

Date and Time for Programmers



Press Space for next page →



Objectives

Understand the way date and time are represented and the key notes for programmers

-  **The Globe** - world timezones
-  **Definitions** - UTC, GMT, Zulu, DST and UTC offsets
-  **Server Side talks** - azure storage, mongodb and sqlalchemy use case
-  **More of UTC** - UTC date and time formats
-  **Python Playground** - explore some date time utils in python
-  **API UX** - returning date and time: ISO8601, Unix Epoch, RFC and Custom formats
-  **Other Usecases** - different apps, different needs
-  **Recap and Questions** - summary and questions
-  **Resources** - links to resources

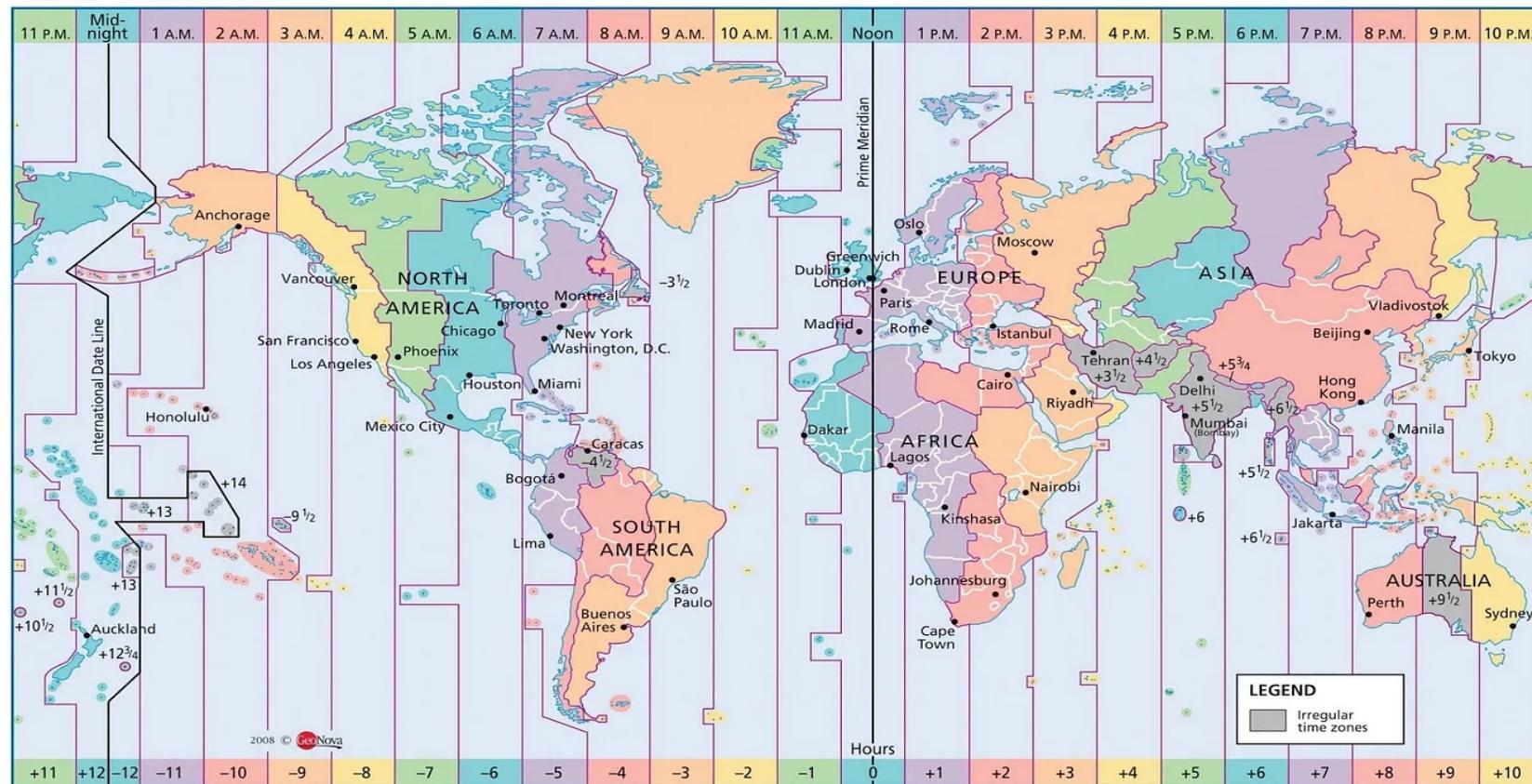
Let's begin 🔥

Table of contents

1. Introduction
2. Objectives
3. The Globe
4. Definitions
5. Storing time on the server
6. UTC date structure
7. Date Manipulation in Python
8. Returning Date from API: 1
9. Returning Date from API: 2
10. Returning Date from API: 3
11. Returning Date from API: 4
12. Thank you.

The Globe

WORLD TIME ZONES



Definitions

GMT	Greenwich Mean Time used to be the official standard, but it was calculated based on the average solar time, that means, it was calculated based on the rotation motion of the earth. <u>Greenwich (London) Royal Observatory</u>
UTC	Coordinated Universal Time (successor of GMT) is the primary time standard by which the world regulates clocks and time. UTC is calculated based on atomic time (more precise).
Z	More related to how Aviation/Military refers to 0 UTC offset
DST	Local time changes by political decisions. Increases/decreases UTC offsets by 1. BST, CEST
UTC offsets	UTC, UTC+1, UTC+2, UTC-5

* NB: UTC is represented as +00:00 or UTC+0 or Z

Storing time on the server

Most sources recommend storing dates in UTC. And manipulate based on timezone.

- Mongo DB
- SqlAlchemy
- Azure Storage

What is the Current UTC time?



UTC date structure

- 2020-01-27T06:00:00Z (Greenwich)
- 2020-01-27T00:00:00-06:00 (México City)
- 2020-01-27T14:00:00+08:00 (Beijing)

The Date format is YYY-MM-DD

T begins the time section

The Time format is HH-MM-SS

The ending is the timezone utc offset:

Z for UTC+0 or GMT

Date Manipulation in Python

Python has aware date time and naive datetime. This [-----](#) article mentions an upcoming change in the `datetime` package.

```
def aware_utcnow() → datetime:  
    """Generates a utc time with UTC timezone.  
    NB: only the naive is stored in the DB."""  
    return datetime.now(timezone.utc)  
  
def naive_utcnow() → datetime:  
    return aware_utcnow().replace(tzinfo=None)  
  
def convert_date_time_to_iso(value: datetime) → None:  
    # first convert date time to utc timezone or the user's timezone, then to iso format  
    # and return to user.  
    return value.toisoformat()
```

Do a demo with Python Shell and JS

Returning Date from API: 1

1. Use ISO8601

There are several formats to returning dates.
Depends on the usecase. But a popular and accepted format is the ISO8601 format.

- What is ISO8601 format?

- UTC date in various format

- UTC date in various format

A sample API response:

```
{  
  "created_at": "2024-01-28T15:41:26Z"  
}
```

In JS

```
new Date().toISOString()
```

In Python

```
from datetime import datetime, timezone  
datetime.now(timezone.utc).isoformat()
```

Returning Date from API: 2

2. Use ISO8601 but timezone as separate field

For example, Australia/Sydney You can find timezones here

- IANA timezone database

This is suitable for flight and virtual meetings

A sample API response:

```
{  
  "departure_time": "2023-09-24T11:00:00",  
  "departure_timezone": "Australia/Sydney"  
}
```

Returning Date from API: 3

3. Encode timezone offset with the datefield

If you need to put into account the specific date and timezone with DST inclusive.

This is suitable for gallery apps that could use reminders etc

A sample API response:

```
{  
    "taken_at": "2023-01-02T18:30:01-05:00"  
}
```

Returning Date from API: 4

4. Use Unix timestamp - sparingly

System dates imply a scientific, linear view of time. Most programming languages internally represent dates as the number of seconds that have elapsed since that have elapsed since 00:00:00 UTC on 1 January 1970. This is known as Unix time and is a sensible choice for representing a date as a number.

- Unix Time Stamp

This is suitable for flight and virtual meetings

A sample API response:

```
{  
    "created_at": 1682648103  
}
```

Thank you.

Resources
