

Artigo

[Larissa Prussak](#) · Mar. 8, 2021 2min de leitura

## Comparação JSON básica.

Olá comunidade Dev

Pensei em compartilhar um pequeno método que juntei para comparar 2 objetos JSON para obter o equivalente básico. Atualmente, estou trabalhando em alguma migração de dados e queria uma verificação básica para validar se a saída JSON é basicamente equivalente entre o antigo e o novo, excluindo alguns itens como carimbos de data / hora.

É um pequeno método recursivo básico, que irá transparecer quaisquer diferenças sobre uma estrutura aninhada. É muito baixa tecnologia, pois é tudo o que preciso fazer, mas achei que poderia ser útil para outras pessoas.

Ele pode aceitar um objeto ou array JSON de origem e destino e, opcionalmente, um RefNo se você quiser executá-lo em lote e rastrear as ocorrências de diferenças em um ^zKeyTrap global. Além disso, há uma lista de exclusão para todas as chaves que sempre seriam diferentes, como ModificationTimestamps.

Sinta-se à vontade para usar e adaptar isso se for útil no uso do JSON.

```

ClassMethod CompareJSON(source As %String, target As %String, RefNo As
%String = 1) As %Boolean
{
set Identical = 1
Set propsIteratorsource = source.%GetIterator()
While (propsIteratorsource.%GetNext(.key,.value)) {
//Check for differences, and optionally exclude some keys
if (value '= target.%Get(key)&&("LastModifiedTime,LocalTime" '[key)) {
if (source.%GetTypeOf(key)="object" || (source.%GetTypeOf(key)="array") {
s tSC= ..CompareJSON(value,target.%Get(key),RefNo)
} else {
w !,"Source ",key,?30,value
w !,"Target ",key,?30,target.%Get(key)
//Store a reference if running in Batch mode, for later review
s ^zKeyTrap(key)=RefNo
set tSC = 0 // A mismatch means key-value is not identical
}
// Multiply so that any zeros bubble up
set Identical=(Identical*tSC)
}
}
q Identical
}
    
```

[#Caché](#)

URL de origem: <https://pt.community.intersystems.com/post/compara%C3%A7%C3%A3o-json-b%C3%A1sica-0>