DATA FORMATS(JSON, XML, SCHEMAS)

DATE:13.02.2024

aspire SYSTEMS

attention. always.

AGENDA

- ✔ **What is Data Format?**
- ✔ **Why we use data format ?**
- ✔ **What is JSON?**
- ✔ **What is XML?**
- ✔ **What is Schemas?**

DATA FORMAT

✓ A data format refers to the structure
and organization of data in a specific
way to facilitate storage, transmission,
and processing.

Why we use data format?

1. Interoperability
2. consistency
3. Efficiency

JSON

✓ JSON (JavaScript Object Notation) is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate. It is a text-based data format derived from JavaScript, but it is language-independent, meaning it can be used with most programming languages. JSON is often used to transmit data between a server and a web application.

WHY WE USE JSON & ADVANTAGES:

- ❑ **Easy to read and write.**
- ❑ **Language independence**
- ❑ **Web integration.**
- ❑ **API**
- ❑ **Data types are there in json.**

JSON CODE

```
{
"employees":[
    {"firstName":"Raju", "lastName":"Bai"},
    {"firstName":"Ajay", "lastName":"kumar"},
    {"firstName":"Peter", "lastName":"Jay"}
]
}
```

XML

✔ XML(Extensible Markup Language), is a text-based data format designed for storing and transporting structured information.

• XML uses tags to define elements and attributes.

• Syntax is similar to Html.

Tags and Elements:

•**Tags:** In XML, data is enclosed within tags. Tags are defined by using angle brackets (**<** and **>**). Tags come in pairs, with an opening tag and a closing tag. For example: **<name>** and **</name>**.

•**Elements:** The content between the opening and closing tags forms an element. Elements can contain text.

WHY WE USE XML & ADVANTAGES:

- **Versatility and Extensibility**
- **Platform-Independent**
- **Human-Readable**
- **Structured data representation**

XML CODE

```xml
<?xml version="1.0" encoding="UTF-8"?>
<data>
 <person>
   <name>Raju</name>
   <age>30</age>
   <city>New York</city>
 </person>
 <person>
   <name>Bheem</name>
   <age>25</age>
   <city>Paris</city>
 </person>
</data>
```

SCHEMAS

✓ A schema is a formal specification that defines the rules and structure, for organizing and validating data within a specific format.

✓ Schemas act as blueprints.

Three types of Schema:

- XML Schema Definition (XSD)
- JSON Schema
- Protocol Buffers Schema (Proto).

XML SCHEMA CODE

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="book">
    <xs:complexType>
     <xs:sequence>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="author" type="xs:string"/>
        <xs:element name="publicationYear" type="xs:positiveInteger"/>
        < xs:element name="isbn" type="xs:string"/>
     </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

JSON SCHEMA CODE

```
{
 "$schema": "http://json-schema.org/draft-07/schema#",
 "type": "object",
 "properties": {
  "name": { "type": "string" },
  "age": { "type": "integer" },
  "city": { "type": "string" },
  "isStudent": { "type": "boolean" },
  "hobbies": {
   "type": "array",
   "items": { "type": "string" },
  }
 },
 "required": ["name", "age", "city", "isStudent", "hobbies"]
}
```

PROTOCOL BUFFERS SCHEMA (PROTO)

```
syntax = "proto3";

message Person {
  string name = 1;
  int32 age = 2;
  string city = 3;
  repeated string hobbies = 4;
  bool isStudent = 5;
}
```

THANK YOU!