

Bug report: jquery-dateFormat.js parseTime() zeroes the time in colon-containing timezones

File: platform/core/webapp/internal/jQuery/jquery-dateFormat.js — parseTime() (and its caller parseDate())

Symptom: DateFormat.format.date(dateObj, '...HH:mm...') returns 00 for hour/minute/second — silently, no error — for browsers/timezones whose Date.toTimeString() renders a timezone label containing a colon (e.g. Honolulu → (GMT-10:00)). Date-only formats (yyyy-MM-dd) are unaffected. Fails on Firefox in Honolulu, passes in e.g. Tokyo (Japan Standard Time) — no colon).

Root cause: For a Date, parseDate() builds the time string from value.toTimeString():

```
parsedDate.time = parseTime(value.toTimeString()) + "." + value.getMilliseconds();
```

```
// e.g. "00:00:00 GMT-1000 (GMT-10:00).0"
```

parseTime() then splits the entire string on ':':

```
timeArray = time.split(':');    // <-- the bug
```

```
if (timeArray.length === 3) { ... } // expects exactly HH:MM:SS
```

```
return { hour:", minute:", second:", ... }; // otherwise bails to empty
```

It assumes the only colons come from HH:MM:SS. When the trailing timezone label also has a colon (GMT-10:00), the split yields 4 parts, length === 3 is false, and it returns empty hour/minute. format.date then does padding(", 2) → "00", so every time renders as 00:00.

Why it broke our page: Two datetime-local inputs (reservation start/end) were populated via DateFormat.format.date(...). Both times came back 00:00, collapsing start == end → server rejected with "StartTime must be before EndTime." It only reproduced on the Honolulu TeamCity agent because that's where the tz label carries a colon.

Fix (one line): isolate the clock portion before splitting on : — drop everything after the first space (the tz label) first:

```
// before
```

```
timeArray = time.split(':');
```

```
// after
```

```
timeArray = time.split(' ')[0].split(':'); // strip trailing tz label like " GMT-1000 (GMT-10:00)"
```

time.split(' ')[0] is "00:00:00" for a toTimeString(), and is a no-op for the other inputs
parseTime handles ("10:54:50.546", "10:43:41"), so it's low-risk and keeps the existing
length === 3 path.

Blast radius / why worth fixing centrally: any DateFormat.format.date call that includes a time token (HH, mm, ss, hh) is affected in colon-label timezones. In our module alone it also silently mis-renders the calendar event labels, hover tooltips, and cost-table times — we only noticed via the datetime-local inputs because there it caused a hard test failure instead of just wrong display.