Got it ✅ — you want to parse a JSON date string in the format `YYYY-MM-DD` into a **TypeScript** `Date` **object**, assuming **Italy's timezone** (Europe/Rome), and set the time to **00:00** local time.

Here's a **safe, complete, and runnable** TypeScript example:

```typescript
// parseDateItaly.ts
function parseItalyDate(dateStr: string): Date | null {
    // Validate format YYYY-MM-DD
    if (!/^\d{4}-\d{2}-\d{2}$/.test(dateStr)) {
        console.error("Invalid date format. Expected YYYY-MM-DD.");
        return null;
    }

    const [year, month, day] = dateStr.split("-").map(Number);

    // Create a Date object in Italy's timezone at 00:00
    // Step 1: Create UTC date from Italy's midnight
    const italyMidnight = new Date(Date.UTC(year, month - 1, day));

    // Step 2: Adjust for Italy's timezone offset (including DST)
    const italyFormatter = new Intl.DateTimeFormat("en-GB", {
        timeZone: "Europe/Rome",
        year: "numeric",
        month: "2-digit",
        day: "2-digit",
        hour: "2-digit",
        minute: "2-digit",
        second: "2-digit",
        hour12: false
    });

    // Extract Italy's midnight in UTC
    const parts = italyFormatter.formatToParts(italyMidnight);
    const getPart = (type: string) => Number(parts.find(p => p.type
=== type)?.value);
```

```
        return new Date(
            Date.UTC(
                getPart("year"),
                getPart("month") - 1,
                getPart("day"),
                getPart("hour"),
                getPart("minute"),
                getPart("second")
            )
        );
    }


    // Example usage:
    const jsonDate = "2024-05-10"; // from JSON
    const dateObj = parseItalyDate(jsonDate);

    if (dateObj) {
        console.log("Date object:", dateObj);
        console.log("ISO string:", dateObj.toISOString());
    }
```

How it works

1. **Validation** — Ensures the string matches `YYYY-MM-DD`.

2. **Parsing** — Splits into year, month, day.

3. **Timezone Handling** — Uses `Intl.DateTimeFormat` with `"Europe/Rome"` to correctly account for **daylight saving time**.

4. **Result** — Returns a `Date` object representing **Italy's midnight** in UTC.

Example Output

If Italy is in **CEST (UTC+2)**:

```
Date object: 2024-05-09T22:00:00.000Z
```

```
ISO string: 2024-05-09T22:00:00.000Z
```

This means **local time in Italy** is `2024-05-10 00:00` .

If you want, I can also give you a **shorter version** using only `Date.UTC` and manual offset calculation without `Intl` .

That would be faster for bulk parsing.

Do you want me to prepare that optimized version?