Got it ✅ — you want a **TypeScript** solution that formats a **`Date`** into a **locale-aware string** with:

1. **Day name abbreviation** (e.g., **`Mon`**)
2. **Day number** (e.g., **`1`**)
3. **Month abbreviation** (e.g., **`Jan`**)
4. **Custom timezone** (not just the system default)

We can achieve this using **`toLocaleDateString`** with **`Intl.DateTimeFormat`** options.

TypeScript Example

```typescript
// date-format.ts
function formatDateWithTimezone(
  date: Date,
  timeZone: string,
  locale: string = "en-US"
): string {
  try {
    // Validate timezone
    Intl.DateTimeFormat(undefined, { timeZone }).format(date);

    // Format: DayNameAbbrev DayNumber MonthAbbrev
    return date.toLocaleDateString(locale, {
      weekday: "short", // e.g., Mon
      day: "numeric",   // e.g., 1
      month: "short",   // e.g., Jan
      timeZone          // Custom timezone
    });
  } catch (error) {
    console.error("Invalid timezone or date:", error);
    return "";
  }
}

// Example usage:
const date = new Date("2026-01-01T12:00:00Z");
```

```
console.log(formatDateWithTimezone(date, "America/New_York")); // Thu, Jan 1
console.log(formatDateWithTimezone(date, "Europe/Rome"));      // Thu, Jan 1
console.log(formatDateWithTimezone(date, "Asia/Tokyo"));       // Thu, Jan 1
```

How it works

1. **`weekday: "short"`** → Abbreviated day name (**`Mon`**, **`Tue`**, etc.).
2. **`day: "numeric"`** → Day number without leading zero.
3. **`month: "short"`** → Abbreviated month name (**`Jan`**, **`Feb`**, etc.).
4. **`timeZone`** → Forces formatting in the specified timezone.
5. **Validation** → The **`Intl.DateTimeFormat`** call ensures the timezone is valid before formatting.

Example Output

For **`2026-01-01T12:00:00Z`**:

Timezone	Output
America/New_York	Thu, Jan 1
Europe/Rome	Thu, Jan 1
Asia/Tokyo	Thu, Jan 1

If you want **exactly `"Thu 1 Jan"`** without commas, we can strip them:

```
return date
  .toLocaleDateString(locale, {
    weekday: "short",
    day: "numeric",
    month: "short",
    timeZone
  })
  .replace(/,/g, "");
```

Do you want me to make a **version that always outputs in `Thu 1 Jan` format regardless of locale punctuation**? That would make it consistent across browsers.