# **条件构造器**

**说明:**

* 以下出现的第一个入参boolean condition表示该条件****是否****加入最后生成的sql中
* 以下代码块内的多个方法均为从上往下补全个别boolean类型的入参,默认为true
* 以下出现的泛型Param均为Wrapper的子类实例(均具有AbstractWrapper的所有方法)
* 以下方法在入参中出现的R为泛型,在普通wrapper中是String,在LambdaWrapper中是****函数****(例:Entity::getId,Entity为实体类,getId为字段id的****getMethod****)
* 以下方法入参中的R column均表示数据库字段,当R具体类型为String时则为数据库字段名(****字段名是数据库关键字的自己用转义符包裹!****)!而不是实体类数据字段名!!!,另当R具体类型为SFunction时项目runtime不支持eclipse自家的编译器!!!
* 以下举例均为使用普通wrapper,入参为Map和List的均以json形式表现!
* 使用中如果入参的Map或者List为****空****,则不会加入最后生成的sql中!!!
* 有任何疑问就点开源码看,看不懂****函数****的[点击我学习新知识](https://www.jianshu.com/p/613a6118e2e0" \t "https://mp.baomidou.com/guide/_blank)

**警告:**

不支持以及不赞成在 RPC 调用中把 Wrapper 进行传输

1. wrapper 很重
2. 传输 wrapper 可以类比为你的 controller 用 map 接收值(开发一时爽,维护火葬场)
3. 正确的 RPC 调用姿势是写一个 DTO 进行传输,被调用方再根据 DTO 执行相应的操作
4. 我们拒绝接受任何关于 RPC 传输 Wrapper 报错相关的 issue 甚至 pr

## [#](https://mp.baomidou.com/guide/wrapper.html" \l "abstractwrapper)**AbstractWrapper**

**说明:**

QueryWrapper(LambdaQueryWrapper) 和 UpdateWrapper(LambdaUpdateWrapper) 的父类  
用于生成 sql 的 where 条件, entity 属性也用于生成 sql 的 where 条件  
注意: entity 生成的 where 条件与 使用各个 api 生成的 where 条件****没有任何关联行为****

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "alleq)**allEq**

allEq(Map<R, V> params)allEq(Map<R, V> params, boolean null2IsNull)allEq(boolean condition, Map<R, V> params, boolean null2IsNull)

* 全部[eq](https://mp.baomidou.com/guide/wrapper.html" \l "eq)(或个别[isNull](https://mp.baomidou.com/guide/wrapper.html" \l "isnull))

**个别参数说明:**

params : key为数据库字段名,value为字段值  
null2IsNull : 为true则在map的value为null时调用 [isNull](https://mp.baomidou.com/guide/wrapper.html" \l "isnull) 方法,为false时则忽略value为null的

* 例1: allEq({id:1,name:"老王",age:null})--->id = 1 and name = '老王' and age is null
* 例2: allEq({id:1,name:"老王",age:null}, false)--->id = 1 and name = '老王'

allEq(BiPredicate<R, V> filter, Map<R, V> params)allEq(BiPredicate<R, V> filter, Map<R, V> params, boolean null2IsNull)allEq(boolean condition, BiPredicate<R, V> filter, Map<R, V> params, boolean null2IsNull)

**个别参数说明:**

filter : 过滤函数,是否允许字段传入比对条件中  
params 与 null2IsNull : 同上

* 例1: allEq((k,v) -> k.indexOf("a") > 0, {id:1,name:"老王",age:null})--->name = '老王' and age is null
* 例2: allEq((k,v) -> k.indexOf("a") > 0, {id:1,name:"老王",age:null}, false)--->name = '老王'

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "eq)**eq**

eq(R column, Object val)eq(boolean condition, R column, Object val)

* 等于 =
* 例: eq("name", "老王")--->name = '老王'

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "ne)**ne**

ne(R column, Object val)ne(boolean condition, R column, Object val)

* 不等于 <>
* 例: ne("name", "老王")--->name <> '老王'

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "gt)**gt**

gt(R column, Object val)gt(boolean condition, R column, Object val)

* 大于 >
* 例: gt("age", 18)--->age > 18

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "ge)**ge**

ge(R column, Object val)ge(boolean condition, R column, Object val)

* 大于等于 >=
* 例: ge("age", 18)--->age >= 18

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "lt)**lt**

lt(R column, Object val)lt(boolean condition, R column, Object val)

* 小于 <
* 例: lt("age", 18)--->age < 18

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "le)**le**

le(R column, Object val)le(boolean condition, R column, Object val)

* 小于等于 <=
* 例: le("age", 18)--->age <= 18

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "between)**between**

between(R column, Object val1, Object val2)between(boolean condition, R column, Object val1, Object val2)

* BETWEEN 值1 AND 值2
* 例: between("age", 18, 30)--->age between 18 and 30

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "notbetween)**notBetween**

notBetween(R column, Object val1, Object val2)notBetween(boolean condition, R column, Object val1, Object val2)

* NOT BETWEEN 值1 AND 值2
* 例: notBetween("age", 18, 30)--->age not between 18 and 30

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "like)**like**

like(R column, Object val)like(boolean condition, R column, Object val)

* LIKE '%值%'
* 例: like("name", "王")--->name like '%王%'

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "notlike)**notLike**

notLike(R column, Object val)notLike(boolean condition, R column, Object val)

* NOT LIKE '%值%'
* 例: notLike("name", "王")--->name not like '%王%'

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "likeleft)**likeLeft**

likeLeft(R column, Object val)likeLeft(boolean condition, R column, Object val)

* LIKE '%值'
* 例: likeLeft("name", "王")--->name like '%王'

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "likeright)**likeRight**

likeRight(R column, Object val)likeRight(boolean condition, R column, Object val)

* LIKE '值%'
* 例: likeRight("name", "王")--->name like '王%'

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "isnull)**isNull**

isNull(R column)isNull(boolean condition, R column)

* 字段 IS NULL
* 例: isNull("name")--->name is null

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "isnotnull)**isNotNull**

isNotNull(R column)isNotNull(boolean condition, R column)

* 字段 IS NOT NULL
* 例: isNotNull("name")--->name is not null

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "in)**in**

in(R column, Collection<?> value)in(boolean condition, R column, Collection<?> value)

* 字段 IN (value.get(0), value.get(1), ...)
* 例: in("age",{1,2,3})--->age in (1,2,3)

in(R column, Object... values)in(boolean condition, R column, Object... values)

* 字段 IN (v0, v1, ...)
* 例: in("age", 1, 2, 3)--->age in (1,2,3)

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "notin)**notIn**

notIn(R column, Collection<?> value)notIn(boolean condition, R column, Collection<?> value)

* 字段 IN (value.get(0), value.get(1), ...)
* 例: notIn("age",{1,2,3})--->age not in (1,2,3)

notIn(R column, Object... values)notIn(boolean condition, R column, Object... values)

* 字段 NOT IN (v0, v1, ...)
* 例: notIn("age", 1, 2, 3)--->age not in (1,2,3)

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "insql)**inSql**

inSql(R column, String inValue)inSql(boolean condition, R column, String inValue)

* 字段 IN ( sql语句 )
* 例: inSql("age", "1,2,3,4,5,6")--->age in (1,2,3,4,5,6)
* 例: inSql("id", "select id from table where id < 3")--->id in (select id from table where id < 3)

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "notinsql)**notInSql**

notInSql(R column, String inValue)notInSql(boolean condition, R column, String inValue)

* 字段 NOT IN ( sql语句 )
* 例: notInSql("age", "1,2,3,4,5,6")--->age not in (1,2,3,4,5,6)
* 例: notInSql("id", "select id from table where id < 3")--->age not in (select id from table where id < 3)

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "groupby)**groupBy**

groupBy(R... columns)groupBy(boolean condition, R... columns)

* 分组：GROUP BY 字段, ...
* 例: groupBy("id", "name")--->group by id,name

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "orderbyasc)**orderByAsc**

orderByAsc(R... columns)orderByAsc(boolean condition, R... columns)

* 排序：ORDER BY 字段, ... ASC
* 例: orderByAsc("id", "name")--->order by id ASC,name ASC

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "orderbydesc)**orderByDesc**

orderByDesc(R... columns)orderByDesc(boolean condition, R... columns)

* 排序：ORDER BY 字段, ... DESC
* 例: orderByDesc("id", "name")--->order by id DESC,name DESC

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "orderby)**orderBy**

orderBy(boolean condition, boolean isAsc, R... columns)

* 排序：ORDER BY 字段, ...
* 例: orderBy(true, true, "id", "name")--->order by id ASC,name ASC

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "having)**having**

having(String sqlHaving, Object... params)having(boolean condition, String sqlHaving, Object... params)

* HAVING ( sql语句 )
* 例: having("sum(age) > 10")--->having sum(age) > 10
* 例: having("sum(age) > {0}", 11)--->having sum(age) > 11

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "or)**or**

or()or(boolean condition)

* 拼接 OR

**注意事项:**

主动调用or表示紧接着下一个****方法****不是用and连接!(不调用or则默认为使用and连接)

* 例: eq("id",1).or().eq("name","老王")--->id = 1 or name = '老王'

or(Function<Param, Param> func)or(boolean condition, Function<Param, Param> func)

* OR 嵌套
* 例: or(i -> i.eq("name", "李白").ne("status", "活着"))--->or (name = '李白' and status <> '活着')

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "and)**and**

and(Function<Param, Param> func)and(boolean condition, Function<Param, Param> func)

* AND 嵌套
* 例: and(i -> i.eq("name", "李白").ne("status", "活着"))--->and (name = '李白' and status <> '活着')

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "nested)**nested**

nested(Function<Param, Param> func)nested(boolean condition, Function<Param, Param> func)

* 正常嵌套 不带 AND 或者 OR
* 例: nested(i -> i.eq("name", "李白").ne("status", "活着"))--->(name = '李白' and status <> '活着')

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "apply)**apply**

apply(String applySql, Object... params)apply(boolean condition, String applySql, Object... params)

* 拼接 sql

**注意事项:**

该方法可用于数据库****函数**** 动态入参的params对应前面applySql内部的{index}部分.这样是不会有sql注入风险的,反之会有!

* 例: apply("id = 1")--->id = 1
* 例: apply("date\_format(dateColumn,'%Y-%m-%d') = '2008-08-08'")--->date\_format(dateColumn,'%Y-%m-%d') = '2008-08-08'")
* 例: apply("date\_format(dateColumn,'%Y-%m-%d') = {0}", "2008-08-08")--->date\_format(dateColumn,'%Y-%m-%d') = '2008-08-08'")

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "last)**last**

last(String lastSql)last(boolean condition, String lastSql)

* 无视优化规则直接拼接到 sql 的最后

**注意事项:**

只能调用一次,多次调用以最后一次为准 有sql注入的风险,请谨慎使用

* 例: last("limit 1")

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "exists)**exists**

exists(String existsSql)exists(boolean condition, String existsSql)

* 拼接 EXISTS ( sql语句 )
* 例: exists("select id from table where age = 1")--->exists (select id from table where age = 1)

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "notexists)**notExists**

notExists(String notExistsSql)notExists(boolean condition, String notExistsSql)

* 拼接 NOT EXISTS ( sql语句 )
* 例: notExists("select id from table where age = 1")--->not exists (select id from table where age = 1)

## [#](https://mp.baomidou.com/guide/wrapper.html" \l "querywrapper)**QueryWrapper**

**说明:**

继承自 AbstractWrapper ,自身的内部属性 entity 也用于生成 where 条件  
及 LambdaQueryWrapper, 可以通过 new QueryWrapper().lambda() 方法获取

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "select)**select**

select(String... sqlSelect)select(Predicate<TableFieldInfo> predicate)select(Class<T> entityClass, Predicate<TableFieldInfo> predicate)

* 设置查询字段

**说明:**

以上方分法为两类.  
第二类方法为:过滤查询字段(主键除外),入参不包含 class 的调用前需要wrapper内的entity属性有值! 这两类方法重复调用以最后一次为准

* 例: select("id", "name", "age")
* 例: select(i -> i.getProperty().startsWith("test"))

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "excludecolumns)**excludeColumns @Deprecated**

* 排除查询字段

已从3.0.5版本上移除此方法!

## [#](https://mp.baomidou.com/guide/wrapper.html" \l "updatewrapper)**UpdateWrapper**

**说明:**

继承自 AbstractWrapper ,自身的内部属性 entity 也用于生成 where 条件  
及 LambdaUpdateWrapper, 可以通过 new UpdateWrapper().lambda() 方法获取!

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "set)**set**

set(String column, Object val)set(boolean condition, String column, Object val)

* SQL SET 字段
* 例: set("name", "老李头")
* 例: set("name", "")--->数据库字段值变为****空字符串****
* 例: set("name", null)--->数据库字段值变为null

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "setsql)**setSql**

setSql(String sql)

* 设置 SET 部分 SQL
* 例: setSql("name = '老李头')

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "lambda)**lambda**

* 获取 LambdaWrapper  
  在QueryWrapper中是获取LambdaQueryWrapper  
  在UpdateWrapper中是获取LambdaUpdateWrapper

## [#](https://mp.baomidou.com/guide/wrapper.html" \l "%E4%BD%BF%E7%94%A8-wrapper-%E8%87%AA%E5%AE%9A%E4%B9%89sql)**使用 Wrapper 自定义SQL**

**需求来源:**

在使用了mybatis-plus之后, 自定义SQL的同时也想使用Wrapper的便利应该怎么办？ 在mybatis-plus版本3.0.7得到了完美解决 版本需要大于或等于3.0.7, 以下两种方案取其一即可

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "service-java)**Service.java**

mysqlMapper.getAll(Wrappers.<MysqlData>lambdaQuery().eq(MysqlData::getGroup, 1));

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "%E6%96%B9%E6%A1%88%E4%B8%80-%E6%B3%A8%E8%A7%A3%E6%96%B9%E5%BC%8F-mapper-java)**方案一 注解方式 Mapper.java**

@Select("select \* from mysql\_data ${ew.customSqlSegment}")

List<MysqlData> getAll(@Param(Constants.WRAPPER) Wrapper wrapper);

### [#](https://mp.baomidou.com/guide/wrapper.html" \l "%E6%96%B9%E6%A1%88%E4%BA%8C-xml%E5%BD%A2%E5%BC%8F-mapper-xml)**方案二 XML形式 Mapper.xml**

<select id="getAll" resultType="MysqlData">

SELECT \* FROM mysql\_data ${ew.customSqlSegment}</select>