**Cmd task命令：**

|  |
| --- |
| tasklist | findstr 'java'  taskkill /f /t /im 6428  netstat -ano  netstat -aon | findstr '8080' |

**相对路径转绝对路径：**

|  |
| --- |
| **import** java.net.URL;  *//绝对路径* String absolutePath = "http://www.szzyjy.com.cn/jyxx.html"; *//相对路径* String relativePath = "/jyxx/003004.html"; *//以下方法对相对路径进行转换* **try *{*** URL absoluteUrl = **new** URL***(***absolutePath***)***;  URL parseUrl = **new** URL***(***absoluteUrl, relativePath***)***;  System.***out***.println***(***parseUrl.toString***())***; ***}* catch *(***MalformedURLException e***) {*** e.printStackTrace***()***; ***}*** |

**设置mysql主键自增值：**

|  |
| --- |
| alter table 表名 AUTO\_INCREMENT=1; |

**设置mysql时间自动更新：**

|  |
| --- |
| Timestamp，datetime CURRENT\_TIMESTAMP |

**获取当前日期的前一个月的第一天：**

|  |
| --- |
| *//获取最近两个月的信息* **public** Date getLastTowMonth***()* throws** ParseException ***{*** Calendar calendar = Calendar.*getInstance****()***;  calendar.add***(***Calendar.***MONTH***, -1***)***;  calendar.set***(***Calendar.***DAY\_OF\_MONTH***, calendar.getActualMinimum***(***Calendar.***DAY\_OF\_MONTH))***;  **return *simpleDateFormat***.parse***(simpleDateFormat***.format***(***calendar.getTime***()))***; ***}*** |

**发送GET请求：**

|  |
| --- |
| **public static** String sendGet***(***String url***) {*** String result = "";  BufferedReader in = **null**;  **try *{*** URL realUrl = **new** URL***(***url***)***;  *// 打开和URL之间的连接* URLConnection connection = realUrl.openConnection***()***;  *// 设置通用的请求属性* connection.setRequestProperty***(***"accept", "\*/\*"***)***;  connection.setRequestProperty***(***"connection", "Keep-Alive"***)***;  connection.setRequestProperty***(***"user-agent",  "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)"***)***;  *// 建立实际的连接* connection.connect***()***;  *// 定义 BufferedReader输入流来读取URL的响应* in = **new** BufferedReader***(*new** InputStreamReader***(*** connection.getInputStream***()))***;  String line;  **while *((***line = in.readLine***())*** != **null*) {*** result += line;  ***}  }* catch *(***Exception e***) {*** System.***out***.println***(***"发送GET请求出现异常！" + e***)***;  e.printStackTrace***()***;  ***}*** *// 使用finally块来关闭输入流* **finally *{* try *{* if *(***in != **null*) {*** in.close***()***;  ***}  }* catch *(***Exception e2***) {*** e2.printStackTrace***()***;  ***}  }* return** result; ***}*** |

**获取两个时间差的毫秒数:**

|  |
| --- |
| Calendar start = Calendar.*getInstance****()***;  **long** initDelay = start.getTimeInMillis***()***- System.*currentTimeMillis****()***; |

**Object转map：**

|  |
| --- |
| JSON.*parseObject****(***JSON.*toJSONString****(***bean***))*** |

**JSONObject转实体类：**

|  |
| --- |
| object.toJavaObject***(***SourceSysManage.**class*)*** |

**转Base64：**

|  |
| --- |
| ***<*dependency*>  <*groupId*>***org.apache.axis***</*groupId*>  <*artifactId*>***axis***</*artifactId*>  <*version*>***1.4***</*version*> </*dependency*>***  Base64.*encode****(***stream.toByteArray***())***;  **JDK1.8之后：**  final Base64.Decoder decoder = Base64.getDecoder();  final Base64.Encoder encoder = Base64.getEncoder();  final String text = "字串文字";  final byte[] textByte = text.getBytes("UTF-8");  //编码  final String encodedText = encoder.encodeToString(textByte);  System.out.println(encodedText);  //解码  System.out.println(new String(decoder.decode(encodedText), "UTF-8")); |

**Unicode转码：**

|  |
| --- |
| *//Unicode -> 中文* URLDecoder.*decode****(***"%E5%85%A8%E9%83%A8%E7%A7%8D%E7%B1%BB", "UTF-8"***)***; *//中文 -> Unicode* URLEncoder.*encode* ***(***"全部种类", "UTF-8"***)***; |

**Java8获取当前时间的时间戳：**

|  |
| --- |
| Instant.*now****()***.toEpochMilli***())*** |

**Map转对象：**

|  |
| --- |
| *// 将 Map 转换为 实体类* User user = JSON.parseObject***(***JSON.toJSONString***(***user01***)***, User.**class*)***; *// 将 实体类 转换为 Map* Map map = JSON.parseObject***(***JSON.toJSONString***(***user**)**, Map.**class*)***; |

**List map转list 对象：**

|  |
| --- |
| Fastjson：List***<***Map***<***String, Object***>>*** listMap = **new** HashMap***<***>***(***10***)***; JSONArray jsonArray = **new** JSONArray***()***; jsonArray.addAll***(***listMap***)***; List***<***User***>*** list = jsonArray.toJavaList***(***User.**class*)***;  Jackson：  ObjectMapper objectMapper = new ObjectMapper()  List<Clazz> list = objectMapper.convertValue(lists|str, new TypeReference<List<Clazz>>(){}); |

**List转数组，数组转list：**

|  |
| --- |
| *//list转数组* String***[]*** array2 = list.toArray***(*new** String***[***list.size***()])***; *//数组转list* ArrayList***<***String***>*** list = **new** ArrayList***<***String***>(***Arrays.*asList****(***array***))***;  Collections.*addAll****(***list, tableCodes***)***; |

**数据库层级路径：**

|  |
| --- |
| ***(***SELECT *GROUP\_CONCAT****(***a.mlmc***)*** from ***(***SELECT  *@rowNum*:=@rowNum + 1 AS num,  @r AS \_id,  ***(***SELECT **mlmc** from b\_mls WHERE **mmid** = \_id***)*** as mlmc,  ***(***SELECT *@r* := **sjmmid** FROM b\_mlgxs WHERE **mmid** = \_id***)*** AS sjmmid  FROM ***(***SELECT @r := ***#{***mlid***})*** vars,  ***(***SELECT @rowNum:=0***)*** b,  b\_mlgxs h  WHERE @r IS NOT NULL  GROUP BY num desc***)*** a***)*** |

**Java8 list按字段分组：**

|  |
| --- |
| Map***<***Object, List***<***ProjectTerrain***>>*** type = dataList.stream***()***.collect***(***Collectors.*groupingBy****(***ProjectTerrain::getCollectid***))***; |

**获取当前系统的文件路径分隔符：**

|  |
| --- |
| File.***separator*** |

**Maven安装jar包到仓库：**

|  |
| --- |
| mvn install:**install**-file -DgroupId=com.github.pagehelper -DartifactId=pagehelper -Dversion=5.1.2-xcloud-1 -Dpackaging=jar -Dfile=D:\idea\HyhtProjects\qhsw\qinghai-server-parent\lib\pagehelper-5.1.2-xcloud-1.jar |

**List排序：**

|  |
| --- |
| list.sort***(*new** Comparator***<***Integer***>() {*** @Override  **public int** compare**(**Integer o1, Integer o2**) *{* return** o1-o2;  ***} }*)**;    list.sort**((**o1, o2**)** -> o1.compareTo**(**o2**))**;  Collections.sort**(**list**)**;  list.stream***()***.sorted***((***o1, o2***)*** -> o1.compareTo***(***o2***))***.collect***(***Collectors.*toList****())***;  list.stream***()***.sorted***(***Comparator.*comparing****(***Integer::**new**, ***(***o1, o2***)*** -> o1.compareTo***(***o2***)))***.collect***(***Collectors.*toList****())***; |

**枚举转json字符串：**

|  |
| --- |
| 1、注解  @JsonFormat***(***shape = JsonFormat.Shape.***OBJECT)***  2、转json字符串  new ObjectMapper().writeValueAsString**(**TableCode.*values***())**; |

**数组转list：**

|  |
| --- |
| Collections.*addAll****(***list, tableCodes***)***; |

**Map的value转list：**

|  |
| --- |
| List***<***Object***>*** values = **new** ArrayList***<>(***map.values***());***  List list = map.values***()***.stream***()***.collect***(***Collectors.*toList****())***; |

**Mysql去除小数多余的0：**

|  |
| --- |
| 0 + *cast****(***1.1000 as **char*) //1.0无法去除***  *cast****(****cast****(***1.0000 as **real*)*** as **char*) //8.0***  TRIM(TRAILING '.' FROM TRIM(TRAILING '0' FROM CAST(cgj AS CHAR))); |

**Mysql慢查询：**

|  |
| --- |
| select \* from mysql.slow\_log where db = 'dldp\_pwy\_jj' ORDER BY start\_time desc |

**Mysql UUID重复问题：**

|  |
| --- |
| SELECT MD5(UUID***())*** |

**String转Date：**

|  |
| --- |
| **new** SimpleDateFormat***(***"yyyy-MM-dd HH:mm:ss"***)***.parse***(***"2020-11-01 08:30:20"***)***; |

**Date**

|  |
| --- |
| **new** SimpleDateFormat***(***"yyyy-MM-dd HH:mm:ss"***)***.format***(*new** Date***())***; |

**Mysql近一年的月份列表：**

|  |
| --- |
| SELECT DATE\_FORMAT***(*** DATE\_SUB**(** now**()**, INTERVAL CAST**(** help\_topic\_id AS signed INTEGER **)** MONTH **)**, '%Y-%m' **)** AS `month` FROM mysql.help\_topic  WHERE help\_topic\_id < TIMESTAMPDIFF**(** MONTH, DATE\_SUB**(** now**()**, INTERVAL 12 MONTH**)**, now**() )** ; |

**Mysql近一个月的日期列表：**

|  |
| --- |
| SELECT DATE\_FORMAT***(*** DATE\_SUB**(** now***()***, INTERVAL CAST**(** help\_topic\_id AS signed INTEGER **)** DAY **)**, '%m-%d' **) AS** `DAY` FROM mysql.help\_topic  WHERE help\_topic\_id < TIMESTAMPDIFF**(** DAY, DATE\_SUB***(*** now**(*)***, INTERVAL 1 MONTH **)**, now***()* )**; |

**Mysql近一周的日期列表：**

|  |
| --- |
| SELECT DATE\_FORMAT***(*** DATE\_SUB**(** now***()***, INTERVAL CAST**(** help\_topic\_id AS signed INTEGER **)** DAY **)**, '%m-%d' **) AS** `DAY` FROM mysql.help\_topic  WHERE help\_topic\_id < 7; |

**Mysql获取本周一的时间：**

|  |
| --- |
| SELECT DATE\_SUB(CURDATE(), INTERVAL WEEKDAY(CURDATE()) DAY) AS monday\_date; |

**Mysql获取本月一的时间：**

|  |
| --- |
| SELECT DATE\_ADD(CURDATE(), INTERVAL - DAY(CURDATE()) + 1 DAY) AS first\_day\_of\_month; |

**Mysql获取本年一号的时间：**

|  |
| --- |
| SELECT MAKEDATE(YEAR(CURDATE()), 1) AS first\_day\_of\_year; |

**Mysql查找所有父级目录：**

|  |
| --- |
| SELECT @t AS **\_id** ,  **(**SELECT @t := **fjid** FROM b\_zcfl\_gx WHERE jdid = **\_id)** AS fjid FROM  ***(***SELECT @t :='5df01ab758c5417c800585a9fa84605a' ***)*** vars , b\_zcfl\_gx h **WHERE** @t != '' and @t is not **null** |

**Mysql行号：**

|  |
| --- |
| SELECT @**rownum** := @**rownum** + 1 AS rownum FROM  ***(*** SELECT @**rownum** := 0 ***) r*** |

**Java8流分组：**

|  |
| --- |
| Map<Object, List<HashMap<String, Object>>> map= list.stream**()**.collect**(**Collectors.*groupingBy***(**map -> map.get**(**"name"**)))**; |

**Mysql时间转换：**

|  |
| --- |
| 日期转时间戳： unix\_timestamp(now()) 时间戳转日期： from\_unixtime(1632902493)  时间转字符串：  DATE\_FORMAT(update\_time,'%Y-%m-%d %H:%i:%s') |

**Mysql 查询进程：**

|  |
| --- |
| show **processlist**;  kill pid 结束进程 |

**Mysql 查询表结构：**

|  |
| --- |
| SELECT TABLE\_NAME **TABLES**, COLUMN\_NAME **字段**, COLUMN\_TYPE **类型**, CHARACTER\_MAXIMUM\_LENGTH **长度**, IS\_NULLABLE **是否为空**, COLUMN\_COMMENT 字段说明 FROM INFORMATION\_SCHEMA.COLUMNS  WHERE TABLE\_NAME = **(**'表名'**)**  AND **table\_schema** = 'dldp\_pwy\_v2' |

**Mysql 查询数据库的表（表名筛选）：**

|  |
| --- |
| SELECT \* FROM information\_schema.`TABLES` WHERE TABLE\_NAME LIKE '%history\_version%' AND **table\_schema** = 'dldp\_pwy\_v2' |

**Mysql查询表主键：**

|  |
| --- |
| SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS  WHERE table\_name = 'd\_main\_project\_header\_n' AND **table\_schema** = 'dldp\_pwy\_v2' AND **column\_key** = 'PRI' |

**Mysql查询事务死锁：**

|  |
| --- |
| SELECT \* FROM information\_schema.INNODB\_TRX; |

**Mysql GBK编码：**

|  |
| --- |
| CONVERT(a.devicename USING gbk) COLLATE gbk\_chinese\_ci |

**Mysql查看存过：**

|  |
| --- |
| //所有存过  SHOW PROCEDURE STATUS;  //查看存过  SHOW CREATE PROCEDURE name |

**Mysql计算日期差值：**

|  |
| --- |
| SELECT DATEDIFF('2021-01-01', '2022-1-31') AS days\_diff; |

**Mysql计算今年已过天数：**

|  |
| --- |
| SELECT DATEDIFF(LAST\_DAY(CURDATE()), CURDATE()) AS days\_left\_in\_year;  SELECT DAYOFYEAR(CURDATE()) AS day\_of\_year; |

**Mysql计算今年总天数：**

|  |
| --- |
| SELECT DAYOFYEAR(CONCAT(YEAR(NOW()),'-12-31')) |

**Mysql 去除字符串的逗号：**

|  |
| --- |
| //最后  TRIM(TRAILING ',' FROM '1,2,3,')  //两头  TRIM(BOTH ',' FROM ',1,2,3,')  //开头  TRIM(LEADING',' FROM ',1,2,3,') |

**Java 二维数组转一维数组：**

|  |
| --- |
| int[][] tiles = new int[][]{  {1,2,3},{4,5,6},{7,8,0} }; int[] array = Arrays.stream(tiles).flatMapToInt(Arrays::stream).toArray(); |

**Hikari连接池sql异常事务失效：**

|  |
| --- |
| 两个参数不能同时设置成true。优先useLocalTransactionState: false  useServerPrepStmts = false，使 PrepareStatement 也会降级成文本协议而不是二进制协议  spring: datasource:  hikari:  data-source-properties:  useServerPrepStmts: true  useLocalTransactionState: false |