

Desenvolvimento de Componentes Distribuidos

Formatos de Arquivos

Leandro Duarte Pulgatti

2021

1 O que deve ser entregue

Para os três tipos de arquivos mais comuns utilizados para troca de dados entre aplicativos (.CSV, JSON e .XML) responda as seguintes perguntas:

1. Descreva brevemente como os dados são organizados no formato.

R: .CSV: CSV significa comma-separated-values, ou em português, valores separados por vírgulas, é o formato em que os campos serão separados por vírgulas, mas isso não é uma regra, pois os campos existentes em um formato csv podem ser separados por qualquer tipo de caracteres pouco comuns, como ponto e vírgula, tecla tab, tecla de espaço, etc.

.JSON: JSON significa JavaScript Object Notation, ou em português Notação de Objeto JavaScript, é um formato estruturado por meio de uma coleção de pares com nome e valor ou ser uma lista ordenada de valores. As chaves citadas acima são campos em que o conteúdo deve ser preenchido com uma string e ser delimitado por aspas, já o valor também representam os campos, mas esses podem conter dados do tipo string, array, objeto, número, booleano ou nulo e podem ser separados por vírgula.

.XML: XML significa EXtensible Markup Language, ou em português Linguagem Padronizada de Marcação, é um tipo de formato em que seus dados são organizados de forma hierárquica, sendo dividido em duas partes, uma chamada prolog que consiste em metadados administrativos, como instruções de processamento, declaração de tipo de documento ou comentários, já a parte chamada de body é composta por duas outras partes, a estrutural e o conteúdo, que pode ser um texto simples.

2. Quais as facilidades que este formato possui.

R: .CSV: A grande vantagem de se usar um arquivo com formato csv é que ele permite a importação e exportação de arquivos entre vários aplicativos, por ser um formato que muitos aplicativos podem ler.

.JSON: O JSON possui uma sintaxe simples, o que é de fácil aprendizado e implementação, seus arquivos geralmente tem um tamanho pequeno comparado com outras linguagens do mesmo patamar “intermediário”, como xml, também conta com uma facilidade de desenvolvimento e alto desempenho em aplicações que o utilizam.

.XML: XML possui alta integração com diferentes tipos de fontes, um fácil desenvolvimento por ser de fácil entendimento, fácil compressão e escalabilidade devido a ter uma estrutura hierarquizada.

3. Quais as dificuldades que este formato impõe.

R: .CSV: Não tem uma estrutura padronizada, o que pode ocasionar em erros ao exportar ou importa-lo por não reconhecer o que é de um campo e o que é de outro, assim como determinados caracteres.

.JSON: Por não possuir um esquema específico pode acabar por criar dados deformados, permite apenas um tipo de número, o formato de ponto flutuante de dupla precisão, não tem date type específico o que pode ocasionar em discrepância de formatação, não possui a possibilidade de colocar comentários, etc.

.XML: Por ter uma sintaxe redundante, acaba por ficar um documento maçante de ser analisado, essa redundância também pode afetar a eficiência de seus arquivos por ter arquivos mais pesados.

4. Em que situação você utilizaria este formato.

R: .CSV: Geralmente csv é utilizado em arquivos como Google Sheets, OpenOffice Calc, Apple Numbers e Microsoft Excel, pois esses programas são para trabalhar com números e se dão muito bem com o csv.

.JSON: Geralmente utilizado para transferir dados entre um servidor e o cliente por ser um formato mais leve do que outras linguagens.

.XML: Por possuir uma grande capacidade de guardar ou vincular dados de qualquer formato o xml é muito bom para ser utilizado em momentos em que você precisa garantir a integridade dos dados, como em sistemas de notas fiscais.

2 Sobre o trabalho

Individual

Método de entrega: Subir o arquivo gerado no formato DISTRIBUIDO_9_<NOME>
<MATRÍCULA> para seu github no projeto DesenvCompDist

Geralmente http://github.com/SEU_USUARIO/DesenvCompDist

Prazo: 23:59:59 horas do dia 27/04/2021

3 Demais Regras

- Os trabalhos são individuais.
- Os trabalhos não serão aceitos após a data/hora limite.

4 referencias

- <https://developers.google.com/identity/gsi/web>
- <https://developers.google.com/api-client-library/java/google-api-java-client/oauth2>
- <https://spring.io/guides/tutorials/spring-boot-oauth2/>
- <https://oauth.net/code/java/>
- <https://googleapis.dev/java/google-api-client/latest/com/google/api/client/googleapis/auth/oauth2/package-summary.html>