10000;
    long expMillis = nowMillis + ttlMillis;
    Date exp = new Date(expMillis);
    builder.setExpiration(exp).setNotBefore(now);

    return builder.compact();
}
```

解密代码：

```
private String base64Security = "SES" + "983298sad32fhi9uy237632";

protected String createJWT() {
   long ttlMillis = 10 * 1000;
   return createJWT(ttlMillis);
}



protected String createJWT(long ttlMillis) {
   String workerID = Common.FormatID(getWorkerID()), deptID =
Common.FormatID(getDeptID());
   if (Common.IsEmpty(workerID) || Common.IsEmpty(deptID)) return "";
   //The JWT signature algorithm we will be using to sign the token
   SignatureAlgorithm signatureAlgorithm = SignatureAlgorithm.HS256;

   long nowMillis = System.currentTimeMillis();
   Date now = new Date(nowMillis);
   //We will sign our JWT with our ApiKey secret
   byte[] apiKeySecretBytes = DatatypeConverter.parseBase64Binary(base64Security);
   Key signingKey = new SecretKeySpec(apiKeySecretBytes,
signatureAlgorithm.getJcaName());

   //Let's set the JWT Claims
   JwtBuilder builder = Jwts.builder()
         .setId(workerID)
         .setIssuedAt(now)
         .setIssuer("SESSS")
         .setSubject(getWorkerName())
         .claim("workID", workerID)                // 人员ID
         .claim("workerName", getWorkerName())        // 人员姓名
         .claim("deptID", deptID)                  // 所属部门ID
         .claim("deptName", getDeptName())           // 所属部门名称
         .claim("workerPhone", this.getSessionValue(Common.SESSION_WorkerPhone)) // 手机
号
         .claim("workerMenus", this.getSessionValue(Common.SESSION_WorkerMenus)) // 功能
模块
         .claim("workerRights", getWorkerRight())       // 人员权限ID集合
         .claim("currTime", Common.getCurrTime())      // 当前时间
         .signWith(signatureAlgorithm, signingKey);

   if (ttlMillis < 10000) ttlMillis = 10000;
   long expMillis = nowMillis + ttlMillis;
   Date exp = new Date(expMillis);
   builder.setExpiration(exp).setNotBefore(now);

   return builder.compact();
}



protected Claims parseJWT(String jsonWebToken) {
      try {
         Claims claims = Jwts.parser()
               .setSigningKey(DatatypeConverter.parseBase64Binary(base64Security))
               .parseClaimsJws(jsonWebToken).getBody();
//       System.out.println("parseJWT:");
//       System.out.println("workID="+claims.get("workID",String.class));
//       System.out.println("workName="+claims.get("workName",String.class));
//       System.out.println("deptID="+claims.get("deptID",String.class));
//       System.out.println("deptName="+claims.get("deptName",String.class));
//       System.out.println("workerRights="+claims.get("workerRights",String.class));
         return claims;
      } catch (Exception ex) {
         return null;
      }
   }
```

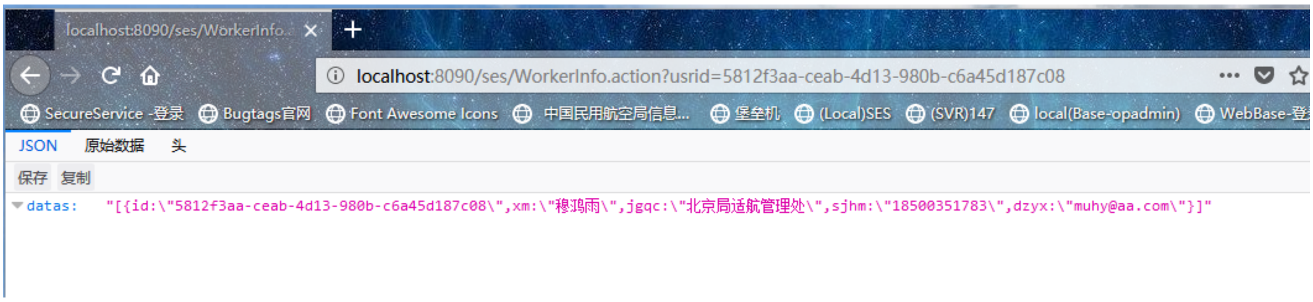
用户信息查询封装



localhost:8090/ses/WorkerInfo.action?usrid=8d29b509-36d7-46d0-bdd1-9f4c589bbe81,fe631b39-7

```
{"datas":"[{id:\"8d29b509-36d7-46d0-bdd1-9f4c589bbe81\",xm:\"bjsyjcy22\",jgmc:\"适航管理处\",sjhm:\"13812312388\",dzyx:\"\"},{id:\"fe631b39-785a-4fae-98b2-2aa7f77db788\",xm:
\"bjyscycj\",jgmc:\"运输管理处\",sjhm:\"13812312313\",dzyx:\"\"}]"}
```

员工单位的司局信息，jgqc：所属机构;



localhost:8090/ses/WorkerInfo.action?usrid=5812f3aa-ceab-4d13-980b-c6a45d187c08

```
datas:    "[{id:\"5812f3aa-ceab-4d13-980b-c6a45d187c08\",xm:\"穆鸿雨\",jgqc:\"北京局适航管理处\",sjhm:\"18500351783\",dzyx:\"muhy@aa.com\"}]"
```