GCM 后台地址：http://124.16.144.48:81

测试帐号：[yangs@im.ac.cn](mailto:yangs@im.ac.cn)

密码：test

注：{表单中的字段} 均代表gcm登录后添加菌种页面中所包含的字段，由于参数较多，不一一列举，可登录gcm后台查看字段对应的参数名称，使用http post 请求，最好使用post

1. 接口调用之前必须登录，登录之后会返回一个jsessionid作为key,与登录标识（loginflag）

|  |  |
| --- | --- |
| URL | http://124.16.144.48:81/api/login |
| 参数 | username=yangs@im.ac.cn  password=test |
| 返回结果 | {"jsessionid":"ei0syyo6fn26iw4u5vfs3d2r","loginFlag":"true"} //登录成功  {"loginFlag":"false"}//登录失败 |

Code:

|  |
| --- |
| **public** String httplogin() **throws** Exception{  String urlStr="http://124.16.144.48:81/api/login?username=yangs@im.ac.cn&password=test";  PrintWriter out = **null**;  BufferedReader in = **null**;  StringBuffer result = **new** StringBuffer();  URL url = **new** URL(urlStr);  URLConnection conn = url.openConnection();  conn.setRequestProperty("accept", "\*/\*");  conn.setRequestProperty("connection", "Keep-Alive");  conn.setRequestProperty("user-agent","Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)");  conn.setDoOutput(**true**);  conn.setDoInput(**true**);  out = **new** PrintWriter(conn.getOutputStream());  out.flush();  in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream()));  String line = "";  **while** ((line = in.readLine()) != **null**) {  result.append(line);  }  **if** (out != **null**) {  out.close();  out = **null**;  }  **if** (in != **null**) {  in.close();  in = **null**;  }  **return** result.toString();  } |

1. 菌种增加

|  |  |
| --- | --- |
| URL | http://124.16.144.48:81/api/strains/add;jsessionid= |
| 参数 | 表单中的字段 |
| 返回结果 | True  False |

|  |
| --- |
| **public** **void** addstrain() **throws** Exception{  String targetURL=**null**;  **try**{  targetURL="http://124.16.144.48:81/api/strains/add;jsessionid="+JSON.*parseObject*(**new** Test().httplogin()).getString("jsessionid") ;  }**catch**(Exception e){  e.printStackTrace();  }  PostMethod post = **new** PostMethod(targetURL);  post.setParameter("strain\_passport.strain\_number", "test");  post.setParameter("strain\_passport.other\_collection\_numbers", "test");  post.setParameter("strain\_passport.genus\_name", "test");  post.setParameter("strain\_passport.species\_name", "test");  post.setParameter("strain\_passport.variety\_name", "test");  post.setParameter("strain\_passport.subspecies\_name", "test");  post.setParameter("strain\_passport.organism\_type", "test");  post.setParameter("strain\_passport.author", "test");  post.setParameter("strain\_passport.date\_of\_Isolation", "test");  post.setParameter("strain\_passport.history\_of\_deposit", "test");  post.setParameter("strain\_passport.isolated\_from", "test");  post.setParameter("strain\_passport.geographic\_origin", "test");  post.setParameter("strain\_passport.status", "test");  post.setParameter("strain\_passport.type\_strain", "YES");  post.setParameter("strain\_passport.optimum\_temperature\_for\_growth", "test");  post.setParameter("strain\_passport.maximum\_temperature\_for\_growth", "test");  post.setParameter("strain\_passport.minimum\_temperature\_for\_growth", "test");  post.setParameter("strain\_passport.literature", "test");  post.setParameter("strain\_passport.application", "test");  post.setParameter("strain\_passport.soursurl", "test");  HttpClient client = **new** HttpClient();  client.getHttpConnectionManager().getParams().setConnectionTimeout(5000);  **int** status = client.executeMethod(post);  **if** (status == HttpStatus.*SC\_OK*) {  InputStream in = post.getResponseBodyAsStream();  BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(in));  StringBuilder sb = **new** StringBuilder();  String line = **null**;  **try** {  **while** ((line = reader.readLine()) != **null**) {  sb.append(line);  }  Thread.*sleep*(3000);  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  in.close();  }  System.*out*.println(sb.toString());  } **else** {  System.*out*.println("Can not connect");  }  } |

1. 菌种修改

|  |  |
| --- | --- |
| URL | http://124.16.144.48:81/api/strains/update;jsessionid= |
| 参数 | strain\_passport.id 要更新的菌种id  表单中的字段 |
| 返回结果 | True  False |

|  |
| --- |
| **public** **void** updateStrain() **throws** Exception{  String targetURL=**null**;  **try**{  targetURL="http://124.16.144.48:81/api/strains/update;jsessionid="+JSON.*parseObject*(**new** Test().httplogin()).getString("jsessionid") ;  }**catch**(Exception e){  e.printStackTrace();  }  PostMethod post = **new** PostMethod(targetURL);  post.setParameter("strain\_passport.id", "341079");  post.setParameter("strain\_passport.strain\_number", "yangs");  post.setParameter("strain\_passport.other\_collection\_numbers", "yangs");  post.setParameter("strain\_passport.genus\_name", "yangs");  post.setParameter("strain\_passport.species\_name", "test");  post.setParameter("strain\_passport.variety\_name", "test");  post.setParameter("strain\_passport.subspecies\_name", "test");  post.setParameter("strain\_passport.organism\_type", "test");  post.setParameter("strain\_passport.author", "test");  post.setParameter("strain\_passport.date\_of\_Isolation", "test");  post.setParameter("strain\_passport.history\_of\_deposit", "test");  post.setParameter("strain\_passport.isolated\_from", "test");  post.setParameter("strain\_passport.geographic\_origin", "test");  post.setParameter("strain\_passport.status", "test");  post.setParameter("strain\_passport.type\_strain", "YES");  post.setParameter("strain\_passport.optimum\_temperature\_for\_growth", "test");  post.setParameter("strain\_passport.maximum\_temperature\_for\_growth", "test");  post.setParameter("strain\_passport.minimum\_temperature\_for\_growth", "test");  post.setParameter("strain\_passport.literature", "test");  post.setParameter("strain\_passport.application", "test");  post.setParameter("strain\_passport.soursurl", "test");  HttpClient client = **new** HttpClient();  client.getHttpConnectionManager().getParams().setConnectionTimeout(5000);  **int** status = client.executeMethod(post);  **if** (status == HttpStatus.*SC\_OK*) {  InputStream in = post.getResponseBodyAsStream();  BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(in));  StringBuilder sb = **new** StringBuilder();  String line = **null**;  **try** {  **while** ((line = reader.readLine()) != **null**) {  sb.append(line);  }  Thread.*sleep*(3000);  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  in.close();  }  System.*out*.println(sb.toString());  } **else** {  System.*out*.println("Can not connect");  }  } |

1. 菌种删除

|  |  |
| --- | --- |
| URL | http://124.16.144.48:81/api/strains/del;jsessionid= |
| 参数 | Id |
| 返回结果 | True  False |

Code

|  |
| --- |
| **public** **void** delete() **throws** Exception{  String targetURL = **null**;// URL  **try** {  targetURL = "http://124.16.144.48:81/api/strains/del;jsessionid="+JSON.*parseObject*(**new** Test().httplogin()).getString("jsessionid") ;  } **catch** (Exception e1) {  e1.printStackTrace();  }  PostMethod post = **new** PostMethod(targetURL);  post.setParameter("id", "320767");  HttpClient client = **new** HttpClient();  client.getHttpConnectionManager().getParams().setConnectionTimeout(5000);  **int** status = client.executeMethod(post);  **if** (status == HttpStatus.*SC\_OK*) {  InputStream in = post.getResponseBodyAsStream();  BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(in));  StringBuilder sb = **new** StringBuilder();  String line = **null**;  **try** {  **while** ((line = reader.readLine()) != **null**) {  sb.append(line);  }  Thread.*sleep*(3000);  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  in.close();  }  System.*out*.println(sb.toString());  } **else** {  System.*out*.println("Can not connect");  }  } |

1. 菌种列表查询

|  |  |
| --- | --- |
| URL | http://124.16.144.48:81/api/strains/list;jsessionid= |
| 参数 | currentPage  strainnumber  strainname |
| 返回结果 | {"pageSize":15,"pageNumber":1,"strainname":"","list":[{"id":320767,"history\_of\_deposit":"dsa","name":"test test test","strain\_number":"123345test"},{"id":320768,"history\_of\_deposit":"dsa","name":"dsa dsa dsa","strain\_number":"test"}],"strainnumber":"","totalpage":1,"rows":2} |

1. 单条菌种内容查询

|  |  |
| --- | --- |
| URL | http://124.16.144.48:81/api/strains/view?id=xxx;jsessionid= |
| 参数 | Id |
| 返回结果 | "strain\_bigconfig\_strain":{"heat\_resistance":null,"altitude":null,"humidity\_for\_environment":null,"tolerances\_and\_sensitivities":null,"ethanol\_conditions":null,"ph\_for\_environment":null,"conditions\_for\_fruiting\_or\_sporulation":null,"biotransformations":null,"single\_compounds\_tested":null,"enzymes\_produced":null,"fine\_structure\_data":null,"osmophily\_and\_xerophily":null,"oxygen\_relationship":null,"light\_conditions":null,"deficiencies":null,"date\_of\_accession":null,"form\_of\_supply":null,"conditions\_of\_water\_activity":null,"cell\_contents":null,"phenotype":null,"deposited\_by":null,"minimum\_growth\_ph":null,"cellshape":null,"toxicity\_to\_other\_organisms":null,"maximum\_growth\_ph":null,"restriction\_for\_destribute":null,"hybridization\_with\_other\_strains":null,"maximum\_growthtemperature":null,"rehydration\_fluid":null,"temperature\_of\_hybridization":null,"strainid":null,"serovar":null,"latitude":null,"optimum\_minimum\_maximum\_sugar\_concentrations":null,"internal\_check\_of\_identity\_at\_collection":null,"herbarium\_no":null,"region":null,"motileby":null,"genotype":null,"teleomorph\_name":null,"taxonomyid":null,"carbon\_sources\_that\_have\_been\_tested\_for\_growth":null,"record\_type":null,"mode\_of\_preservation\_storage":null,"optimum\_minimum\_maximum\_nacl\_concentration":null,"country\_of\_export":null,"minimum\_growth\_temperatue":null,"plasmid":null,"date\_of\_collection":null,"special\_form\_name":null,"decomposition\_and\_deteriorating\_capacities":null,"metabolites\_produced":null,"date\_of\_update":null,"mutants":null,"history":null,"conditions\_for\_germination":null,"bio\_hazard\_level":null,"mating\_type":null,"motile":null,"name\_at\_accept":null,"method\_by\_which\_mutation\_was\_induced":null,"optimal\_growth\_temperatue":null,"optimal\_growth\_ph":null,"pathogenicity":null,"collection\_method":null,"race":null,"remarks\_on\_restrictions":null,"type\_of\_dna\_or\_rna":null,"temperature\_relationships\_for\_growth":null,"animal\_quarantine\_no":null,"longitude":null,"isolated\_by":null,"antagonistic\_activities\_against\_other\_organisms":null,"nutritional\_requirements\_and\_growth\_factors\_required":null,"variety\_name":null,"sexual\_reproduction":null,"temperatue\_for\_environment":null,"killer\_properties\_of\_yeast":null,"permission":null,"plant\_quarantine\_no":null,"misapplied\_name":null,"depth":null,"country":null,"strain\_passport\_id":341079,"habitat":null,"hybrids":null,"mta":null,"sexual\_behaviour":null,"identified\_by":null,"sexual\_state":null,"method\_of\_isolation":null,"sporeforming":null,"state":null,"cellsize":null,"gc\_content\_of\_dna":null,"wall\_constituents":null,"anamorph\_name":null,"conditions\_for\_growth\_in\_liquid\_media":null,"substrate":null,"collected\_by":null,"number\_of\_nuclei":null,"staining\_reactions":null,"id":184206,"symbiosis":null,"former\_name":null,"date\_of\_identification":null,"source\_of\_isolation":null,"conditions\_for\_growth\_and\_maintenance\_on\_solid\_media":null,"pigment\_production\_and\_autofluorescence":null,"hybridization\_strain\_number":null,"nitrogen\_sources\_that\_have\_been\_tested\_for\_growth":null,"allergenicity":null,"percentage":null,"coenzyme-q\_system":null,"mycoparasitism":null,"date\_of\_deposition":null,"salinity\_requirements\_for\_growth":null,"group":null},"username":"test","strain\_big":{"heat\_resistance":null,"altitude":null,"humidity\_for\_environment":null,"tolerances\_and\_sensitivities":null,"ethanol\_conditions":null,"ph\_for\_environment":null,"conditions\_for\_fruiting\_or\_sporulation":null,"biotransformations":null,"single\_compounds\_tested":null,"enzymes\_produced":null,"fine\_structure\_data":null,"osmophily\_and\_xerophily":null,"oxygen\_relationship":null,"light\_conditions":null,"deficiencies":null,"date\_of\_accession":null,"form\_of\_supply":null,"conditions\_of\_water\_activity":null,"cell\_contents":null,"phenotype":null,"deposited\_by":null,"minimum\_growth\_ph":null,"cellshape":null,"toxicity\_to\_other\_organisms":null,"maximum\_growth\_ph":null,"restriction\_for\_destribute":null,"hybridization\_with\_other\_strains":null,"maximum\_growthtemperature":null,"rehydration\_fluid":null,"temperature\_of\_hybridization":null,"strainid":null,"serovar":null,"latitude":null,"optimum\_minimum\_maximum\_sugar\_concentrations":null,"internal\_check\_of\_identity\_at\_collection":null,"herbarium\_no":null,"region":null,"motileby":null,"genotype":null,"teleomorph\_name":null,"taxonomyid":null,"carbon\_sources\_that\_have\_been\_tested\_for\_growth":null,"record\_type":null,"mode\_of\_preservation\_storage":null,"optimum\_minimum\_maximum\_nacl\_concentration":null,"country\_of\_export":null,"minimum\_growth\_temperatue":null,"plasmid":null,"date\_of\_collection":null,"special\_form\_name":null,"decomposition\_and\_deteriorating\_capacities":null,"metabolites\_produced":null,"date\_of\_update":null,"mutants":null,"history":null,"conditions\_for\_germination":null,"bio\_hazard\_level":null,"mating\_type":null,"motile":null,"name\_at\_accept":null,"method\_by\_which\_mutation\_was\_induced":null,"optimal\_growth\_temperatue":null,"optimal\_growth\_ph":null,"pathogenicity":null,"collection\_method":null,"race":null,"remarks\_on\_restrictions":null,"type\_of\_dna\_or\_rna":null,"temperature\_relationships\_for\_growth":null,"animal\_quarantine\_no":null,"longitude":null,"isolated\_by":null,"antagonistic\_activities\_against\_other\_organisms":null,"nutritional\_requirements\_and\_growth\_factors\_required":null,"variety\_name":null,"sexual\_reproduction":null,"temperatue\_for\_environment":null,"killer\_properties\_of\_yeast":null,"permission":null,"plant\_quarantine\_no":null,"misapplied\_name":null,"depth":null,"country":null,"strain\_passport\_id":341079,"habitat":null,"hybrids":null,"mta":null,"sexual\_behaviour":null,"identified\_by":null,"sexual\_state":null,"method\_of\_isolation":null,"sporeforming":null,"state":null,"cellsize":null,"gc\_content\_of\_dna":null,"wall\_constituents":null,"anamorph\_name":null,"conditions\_for\_growth\_in\_liquid\_media":null,"substrate":null,"collected\_by":null,"number\_of\_nuclei":null,"staining\_reactions":null,"id":184196,"symbiosis":null,"former\_name":null,"date\_of\_identification":null,"source\_of\_isolation":null,"conditions\_for\_growth\_and\_maintenance\_on\_solid\_media":null,"pigment\_production\_and\_autofluorescence":null,"hybridization\_strain\_number":null,"nitrogen\_sources\_that\_have\_been\_tested\_for\_growth":null,"allergenicity":null,"percentage":null,"coenzyme-q\_system":null,"mycoparasitism":null,"date\_of\_deposition":null,"salinity\_requirements\_for\_growth":null,"group":null},"strain\_passport":{"history\_of\_deposit":"test","maximum\_temperature\_for\_growth":"test","organism\_type":"test","geographic\_origin":"test","othername":null,"medium\_id":"353866","isolated\_from":"test","id":341079,"author":"test","showdate":"test","species\_name":"test","name":"yangs test test test","genus\_name":"yangs","other\_collection\_numbers":"yangs","user\_id":152,"subspecies\_name":"test","bacteria\_id":null,"application":"test","variety\_name":"test","status":"test","literature":"test","permission":2,"soursurl":"test","type\_strain":"YES","minimum\_temperature\_for\_growth":"test","optimum\_temperature\_for\_growth":"test","genbanktype":null,"strain\_number":"yangs","date\_of\_Isolation":"test"},"sequence":{"strain\_passport\_id":341079,"sequencing\_time":null,"accession\_number":null,"id":340814,"sequencing\_method":null,"sequence":null,"description":null,"length":null,"sequencing\_image":null,"sequence\_type":null,"sequence\_name":null,"primer":null},"medium":{"id":353866,"composition":"","medium\_number":"","medium\_name":"","collection":""}} |

Code

|  |
| --- |
| **public** **void** viewStrain() **throws** Exception{  String targetURL=**null**;  **try**{  targetURL="http://124.16.144.48:81/api/strains/view;jsessionid="+JSON.*parseObject*(**new** Test().httplogin()).getString("jsessionid") ;  }**catch**(Exception e){  e.printStackTrace();  }  PostMethod post = **new** PostMethod(targetURL);  post.setParameter("id", "341079");  HttpClient client = **new** HttpClient();  client.getHttpConnectionManager().getParams().setConnectionTimeout(5000);  **int** status = client.executeMethod(post);  **if** (status == HttpStatus.*SC\_OK*) {  InputStream in = post.getResponseBodyAsStream();  BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(in));  StringBuilder sb = **new** StringBuilder();  String line = **null**;  **try** {  **while** ((line = reader.readLine()) != **null**) {  sb.append(line);  }  Thread.*sleep*(3000);  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  in.close();  }  System.*out*.println(sb.toString());  } **else** {  System.*out*.println("Can not connect");  }  } |

7.XML上传，批量导入（如果菌号相同，则更新均中信息）

|  |  |
| --- | --- |
| URL | http://124.16.144.48:81/api/xmlimport |
| 参数 | 上传的文件对应的参数名为filename |
| 返回结果 | 例：{"msg\_num":"2"}  返回结果代码参考:  0:文件类型错误  1:xml文件内容错误并返回错误行数据  2:导入成功  3:xml解析异常，xml文件格式不正确  4:导入失败，请与管理员联系  loginflag:"false" 没有登录 JSESSIONID失效 |

Code

|  |
| --- |
| **public** **void** importxml(){  String targetURL = **null**;// URL  File targetFile = **null**;// upload file  targetFile = **new** File("C:/xmlFile.xml");  **try** {  targetURL = "http://124.16.144.48:81/api/xmlimport;jsessionid="+JSON.*parseObject*(**new** Test().httplogin()).getString("jsessionid") ;  } **catch** (Exception e1) {  e1.printStackTrace();  }  PostMethod filePost = **new** PostMethod(targetURL);  **try** {  // filePost.setParameter("name", "中文");  // filePost.setParameter("pass", "1234");  Part[] parts = { **new** FilePart("filename", targetFile) };  filePost.setRequestEntity(**new** MultipartRequestEntity(parts,  filePost.getParams()));  HttpClient client = **new** HttpClient();  client.getHttpConnectionManager().getParams().setConnectionTimeout(5000);  **int** status = client.executeMethod(filePost);  **if** (status == HttpStatus.*SC\_OK*) {  InputStream in = filePost.getResponseBodyAsStream();  BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(in));  StringBuilder sb = **new** StringBuilder();  String line = **null**;  **try** {  **while** ((line = reader.readLine()) != **null**) {  sb.append(line);  }  Thread.*sleep*(3000);  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  in.close();  }  System.*out*.println(sb.toString());  } **else** {  System.*out*.println("Can not connect");  }  } **catch** (Exception ex) {  ex.printStackTrace();  } **finally** {  filePost.releaseConnection();  }    } |