

CalendarServiceTester

August 18, 2026

CalendarServiceTester

Kind: Class

Source: Outlook_plugin/src/calendar/comprehensive-test.ts (line 7)

Part of: [Calendar](#)

Comprehensive Test Suite for Calendar Service

This ensures no functionality is broken in the optimized version

`CalendarServiceTester` runs checks against calendar service behavior and reports the results. It verifies exported functions, event shapes, date handling, batch limits, error paths, and mock synchronization before producing a test report.

Methods

Method	Signature	Returns
<code>testExportedFunctions</code>	<code>testExportedFunctions()</code>	<code>boolean</code>
<code>testEventObjectStructure</code>	<code>testEventObjectStructure()</code>	<code>boolean</code>
<code>testDateHandling</code>	<code>testDateHandling()</code>	<code>boolean</code>
<code>testBatchSizeLimits</code>	<code>testBatchSizeLimits()</code>	<code>boolean</code>
<code>testErrorHandling</code>	<code>testErrorHandling()</code>	<code>boolean</code>
<code>testMockSync</code>	<code>testMockSync()</code>	<code>Promise<boolean></code>
<code>generateReport</code>	<code>generateReport()</code>	<code>void</code>
<code>runAllTests</code>	<code>runAllTests()</code>	<code>Promise<void></code>

When something fails

- `CalendarServiceTester` handles failure in 1 place: it turns it into a return value in all
 -

Diagram

```
graph LR
  A[CalendarServiceTester] --> B[testExportedFunctions]
  A --> C[testEventObjectStructure]
  A --> D[testDateHandling]
  A --> E[testBatchSizeLimits]
  A --> F[testErrorHandling]
  A --> G[testMockSync]
  A --> H[generateReport]
  A --> I[runAllTests]

  I --> B
  I --> C
  I --> D
  I --> E
  I --> F
  I --> G
  I --> H
```

Usage

```
import { CalendarServiceTester } from "../calendar/comprehensive-test";

async function verifyCalendarService() {
  const tester = new CalendarServiceTester();

  await tester.runAllTests();

  const exportsValid = tester.testExportedFunctions();
  const datesValid = tester.testDateHandling();

  if (!exportsValid || !datesValid) {
    throw new Error("Calendar service checks failed.");
  }
}

void verifyCalendarService();
```

AI Coding Instructions

- Keep synchronous checks returning `boolean` and preserve the asynchronous `Promise<boolean>` contract for mock synchronization.
- Add new service checks through `runAllTests()` so they are included before `generateReport()` runs.

- Test date values with explicit timezone and boundary cases to avoid environment-dependent failures.
- Keep batch-limit tests aligned with the calendar service limits rather than duplicating unrelated limits.