

nodejs 实现新增时间段交集判断

作者: [XPPA](#)

原文链接: <https://ld246.com/article/1537169562677>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

功能

传入多个不想交的时间段（开始、结束时间）及新加入的开始、结束时间，判断新加入的时间与现有时间是否有交集。数据的校验应该在上层就已经完成，默认进入这里的数据都是合法的，比如不会出现结束时间早于开始时间的情况。

```
/**
 * 构造checkTimesIsIntersect用时间对象
 * @param time 传入的时间
 * @param label 时间标签 start/end
 * @returns {{time: *, label: *}}
 */
function constructTimeObj(time, label) {
  return {time, label};
}

/**
 * 校验新增时间段是否和现有时间段有交集
 * @param {Object[]} timeInfos 时间数组对象
 * @param {Number} timeInfos.startTime 开始时间
 * @param {Number} timeInfos.endTime 结束时间
 * @param {Number} startTime 新加入开始时间
 * @param {Number} endTime 新加入结束时间
 */
function checkTimesIsIntersect(timeInfos, startTime, endTime) {
  const timeObjs = [
    constructTimeObj(startTime, 'start'),
    constructTimeObj(endTime, 'end'),
  ];
  _forEach(timeInfos, (v) => {
    timeObjs.push(constructTimeObj(v.startTime, 'start'));
    timeObjs.push(constructTimeObj(v.endTime, 'end'));
  });
  // 升序
  const orderedTimeObjs = timeObjs.sort((a, b) => {
    return a.time - b.time;
  });
  _forEach(orderedTimeObjs, (v, k) => {
    // 最后一个元素不做判断
    if (orderedTimeObjs.length === k + 1) {
      return;
    }
    // 判断相邻时间秒数是否相同
    if (parseInt(v.time / 1000, 10) === parseInt(orderedTimeObjs[k + 1].time / 1000, 10)) {
      // throw new MError(MError.GROUPCOMMODITY_TIME_CONFLICT);
    }
    // 判断相邻标签是否相同
    if (v.label === orderedTimeObjs[k + 1].label) {
      // throw new MError(MError.GROUPCOMMODITY_TIME_CONFLICT);
    }
  });
}
```