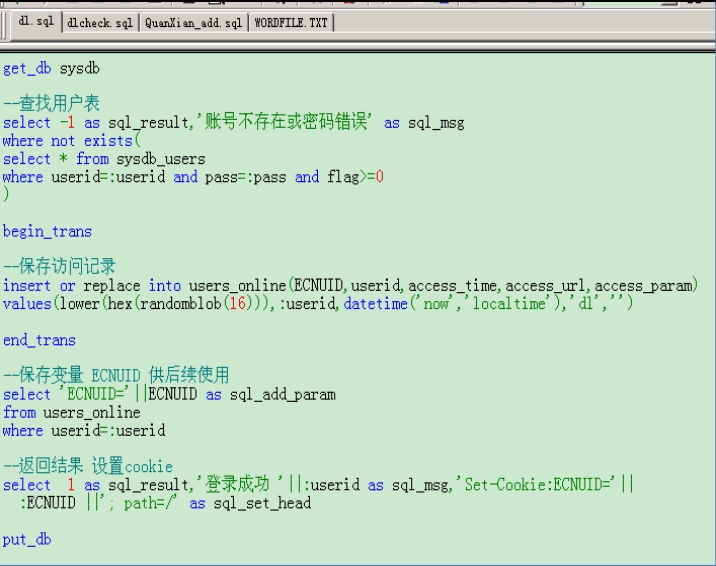



sql脚本功能			
概述	后台提供WebSqlScript接口服务：前端指定某个名称作为脚本的名称，并带上需要的参数提交到后台，由后台执行这个名称对应的一个.sql脚本文件，文件里主要存放数据库执行的sql语句并由后台依次执行		
脚本名称	可以取有业务含义的名称，如get_res，表示请求资源信息，名称一般要带上前缀，表示属于某个应用，如mapb.get_res,表示mapbuilder下的get_res脚本；请求时，名称以json的属性格式“scriptname”:“xxxx”表示。这个脚本存放在后台data\sqlscript\对应的应用目录下。		
脚本参数	请求sql脚本时，需要带上业务上需要的参数，这些参数由后台具体的脚本内容指定。 如右的一个新增修改权限表的脚本，有两个冒号的标识(:user_add)和一个冒号的标识(:userid)；这些标识都可以由前端带入对应的参数，带入时冒号就不需要了。		
双冒号参数	(空格)::标识(空格) 表示一个 双冒号参数 双冒号参数，一般是用干控制流程，和sql语句的调整；后台使用 字符串替换 的方式，将前端对应的参数替换冒号和标识， 双冒号参数前端必须带入 ，否则sql脚本无法执行。带入的参数不能含有 标准sql语言 的关键字如select drop create等。 如右的权限表新增编辑脚本，:user_add跟随在 控制语句 (见下文)sql_if后面，需要前端带入“true”或“false”，表示如果为true，则执行其后的insert语句。 Sql语句 (见下文)中也可以指定形如select * from tbl order by ::col_order的参数，由前端带入具体的列名和排序方法，如“col_order”:“coll desc”		
单冒号参数	对于 单冒号参数 ，则是在sql语句里出现的 sql参数 ，后台将带冒号的sql语句提交给数据库引擎（目前是sqlite，下同），然后调后数据库 参数绑定 的功能将前端对应的参数值提供给数据库引擎，由数据库引擎执行这个sql语句。 单冒号参数前端可以带入，也可以不带入，不带入时后台就使用null来绑定。		
脚本语句类型	脚本中语句由多个由 空格隔开的语句块 组成；每个语句块可以有多行，可以用“--”开头进行注释；语句块分为 控制性语句 和 sql语句 ；控制语句用来控制脚本是否执行、调用其它的sql脚本等，由 后台解释执行 。Sql语句即为 标准sql语言 ，由后台提交给 数据库引擎执行 。		
sql语句及几个特殊字段 sql_result sql_set_head sql_add_param	1、 sql语句 就是标准的数据库sql语言，支持select insert create 之类；具体支持类型及技术文档可以参考 http://www.sqlite.org/ ；sql语句由后台提交给数据库引擎，并绑定好参数来执行，sql语句的执行结果都会返回给前端，下文提及的含特殊字段的select型的sql语句若无返回记录，则不返回给前端。 2、对于select型的sql语句，若第一个字段名为sql_result,则后台假设此语句返回1条或0条记录，它表示执行到此语句块的一个成功与否的结果；且若这个字段值小于0，则后台 不再执行后续所有的脚本 ；若大于0，则代表正常的执行结果；这条语句的后续字段，有sql_msg字段 其值做为对sql_result的描述；可以有sql_set_head字段 其值将由后台附加到http消息头里返回给前端；当需要返回给前端 cookie 时就需要用到这种机制。见右端的登录脚本。 3、对于select型的sql语句，若第一个字段名为sql_add_param,则其值将附加到前端提交参数里，供后续脚本使用。一个典型的作用是：后台在执行登录验证脚本时，依据ECNUID查询到当前用户id，并利用sql_add_param加入到参数里，供后续脚本继续验证其它权限，若由前端来传入当前用户id就不安全了。见右端的登录检查脚本		
控制语句 get_db put_db	这两个语句用来请求数据库和释放数据库； 1、get_db用来请求数据库，后续的sql语句就可以使用数据库里的表；get_db后跟一个数据库或多个数据库 如get_db sysdb;或get_db sysdb,ecnugis;请求多个数据库时，sql语句就可以 跨库访问表 ，若有同名的表，则需要加上对应的数据库名做前缀；这些数据库文件存放在data\geodb\目录下； 2、数据库使用完成后需要调用put_db;释放最近请求的数据； 3、get_db,put_db只能一对一出现，不能嵌套使用，即get_db和put_db中间不能再出现get_db,也不能调用其它脚本（见下文SqlScript控制语句）；		

控制语句 begin_trans end_trans	<p>这两个语句用来启动和提交数据库事务；当业务上针对多个表的更新操作需要具有原子性时，就需要用到这两个语句；如果有针对跨数据库表的更新操作，则事务在每个数据库上具有原子性，而全部数据库不具有原子性；</p> <p>数据库事务典型应用是：新建图层操作，既有在ecnugis库里新增图层记录 and 地图空间记录共两次操作，又有在sysdb里新增对应的资源表和资源权限控制表的两次操作；建议先更新sysdb再更新ecnugis；这样若有事务失败的情景，通过sysdb能检查出不一致。</p>	 <pre>dl sql dlcheck sql QuanXian_add.sql WORDFILE.TXT get_db sysdb --查找用户表 select -1 as sql_result, '账号不存在或密码错误' as sql_msg where not exists(select * from sysdb_users where userid=:userid and pass=:pass and flag>=0) begin_trans --保存访问记录 insert or replace into users_online(ECNUID,userid,access_time,access_url,access_param) values(lower(hex(randomblob(16))),:userid,datetime('now','localtime'),'dl','') end_trans --保存变量 ECNUID 供后续使用 select 'ECNUID=' ECNUID as sql_add_param from users_online where userid=:userid --返回结果 设置cookie select 1 as sql_result, '登录成功' :userid as sql_msg, 'Set-Cookie:ECNUID=' :ECNUID ';' path=/ as sql_set_head put_db</pre>	
控制语句 sql_if	sql_if ::参数 表示是否执行紧接着它的那1条sql语句；参数由前端传入。		
控制语句 sqlScript	sqlScript sql脚本名称 表示调用指定的sql脚本；这个脚本中若有个sql_result类型的sql语句且值小于0，则调用脚本及本脚本都不继续执行。		
控制语句 sql_csv	sql_csv 将其后的sql语句的查询结果导入到csv文件并压缩下载，含字段。	 <pre>sql脚本 req: {"scriptname":"xls报表导出", "fn":"xls报表/三林/项目进度管理计划表.xls", "startx":4, "starty":1 } --SQLScript mapb.dlcheck get_db slprj sql_xls ::fn , ::startx , ::starty select * from slprj_conProject limit 50 put_db</pre>	
控制语句 sql_xls	<p>sql_xls 将查询结果导入到指定的xls文件，并压缩下载，不含字段。</p> <p>格式： sql_xls xls模板的文件名,从x,y处开始填写查询到的记录 sql语句</p> <p>调用方式与后台示例脚本见右边截图</p>		