

Méthodologie Gestion Bases de Données

Les Règles Principales:

En somme => Être très stricte sur ce qu'on émet et très tolérant sur ce qu'on accepte

- Doit ignorer : évènements et champs se que l'on ne connaît pas;
- Ne jamais Supprimer : Champs existants ou événements;
- Toujours Ajouter : Nouveau Champ si changement;

Exemple : (Fichier JSON)

V1

```
{  
  "schema": "application/vnd.cvv-holdings-v1+json",  
  "pageinfo": {  
    "after": "CU34",  
    "count": 9  
  },  
  "holdings": [  
    {  
      "weigth": 0.6111143334,  
      "qty": 4564444.0,  
      "value": 27.39,  
      "currency": "EUR",  
      "isin": "FR0010148333",  
    }  
  ]  
}
```

V2

```
{  
  "schema": "application/vdn.csv-holdings-holdings-v2+json",  
  "pageInfos": {  
    "after": "CU34",  
    "count": 9  
  },  
  "holdings": [  
    {  
      "weight": 0.6111111334,  
      "qty": 455566778.0,  
      "quantity": 455566778.0,  
      "value": 27.39,  
      "currency": "EURO",  
      "isin": "FR001014833",  
      "quotation": {  
        "value": 27.39,  
        "currency": "EURO"  
      },  
      "instrument": {  
        "isin": "FR0010148333"  
      }  
    }  
  ]  
}
```

Ce qui faut retenir :

!! Bien Numéroter ses versions !!

App A

App B

1er Cas :

2ième Cas :

```
V1      => Envoie Message.V2      => V2
        <=      Réponse.V1      <=
        <=      Réponse.V2      <=
```

!! A mémoriser !!

idempotence => Opération d'appel d'une fonction une fois et plusieurs fois

=> La fonction doit avoir le même comportement que si on l'appelle une seul fois.

Exemple du Token :

Limite qu'un formulaire soit soumis 2 fois.

- Evocation de la notion de contrat => tout les contrats sont sacrés.
- SLA (timeOut)
- BUGS : des contrats de fait
- Optional/Mandatory
- STATEFUL Temporal Coupling : e.g. long conversation longue et construction d'état par appels successifs => manipulation longue.
- Définition du fonctionnement de l'app (API) , de ses obligations contractuelles (contrats)

Evocation :

- AVRO
- PROTOBUF
- GRAPHQL
- REST

Connaitre le noyau dur des resources , savoir qui fait foi.
