DATA607LAB7

Leslie Tavarez

2024-10-20

JSON Format: Pros: - Readable and easy to understand. - Widely supported in programming languages. - Lightweight compared to XML.

Cons: - Does not support comments. - Less efficient for very large datasets than binary formats.

```r
library(jsonlite)

# Create the data frame
data <- data.frame(
  Category = c("Electronics", "Electronics", "Home Appliances", "Home Appliances",
               "Clothing", "Clothing", "Clothing", "Clothing", "Books", "Books",
               "Books", "Books", "Sports Equipment", "Sports Equipment"),
  ItemName = c("Smartphone", "Smartphone", "Refrigerator", "Washing Machine",
               "T-Shirt", "T-Shirt", "T-Shirt", "Jeans", "Fiction Novel",
               "Fiction Novel", "Non-Fiction Guide", "Non-Fiction Guide",
               "Basketball", "Tennis Racket"),
  ItemID = c(101, 101, 201, 202, 301, 301, 301, 302, 401, 401, 402, 402, 501, 502),
  Brand = c("TechBrand", "TechBrand", "HomeCool", "CleanTech", "FashionCo",
            "FashionCo", "FashionCo", "DenimWorks", "-", "-", "-", "-",
            "SportsGear", "RacketPro"),
  Price = c(699.99, 699.99, 899.99, 499.99, 19.99, 19.99, 19.99, 49.99,
            14.99, 14.99, 24.99, 24.99, 29.99, 89.99),
  VariationID = c("101-A", "101-B", "201-A", "202-A", "301-A",
                  "301-B", "301-C", "302-A", "401-A", "401-B",
                  "402-A", "402-B", "501-A", "502-A"),
  VariationDetails = c("Color: Black, Storage: 64GB",
                       "Color: White, Storage: 128GB",
                       "Color: Stainless Steel, Capacity: 20 cu ft",
                       "Type: Front Load, Capacity: 4.5 cu ft",
                       "Color: Blue, Size: S",
                       "Color: Red, Size: M",
                       "Color: Green, Size: L",
                       "Color: Dark Blue, Size: 32",
                       "Format: Hardcover, Language: English",
                       "Format: Paperback, Language: Spanish",
                       "Format: eBook, Language: English",
                       "Format: Paperback, Language: French",
                       "Size: Size 7, Color: Orange",
                       "Material: Graphite, Color: Black")
)

# Convert to JSON
```

```r
json_data <- toJSON(data, pretty = TRUE)

# Print JSON data (optional)
cat(json_data)
```

```
## [
##   {
##     "Category": "Electronics",
##     "ItemName": "Smartphone",
##     "ItemID": 101,
##     "Brand": "TechBrand",
##     "Price": 699.99,
##     "VariationID": "101-A",
##     "VariationDetails": "Color: Black, Storage: 64GB"
##   },
##   {
##     "Category": "Electronics",
##     "ItemName": "Smartphone",
##     "ItemID": 101,
##     "Brand": "TechBrand",
##     "Price": 699.99,
##     "VariationID": "101-B",
##     "VariationDetails": "Color: White, Storage: 128GB"
##   },
##   {
##     "Category": "Home Appliances",
##     "ItemName": "Refrigerator",
##     "ItemID": 201,
##     "Brand": "HomeCool",
##     "Price": 899.99,
##     "VariationID": "201-A",
##     "VariationDetails": "Color: Stainless Steel, Capacity: 20 cu ft"
##   },
##   {
##     "Category": "Home Appliances",
##     "ItemName": "Washing Machine",
##     "ItemID": 202,
##     "Brand": "CleanTech",
##     "Price": 499.99,
##     "VariationID": "202-A",
##     "VariationDetails": "Type: Front Load, Capacity: 4.5 cu ft"
##   },
##   {
##     "Category": "Clothing",
##     "ItemName": "T-Shirt",
##     "ItemID": 301,
##     "Brand": "FashionCo",
##     "Price": 19.99,
##     "VariationID": "301-A",
##     "VariationDetails": "Color: Blue, Size: S"
##   },
##   {
##     "Category": "Clothing",
```

"ItemName": "T-Shirt",
"ItemID": 301,
"Brand": "FashionCo",
"Price": 19.99,
"VariationID": "301-B",
"VariationDetails": "Color: Red, Size: M"
},
{
"Category": "Clothing",
"ItemName": "T-Shirt",
"ItemID": 301,
"Brand": "FashionCo",
"Price": 19.99,
"VariationID": "301-C",
"VariationDetails": "Color: Green, Size: L"
},
{
"Category": "Clothing",
"ItemName": "Jeans",
"ItemID": 302,
"Brand": "DenimWorks",
"Price": 49.99,
"VariationID": "302-A",
"VariationDetails": "Color: Dark Blue, Size: 32"
},
{
"Category": "Books",
"ItemName": "Fiction Novel",
"ItemID": 401,
"Brand": "-",
"Price": 14.99,
"VariationID": "401-A",
"VariationDetails": "Format: Hardcover, Language: English"
},
{
"Category": "Books",
"ItemName": "Fiction Novel",
"ItemID": 401,
"Brand": "-",
"Price": 14.99,
"VariationID": "401-B",
"VariationDetails": "Format: Paperback, Language: Spanish"
},
{
"Category": "Books",
"ItemName": "Non-Fiction Guide",
"ItemID": 402,
"Brand": "-",
"Price": 24.99,
"VariationID": "402-A",
"VariationDetails": "Format: eBook, Language: English"
},
{
"Category": "Books",

```
##     "ItemName": "Non-Fiction Guide",
##     "ItemID": 402,
##     "Brand": "-",
##     "Price": 24.99,
##     "VariationID": "402-B",
##     "VariationDetails": "Format: Paperback, Language: French"
##   },
##   {
##     "Category": "Sports Equipment",
##     "ItemName": "Basketball",
##     "ItemID": 501,
##     "Brand": "SportsGear",
##     "Price": 29.99,
##     "VariationID": "501-A",
##     "VariationDetails": "Size: Size 7, Color: Orange"
##   },
##   {
##     "Category": "Sports Equipment",
##     "ItemName": "Tennis Racket",
##     "ItemID": 502,
##     "Brand": "RacketPro",
##     "Price": 89.99,
##     "VariationID": "502-A",
##     "VariationDetails": "Material: Graphite, Color: Black"
##   }
## ]
```

```r
# Write JSON to file
write(json_data, "inventory.json")
```

HTML Format   Pros of HTML - Readable: Easy to read and edit. - Web Compatibility: Works well on any web browser. - Clear Structure: Tables make data easy to understand. - Styling Options: Can be styled with CSS for better presentation. - Accessible: Viewable on various devices. Cons of HTML - Larger File Size: Can be bulkier than formats like JSON. - Limited for Processing: Not ideal for data analysis or manipulation. - Less Complex Structure: Can't handle hierarchical data as well as JSON or XML. - Static: Requires regeneration for dynamic updates. - Limited Tool Support: Many analysis tools don't work directly with HTML.

In summary, HTML is great for presentation but not optimal for data processing.

```r
library(knitr)
# Write data to HTML
html_data <- kable(data, format = "html", table.attr = "class='dataframe'")

# Save to HTML file
writeLines(html_data, "inventory.html")
```

XML Format   Pros of XML: - Readable: Easy to read and understand. - Self-Describing: Tags provide context for the data. - Flexible: Can handle complex and nested data structures. - Cross-Platform: Widely supported across various languages and platforms. - Validatable: Can be validated against schemas for structure. Cons of XML - Verbose: Larger file sizes due to extensive tagging. - Complex Parsing: More complicated and resource-intensive to parse. - Performance: Slower read/write operations compared to

lighter formats. - Text-Based: All data is treated as text, complicating data types. - Limited Structures: Doesn't natively support arrays or some modern data structures.

XML is great for structured data but can be bulky and slow for large datasets. For some reason it also gave me the most trouble when trying to code.

```r
library(XML)

data <- data.frame(
  Category = c("Electronics", "Electronics", "Home Appliances", "Home Appliances",
               "Clothing", "Clothing", "Clothing", "Clothing", "Books", "Books",
               "Books", "Books", "Sports Equipment", "Sports Equipment"),
  ItemName = c("Smartphone", "Smartphone", "Refrigerator", "Washing Machine",
               "T-Shirt", "T-Shirt", "T-Shirt", "Jeans", "Fiction Novel",
               "Fiction Novel", "Non-Fiction Guide", "Non-Fiction Guide",
               "Basketball", "Tennis Racket"),
  ItemID = c(101, 101, 201, 202, 301, 301, 301, 302, 401, 401, 402, 402, 501, 502),
  Brand = c("TechBrand", "TechBrand", "HomeCool", "CleanTech", "FashionCo",
            "FashionCo", "FashionCo", "DenimWorks", "-", "-", "-", "-",
            "SportsGear", "RacketPro"),
  Price = c(699.99, 699.99, 899.99, 499.99, 19.99, 19.99, 19.99, 49.99,
            14.99, 14.99, 24.99, 24.99, 29.99, 89.99),
  VariationID = c("101-A", "101-B", "201-A", "202-A", "301-A",
                  "301-B", "301-C", "302-A", "401-A", "401-B",
                  "402-A", "402-B", "501-A", "502-A"),
  VariationDetails = c("Color: Black, Storage: 64GB",
                       "Color: White, Storage: 128GB",
                       "Color: Stainless Steel, Capacity: 20 cu ft",
                       "Type: Front Load, Capacity: 4.5 cu ft",
                       "Color: Blue, Size: S",
                       "Color: Red, Size: M",
                       "Color: Green, Size: L",
                       "Color: Dark Blue, Size: 32",
                       "Format: Hardcover, Language: English",
                       "Format: Paperback, Language: Spanish",
                       "Format: eBook, Language: English",
                       "Format: Paperback, Language: French",
                       "Size: Size 7, Color: Orange",
                       "Material: Graphite, Color: Black")
)

# Create XML document
xml_data <- newXMLDoc()

# Create root node
root <- newXMLNode("Inventory", doc = xml_data)

# Create XML nodes using lapply and store them in a list
item_nodes <- lapply(1:nrow(data), function(i) {
  newXMLNode("Item",
             newXMLNode("Category", data$Category[i]),
             newXMLNode("ItemName", data$ItemName[i]),
             newXMLNode("ItemID", data$ItemID[i]),
             newXMLNode("Brand", data$Brand[i]),
```

```r
            newXMLNode("Price", data$Price[i]),
            newXMLNode("Variation",
                       newXMLNode("VariationID", data$VariationID[i]),
                       newXMLNode("VariationDetails", data$VariationDetails[i]))
  )
})

# Append all item nodes to root
addChildren(root, item_nodes)
```

```
## <Inventory>
##   <Item>
##     <Category>Electronics</Category>
##     <ItemName>Smartphone</ItemName>
##     <ItemID>101</ItemID>
##     <Brand>TechBrand</Brand>
##     <Price>699.99</Price>
##     <Variation>
##       <VariationID>101-A</VariationID>
##       <VariationDetails>Color: Black, Storage: 64GB</VariationDetails>
##     </Variation>
##   </Item>
##   <Item>
##     <Category>Electronics</Category>
##     <ItemName>Smartphone</ItemName>
##     <ItemID>101</ItemID>
##     <Brand>TechBrand</Brand>
##     <Price>699.99</Price>
##     <Variation>
##       <VariationID>101-B</VariationID>
##       <VariationDetails>Color: White, Storage: 128GB</VariationDetails>
##     </Variation>
##   </Item>
##   <Item>
##     <Category>Home Appliances</Category>
##     <ItemName>Refrigerator</ItemName>
##     <ItemID>201</ItemID>
##     <Brand>HomeCool</Brand>
##     <Price>899.99</Price>
##     <Variation>
##       <VariationID>201-A</VariationID>
##       <VariationDetails>Color: Stainless Steel, Capacity: 20 cu ft</VariationDetails>
##     </Variation>
##   </Item>
##   <Item>
##     <Category>Home Appliances</Category>
##     <ItemName>Washing Machine</ItemName>
##     <ItemID>202</ItemID>
##     <Brand>CleanTech</Brand>
##     <Price>499.99</Price>
##     <Variation>
##       <VariationID>202-A</VariationID>
##       <VariationDetails>Type: Front Load, Capacity: 4.5 cu ft</VariationDetails>
```

```
##      </Variation>
##    </Item>
##    <Item>
##      <Category>Clothing</Category>
##      <ItemName>T-Shirt</ItemName>
##      <ItemID>301</ItemID>
##      <Brand>FashionCo</Brand>
##      <Price>19.99</Price>
##      <Variation>
##        <VariationID>301-A</VariationID>
##        <VariationDetails>Color: Blue, Size: S</VariationDetails>
##      </Variation>
##    </Item>
##    <Item>
##      <Category>Clothing</Category>
##      <ItemName>T-Shirt</ItemName>
##      <ItemID>301</ItemID>
##      <Brand>FashionCo</Brand>
##      <Price>19.99</Price>
##      <Variation>
##        <VariationID>301-B</VariationID>
##        <VariationDetails>Color: Red, Size: M</VariationDetails>
##      </Variation>
##    </Item>
##    <Item>
##      <Category>Clothing</Category>
##      <ItemName>T-Shirt</ItemName>
##      <ItemID>301</ItemID>
##      <Brand>FashionCo</Brand>
##      <Price>19.99</Price>
##      <Variation>
##        <VariationID>301-C</VariationID>
##        <VariationDetails>Color: Green, Size: L</VariationDetails>
##      </Variation>
##    </Item>
##    <Item>
##      <Category>Clothing</Category>
##      <ItemName>Jeans</ItemName>
##      <ItemID>302</ItemID>
##      <Brand>DenimWorks</Brand>
##      <Price>49.99</Price>
##      <Variation>
##        <VariationID>302-A</VariationID>
##        <VariationDetails>Color: Dark Blue, Size: 32</VariationDetails>
##      </Variation>
##    </Item>
##    <Item>
##      <Category>Books</Category>
##      <ItemName>Fiction Novel</ItemName>
##      <ItemID>401</ItemID>
##      <Brand>-</Brand>
##      <Price>14.99</Price>
##      <Variation>
##        <VariationID>401-A</VariationID>
```

```
##          <VariationDetails>Format: Hardcover, Language: English</VariationDetails>
##        </Variation>
##    </Item>
##    <Item>
##      <Category>Books</Category>
##      <ItemName>Fiction Novel</ItemName>
##      <ItemID>401</ItemID>
##      <Brand>-</Brand>
##      <Price>14.99</Price>
##      <Variation>
##        <VariationID>401-B</VariationID>
##        <VariationDetails>Format: Paperback, Language: Spanish</VariationDetails>
##      </Variation>
##    </Item>
##    <Item>
##      <Category>Books</Category>
##      <ItemName>Non-Fiction Guide</ItemName>
##      <ItemID>402</ItemID>
##      <Brand>-</Brand>
##      <Price>24.99</Price>
##      <Variation>
##        <VariationID>402-A</VariationID>
##        <VariationDetails>Format: eBook, Language: English</VariationDetails>
##      </Variation>
##    </Item>
##    <Item>
##      <Category>Books</Category>
##      <ItemName>Non-Fiction Guide</ItemName>
##      <ItemID>402</ItemID>
##      <Brand>-</Brand>
##      <Price>24.99</Price>
##      <Variation>
##        <VariationID>402-B</VariationID>
##        <VariationDetails>Format: Paperback, Language: French</VariationDetails>
##      </Variation>
##    </Item>
##    <Item>
##      <Category>Sports Equipment</Category>
##      <ItemName>Basketball</ItemName>
##      <ItemID>501</ItemID>
##      <Brand>SportsGear</Brand>
##      <Price>29.99</Price>
##      <Variation>
##        <VariationID>501-A</VariationID>
##        <VariationDetails>Size: Size 7, Color: Orange</VariationDetails>
##      </Variation>
##    </Item>
##    <Item>
##      <Category>Sports Equipment</Category>
##      <ItemName>Tennis Racket</ItemName>
##      <ItemID>502</ItemID>
##      <Brand>RacketPro</Brand>
##      <Price>89.99</Price>
##      <Variation>
```

```
##        <VariationID>502-A</VariationID>
##        <VariationDetails>Material: Graphite, Color: Black</VariationDetails>
##      </Variation>
##    </Item>
## </Inventory>
```

```r
# Save XML to file
saveXML(xml_data, file = "inventory.xml")
```

```
## [1] "inventory.xml"
```

Parquet Format   Pros of Parquet: - Efficient Storage: Only reads necessary columns, speeding up access. - Compression: Smaller file sizes due to built-in compression. - Flexible Schema: Easily add new columns without affecting existing data. - Great for Big Data: Works well with large datasets in tools like Spark and Hadoop. - Strong Data Types: Supports complex data structures.

Cons of Parquet: - Not Easily Readable: Needs special tools to view or edit. - Overkill for Small Data: Better for large datasets; small ones don't benefit much. - Difficult to Edit: Not as straightforward as text formats like JSON or CSV. - Compatibility Issues: Not supported by all tools, unlike CSV or JSON.

```r
library(arrow)
```

```
##
## Attaching package: 'arrow'

## The following object is masked from 'package:utils':
##
##     timestamp
```

```r
# Write to Parquet file
write_parquet(data, "inventory.parquet")
```

Conclusion: HTML is a great format when displaying data online. XML is ideal for complex data structure that requires validation. JSON is best used for lightweight data interchange of web applications and APIs. Parquet should be used for large-scale data storage and analytics. All of these formats have drawbacks, we should choose format based on our needs.

Files Created

- Download inventory.json
- Download inventory.xml
- Download inventory.parquet
- View inventory.html