

H5iframe.

iReport365

```
// JDK1.8
var version = java.lang.System.getProperty("java.version");

if (version.startsWith("1.8.0")) {
load("nashorn:mozilla_compat.js");
}
```

+

1.

<https://enterprise.ireport365.com/view/report.do?reportId=98062e02a4294eda86423f464485089e>

2.

<https://enterprise.ireport365.com/view/report.do?reportId=98062e02a4294eda86423f464485089e&username=demo&password=123456>

username	
password	
md5password	md5

passwordmd5password

+Token

token

1.

<https://enterprise.ireport365.com/view/report.do?reportId=98062e02a4294eda86423f464485089e>

2. token

https://enterprise.ireport365.com/view/report.do?reportId=98062e02a4294eda86423f464485089e&username=demo&token=d3812j12m1k21127120

username	
token	token

3. -- token

iReport365SQLWebService

1. ---
- 2.

result	
databaseService	
webService	HTTP
ldapService	LDAP
userService	
redisService	Redis
reportService	
wxcpService	
sessionService	Session
request	
out	System.out
logger	Logger

result

success	
message	
context	

useStandardVerification	true
-------------------------	------

databaseService

public List query(String connectionName, String sql)	sqlList
public List query(String connectionName, String sql, List param)	sqlsqlList

webService

public String get(String urlStr)	url
public String get(String urlStr, Map<String, String> header)	
public String post(String urlStr)	
public String post(String urlStr, String bodyContent)	
public String post(String urlStr, String contentType, String bodyContent)	
public String post(String urlStr, Map<String, String> header, String bodyContent)	

userService

public String hasUser(String username)	
public String verifyUser(String username, String password, boolean originPassword)	originPasswordtrue
public boolean loginUser(String username)	

sessionService

public void putString(String key, String value)	session
public String getString(String key)	session

ldapService

public boolean auth(String host, String port, String type, String domain, String username, String password)	LDAP

wxcpService

public void addUserToTag(String configId, String tagName, Collection<String> usernames)	TAG

```
var wxConfigId = "abc"; //ID->->->ID
var wxTagName = ""; //
var usernames = JSONArray();
usernames.add("username1"); //
usernames.add("username2"); //
wxcpService.addUserToTag(wxConfigId, wxTagName, usernames);
```

redisService

public String get(String ip, int port, String key)	
public String get(String ip, int port, String key, String password)	
public String clusterGet(List addrs, String key)	
public void del(String ip, int port, String key)	
public void del(String ip, int port, String key, String password)	
public void clusterDel(List addrs, String key)	

Redis

```
var redisKey = "theKey";
var redisAddrs = new ArrayList();
redisAddrs.add("192.168.1.2:7000");
redisAddrs.add("192.168.1.3:7001");
redisAddrs.add("192.168.1.4:7002");
var theValue = redisService.clusterGet(redisAddrs, redisKey);
```

```
var version = java.lang.System.getProperty("java.version");
if (version.startsWith("1.8.0")) {
    load("nashorn:mozilla_compat.js");
}
importPackage(java.util);
importPackage(org.apache.commons.codec.digest);
//
var username=request.getParameter("username");
var password=request.getParameter("password");
//MD5
var md5Password = DigestUtils.md5Hex(password);
//
var connectionName = "local dataset";
var param = new ArrayList();
param.add(username);
param.add(md5Password);
//
var sql = "select username,password from users where username=? and
password=?";
var data = databaseService.query(connectionName, sql, param);
if(data.size() == 0){
    result.setSuccess(false);
    result.setMessage("");
}else{
    result.setSuccess(true);
}
```

LDAP

```
var version = java.lang.System.getProperty("java.version");
if (version.startsWith("1.8.0")) {
    load("nashorn:mozilla_compat.js");
}
importPackage(java.util);
importPackage(org.apache.commons.codec.digest);
//
var username=request.getParameter("username");
var password=request.getParameter("password");
//LDAP
var host = "127.0.0.1";//LDAP
var port = "389"; //LDAP
var type = "simple"; //
var domain = "abc.com"; //
var success =
ldapService.auth(host,port,type,domain,username,password);
if(success){
    result.setSuccess(true);
}else{
    result.setSuccess(false);
    result.setMessage("");
}
```

WebService

```
var version = java.lang.System.getProperty("java.version");
if (version.startsWith("1.8.0")) {
    load("nashorn:mozilla_compat.js");
}
importPackage(java.util);
importPackage(org.apache.commons.codec.digest);
importPackage(com.alibaba.fastjson);
importPackage(org.apache.commons.lang);
//
var username=request.getParameter("username");
var password=request.getParameter("password");
var str =
webService.get("http://127.0.0.1:8080/login?username="+username+"&password="+password);
if(!StringUtils.isEmpty(str)){
    var json = JSONObject.parseObject(str);
    var success = json.getBoolean("success");
    if(!success){
        result.setSuccess(false);
        result.setMessage("");
    }else{
        result.setSuccess(true);
    }
}else{
    result.setSuccess(false);
    result.setMessage("");
}
```

```

var version = java.lang.System.getProperty("java.version");
if (version.startsWith("1.8.0")) {
    load("nashorn:mozilla_compat.js");
}
importPackage(java.util);
importPackage(com.alibaba.fastjson);
importPackage(org.apache.commons.lang);
var dataPermissionData = new HashMap();
//URL
var accessToken=request.getParameter("accessToken");
var appSecret="some code";
var url =
"http://www.abc.com/shopguide/api/org/getOrgWithinAuth?appSecret="+appSecret+"&accessToken="+accessToken;
logger.debug("ur: "+url);
//
var response = webService.get(url);
logger.debug("response: "+response);
if(!StringUtils.isEmpty(response)){
    //
    var json = JSONObject.parseObject(response);
    var message = json.getString("msg");
    logger.debug("message: "+message);
    if(StringUtils.equals(message, "success")){
        var values = json.getJSONArray("value");
        logger.debug("values: "+values);
        if(values != null){
            var codeArr = new ArrayList();
            for(var m = 0; m < values.size(); m++){
                var item = values.getJSONObject(m);
                var code = item.getString("code");
                codeArr.add(code);
            }
            //SHOPCODE
            dataPermissionData.put("SHOPCODE", codeArr);
        }
    }
}
result.setSuccess(true);
result.setData(dataPermissionData);
logger.debug("dataPermissionData: "+dataPermissionData);

```