Atitit 性能提升 读取缓存 cacheutil attilax艾提拉总结

## 写入缓存使用 多线程。读取缓存也可以结合多线程，或者直接缓存比较简单些。。

[getParameter9999\_9999\_12={val=0, uptime=1523865249283, key=getParameter9999\_9999\_12}, getParameter100\_105002\_375={val=0, uptime=1523865249254, key=getParameter100\_105002\_375}, getParameter100\_105002\_244={val=0, uptime=1523865249287, key=getParameter100\_105002\_244}, getParameter100\_105002\_258={val=0, uptime=1523865190197, key=getParameter100\_105002\_258}, getParameter100\_105002\_383={val=0, uptime=1523865190199, key=getParameter100\_105002\_383},

getParameter9999\_9999\_7={val=0, uptime=1523865249280, key=getParameter9999\_9999\_7}]

getParameter100\_105002\_383

## Mybatis缓存设置

## 自定义缓存

**public** String getParameterCacheVer(Integer productId, Integer programId, Integer paramNo) **throws** Exception {

String **cacheKey**="getParameter"+productId+programId+paramNo;

**try** {

**return** (String) CacheUtil.*getValFromCache*(cacheKey);

} **catch** (CantfindEx **e**) {

HashMap **map** = **new** HashMap();

map.put("productId", productId);

map.put("programId", programId);

map.put("paramNo", paramNo);

String **parameter** = cliSendAdviceDao.getParameter(map);

CacheUtil.*setValInCache*(cacheKey,parameter);

**return** parameter;

}

}

**package** com.cnhis.cloudhealth.clinical.util;

**import** java.util.HashMap;

**import** java.util.Map;

**import** com.google.common.collect.Maps;

**public** **class** CacheUtil {

**public** **static** Map *cacheMap\_index* = Maps.*newConcurrentMap*();

**public** **static** Object getValFromCache(String cacheKey) **throws** CantfindEx {

Map **findCacheRztMap** = (Map) *cacheMap\_index*.get(cacheKey);

**if** (findCacheRztMap != **null**) {

Long **uptime** = (Long) findCacheRztMap.get("uptime");

Long **now** = System.*currentTimeMillis*();

**if** (now - uptime < 60 \* 000) // 60s

{

**return** (String) findCacheRztMap.get("val");

}

}

**throw** **new** CantfindEx();

}

**public** **static** **void** setValInCache(String cacheKey, String parameter) {

String **val** = parameter;

Long **now** = System.*currentTimeMillis*();

HashMap **mapVal** = **new** HashMap();

mapVal.put("key", cacheKey);

mapVal.put("val", val);

mapVal.put("uptime", now);

*cacheMap\_index*.put(cacheKey, mapVal);

}

}