

CRUD_FM_WebApi

Manuel de l'utilisateur des requêtes de l'application CRUD_FM_WebAPI

Version du manuel 1.0.0

Présenté par :

CRUD_FM_WebApi

Edward Renaud
Clément Perron

Version 1.0.0 (000)

Table des matières

1.	Introduction	3
1.	Les requêtes de type POST pour CRUD_WebAPI	4
1.1.	L'entête d'une requête POST pour CRUD_WebAPI	4
1.2.	Le corps d'une requête POST pour CRUD_WebAPI	4
1.3.	Interprétation du résultat JSON d'une requête	6
1.4.	FindAnyRecord : rechercher un enregistrement quelconque	9
1.5.	FindAllRecords rechercher tous les enregistrements	10
1.6.	FindRecords : rechercher les enregistrements répondant à un critère	10
1.7.	GetRecord : rechercher un enregistrement en particulier	11
1.8.	UpdateRecord : mettre à jour un enregistrement en particulier	12
1.9.	InsertRecord : insérer un nouvel enregistrement.....	13
1.10.	DeleteRecord : supprimer un enregistrement.....	14
1.11.	Ajout d'un tri à une action de recherche	15
1.12.	Invocation de scripts Filemaker	16
1.13.	GetValueLists : Obtenir une liste de valeurs Filemaker	18
1.14.	Conseils pour la création des requêtes.....	20
2.	Codes d'erreur Filemaker	21
	Addendum	28
1.	GetLayoutNames : Obtenir une liste des modèles dans la BD.....	28
2.	GetLayoutView : Obtenir des informations sur un modèle dans la BD	28
3.	GetScriptNames : Obtenir une liste des scripts dans la BD.....	30
4.	GetContainer : Obtenir l'objet dans un champ dans la BD	31
5.	SetContainer : Insérer un objet dans un champ dans la BD	31

1. Introduction

L'application CRUD_FM_WebAPI vise à offrir accès aux données contenues dans une base de données Filemaker via des requêtes de type « restfull » (POST) communément employées par de nombreux développeurs utilisant des APIs à des fins similaires.

Les requêtes doivent provenir de comptes usagers WebAPI définis à cette fin avec des informations d'authentification dans l'entête de la requête.

L'application implante les informations d'authentification Statique et Dynamique utilisées jusqu'à maintenant dans les entêtes de requêtes de type POST.

Notes importantes :

1- La version actuelle de la composante d'authentification de CRUD_FM_WebAPI fonctionne strictement à partir d'une table dans la base de données contenant les noms de comptes et leurs mots de passe respectifs et non pas les comptes natifs Filemaker.

Plusieurs tables peuvent servir à l'authentification pour divers besoins dans une même base de données Filemaker.

Ce document présente le fonctionnement des requêtes de l'application.

1. Les requêtes de type POST pour CRUD_WebAPI

Toute requête POST envoyée à CRUD_FM_WebAPI doit adhérer aux exigences qui seront décrites dans cette section.

1.1. L'entête d'une requête POST pour CRUD_WebAPI

Les entêtes de requête POST ont été discutées précédemment, il s'agit d'utiliser celui qui convient aux caractéristiques du compte utilisé, statique ou dynamique et selon les options spécifiées pour le compte.

1.2. Le corps d'une requête POST pour CRUD_WebAPI

Le corps d'une requête de type POST transmet les paramètres de recherche sous forme d'un texte JSON. Voici les éléments typiques qui doivent ou peuvent être présents dans les requêtes, dont certaines sont obligatoires pour toutes les requêtes, d'autres obligatoires selon le type de requête et enfin d'autres qui sont optionnelles dans bien des cas. D'abord une définition des paramètres, puis des exemples suivront quant à leur utilisation :

- **Host** : adresse IP du serveur Filemaker qui est visé par la requête.
- **Protocole** : normalement « http » car dans le même réseau que le serveur Filemaker.
- **Database** : le nom de la base de données visée.
- **Account** et **Password** : le nom et mot de passe d'un compte Filemaker permettant d'accéder à la base de données visée. Les requêtes associées à ce compte sont sujets aux *permissions définies dans Filemaker*.
- **Layout** : le nom du modèle Filemaker à utiliser pour la requête. Le modèle contient des champs d'une ou plusieurs tables et ce sont ces champs qui peuvent servir pour la recherche. Ce sont les valeurs dans ces champs qui seront retournées dans la réponse à la requête.

NOTES IMPORTANTES :

- Il est important de mettre seulement les champs dont on a vraiment besoin dans chacun des modèles visés afin de réduire le volume de texte transmis. Utiliser une grande variété de modèles différents n'a pas d'impact sur Filemaker mais permet d'optimiser les transferts.
- Il est avantageux aussi de ne pas inclure sur un modèle visé des champs d'une autre table liée par relation. Il est de loin préférable de faire plusieurs requêtes successives allant chercher dans des modèles respectifs les données recherchées.

- **Action** : l'opération que doit exécuter la requête dans la base de données Filemaker. Les actions possibles feront l'objet de discussion avec exemple dans une section subséquente :

- | | |
|------------------|---|
| ○ FindAnyRecord | rechercher un enregistrement quelconque |
| ○ FindAllRecords | rechercher tous les enregistrements |
| ○ FindRecords | rechercher les enregistrements répondant à un critère |
| ○ GetRecord | rechercher un enregistrement en particulier |
| ○ UpdateRecord | mettre à jour un enregistrement en particulier |
| ○ InsertRecord | insérer un nouvel enregistrement |
| ○ DeleteRecord | supprimer un enregistrement |
| ○ GetValueLists | Obtenir une liste de valeurs Filemaker |

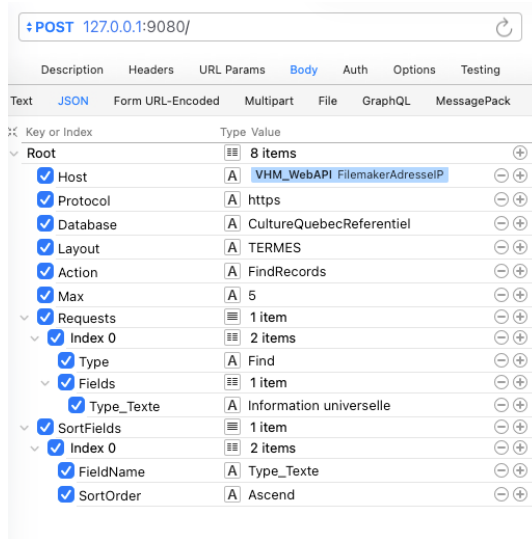
- **Max** : le nombre maximum d'enregistrements à retourner par Filemaker en réponse à la requête.

Encore ici, il est préférable de limiter le nombre d'enregistrements afin de conserver une charge raisonnable à Filemaker et CRUD_WebAPI. Nous avons observé chez des clients des attentes de plusieurs minutes lorsqu'il n'y a pas de maximum spécifié et que la requête retourne des milliers d'enregistrements d'un modèle avec un nombre excessif de champs dont plusieurs inutiles.

- **Skip** : le nombre d'enregistrements à sauter pour le retour des enregistrements en réponse à la requête.

NOTE IMPORTANTE : il est possible de procéder par pagination efficace en utilisant judicieusement les paramètres « Max » et « Skip ».

- **SortFields** : une liste JSON spécifiant les champs et l'ordre de tri pour les enregistrements retournés en réponse à une requête. Le tri peut être opéré en utilisant plusieurs champs du modèle.
- **Requests** : une liste JSON spécifiant les critères de recherche de la requête.
- **Fields** : un élément JSON spécifiant les champs visés par certaines actions autres que des recherches.
- **RecordID** : l'identifiant unique d'un enregistrement Filemaker. Cette valeur non modifiable est assignée par Filemaker lors de la création d'un enregistrement et permet de viser l'enregistrement spécifiquement.



1.3. Interprétation du résultat JSON d'une requête

La réponse à une requête est un item JSON comme suit :

```
{
  "fmresultset": {
    "error": {
      "code": "0"
    },
    "resultset": {
      "count": "1",
      "fetch-size": "1",
      "record": [
        {
          "fields": {
            "City": "Palm Desert",
            "Comments": "",
            "Latitude": "33.755527496337891",
            "Longitude": "-116.36180114746094",
            "Phone Number": "760-341-2211",
            "Photo": "",
            "Postal Code": "92260",
            "State": "CA",
            "Store ID": "16642",
            "Store Name": "Marriott Desert Springs",
            "Street 1": "74855 Country Club Dr",
            "Street 2": "",
            "Street 3": ""
          },
          "mod-id": "1",
          "record-id": "7879"
        }
      ]
    }
  }
}
```

La réponse est l'élément « **fmresultset** » qui contient à son tour 2 éléments :

1. **error** : un élément contenant un code d'erreur numérique dans son élément « **code** ». Ce code d'erreur est une erreur rapportée par Filemaker lors de son traitement de la requête. Tous les codes d'erreur Filemaker sont présentés dans la dernière section du présent document. Les erreurs les plus fréquentes sont colorées en rouge dans le tableau complet et présentées ici :

0	Pas d'erreur
101	Enregistrement manquant
102	Rubrique manquante
105	Modèle manquant
802	Impossible d'ouvrir le fichier

2. **resultset** : un élément contenant les données récupérés, dont :

```
"resultset": {
  "count": "1",
  "fetch-size": "1",
  "record": [
    {
      "fields": {
        "City": "Palm Desert",
        "Comments": "",
        "Latitude": "33.755527496337891",
        "Longitude": "-116.36180114746094",
        "Phone Number": "760-341-2211",
        "Photo": "",
        "Postal Code": "92260",
        "State": "CA",
        "Store ID": "16642",
        "Store Name": "Marriott Desert Springs",
        "Street 1": "74855 Country Club Dr",
        "Street 2": "",
        "Street 3": ""
      },
      "mod-id": "1",
      "record-id": "7879"
    }
  ]
}
```

- 2.1. **count** : le nombre d'enregistrements trouvés par la requête
- 2.2. **fetch-size** : le nombre d'enregistrements joints dans la présente réponse. Cette valeur est affectée par le paramètre « Max » qui peut avoir été joint à la requête
- 2.3. **record** : une liste d'éléments où chacun représente un enregistrement. Les informations relatives à chaque enregistrement :
 - 2.3.1. **mod-id** : un numéro pas utile pour nos requêtes et qu'on ignore tout simplement.
 - 2.3.2. **record-id** : le numéro unique identifiant l'enregistrement. Ce numéro doit être utilisé lors de la suppression ou la mise à jour de l'enregistrement
 - 2.3.3. **fields** : un élément JSON contenant les données de l'enregistrement. Les champs qu'on y retrouve sont ceux du modèle identifié dans la requête, dans cet exemple :

```
"City": "Palm Desert",
"Comments": "",
"Latitude": "33.755527496337891",
"Longitude": "-116.36180114746094",
"Phone Number": "760-341-2211",
"Photo": "",
"Postal Code": "92260",
"State": "CA",
"Store ID": "16642",
"Store Name": "Marriott Desert Springs",
"Street 1": "74855 Country Club Dr",
"Street 2": "",
"Street 3": ""
```

POST http://127.0.0.1:9080/		200 OK
Info Request Response		
Headers JSON Raw		
Key or Index	Type	Value
Root	1 item	
fmresultset	2 items	
resultset	3 items	
count	8	
fetch-size	5	
record	5 items	
Index 0	3 items	
mod-id	23	
record-id	2	
field	12 items	
Index 0	2 items	
name	pk_ID	
data	23	
Index 1	2 items	
name	CreationTimestamp	
data	09/02/2024 15:43:29	
Index 2	2 items	
name	CreatedBy	
data	admin	
Index 3	2 items	
name	ModificationTimestamp	
data	01/24/2026 13:58:04	
Index 4	2 items	
name	ModifiedBy	
data	AQM_DepotCorpus	
Index 5	2 items	
name	Vocabulaire	
data	Accessible aux personnes en fauteuil roulant	
Index 6	2 items	
name	Version_Aff	
data	Accessible aux personnes en fauteuil roulant TEST	
Index 7	2 items	
name	Code	
data		

1.4. FindAnyRecord : rechercher un enregistrement quelconque

Cette action n'est pas d'utilité évidente si ce n'est que pour lancer un script Filemaker dont le résultat ne nous intéresse guère ou recevoir la liste des champs exposés dans le modèle. Nous reviendrons plus loin sur le déclenchement de scripts.

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store Details",
  "Account": "xfm",
  "Password": "latte",
  "Action": "FindAnyRecord"
}
```

Le résultat de cette requête sera une représentation JSON d'un enregistrement quelconque de la table visée :

```
{
  "fmresultset": {
    "error": {
      "code": "0"
    },
    "resultset": {
      "count": "1",
      "fetch-size": "1",
      "record": [
        {
          "fields": {
            "City": "Lacey",
            "Comments": "",
            "Latitude": "47.037693023681641",
            "Longitude": "-122.82286071777344",
            "Phone Number": "360-456-1891",
            "Photo": "",
            "Postal Code": "98503",
            "State": "WA",
            "Store ID": "91390",
            "Store Name": "College & Pacific - Lacey",
            "Street 1": "1110 College Street SE",
            "Street 2": "",
            "Street 3": ""
          },
          "mod-id": "1",
          "record-id": "10162"
        }
      ]
    }
  }
}
```

1.5. FindAllRecords rechercher tous les enregistrements

L'action permet d'obtenir *tous les enregistrements* de la table. Dans ces cas, à moins qu'il ne soit certain que la table ne contient qu'un petit nombre d'enregistrements, vaut mieux utiliser les paramètres « **Max** » et « **Skip** » pour effectuer une pagination raisonnable.

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "FindAllRecords",
  "Max": "50",
  "Skip": "150"
}
```

Le résultat prendra la même forme JSON que l'exemple précédent « FindAnyRecord ».

1.6. FindRecords : rechercher les enregistrements répondant à un critère

Probablement l'action qui sera la plus utile et la plus utilisée, toujours en considérant bien l'utilisation de « **Max** » et « **Skip** » pour effectuer une pagination raisonnable.

Cette requête de recherche ajoute la liste « **Requests** » qui permet de spécifier les valeurs recherchées dans des champs spécifiques et le « Type » de recherche « **find** » :

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "FindRecords",
  "Max": "50",
  "Skip": "150",
  "Requests": [
    {
      "Fields": {
        "State": "VA",
        "City": "Richmond"
      },
      "Type": "find"
    }
  ]
}
```

Il est possible aussi d'introduire l'élément « Type » avec la valeur « **omit** » afin de spécifier des critères d'exclusion. La requête précédente pourrait alors plutôt prendre la forme :

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "FindRecords",
  "Max": "50",
  "Skip": "150",
  "Requests": [
    {
      "Fields": {
        "State": "VA"
      },
      "Type": "find"
    },
    {
      "Fields": {
        "City": "Richmond"
      },
      "Type": "omit"
    }
  ]
}
```

Le résultat prendra la même forme JSON que l'exemple précédent « FindAnyRecord ».

1.7. GetRecord : rechercher un enregistrement en particulier

Cette recherche permet de retrouver un enregistrement particulier selon la valeur de « **RecordID** » de celui-ci. Ceci implique probablement une recherche antérieure ayant retourné l'enregistrement contenant son identifiant et qu'on l'ait conservé quelque part afin de retrouver l'enregistrement à partir de celui-ci. La recherche prend donc cette forme :

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "GetRecord",
  "RecordID": 7879
}
```

Le résultat prendra la même forme JSON que l'exemple précédent « FindAnyRecord ».

1.8. UpdateRecord : mettre à jour un enregistrement en particulier

La mise à jour d'un enregistrement est l'autre type de requête qui doit référencier l'enregistrement avec le paramètre « **RecordID** ». Il n'est pas possible de mettre à jour plusieurs enregistrements à la fois, sinon via un script.

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "UpdateRecord",
  "RecordID": "13227",
  "Fields": {
    "Store Name": "Test Updated",
    "Store ID": "Updated"
  }
}
```

La réponse de Filemaker est une représentation JSON de l'enregistrement modifié :

```
{
  "fmresultset": {
    "error": {
      "code": "0"
    },
    "resultset": {
      "count": "1",
      "fetch-size": "1",
      "record": [
        {
          "fields": {
            "City": "Columbus",
            "Latitude": "",
            "Longitude": "",
            "Phone Number": "",
            "Postal Code": "",
            "State": "OH",
            "Store ID": "Updated",
            "Store Name": "Test Updated",
            "Street 1": "100 Test Avenue",
            "Street 2": "",
            "Street 3": ""
          },
          "mod-id": "1",
          "record-id": "13227"
        }
      ]
    }
  }
}
```

1.9. InsertRecord : insérer un nouvel enregistrement

L'insertion d'un nouvel enregistrement prend la forme suivante où on inclut simplement une liste des champs et les valeurs à inscrire dans le nouvel enregistrement:

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "InsertRecord",
  "Fields": {
    "City": "Columbus",
    "State": "OH",
    "Store Name": "Test",
    "Street 1": "100 Test Avenue"
  }
}
```

La création d'un nouvel enregistrement peut présenter toutefois quelques défis si les champs visés sont définis dans Filemaker avec des valeurs par défaut. Aussi, il ne faut pas tenter d'insérer une valeur définie comme un résultat de calcul. Il n'est pas possible non plus d'insérer des valeurs dans les champs répétés à moins d'avoir recours à un script.

Le résultat de l'insertion nous retournera les valeurs dans tous les champs du modèle ainsi que le « **record-id** » qui pourra servir à une mise à jour ultérieure si nécessaire :

```
{
  "fmresultset": {
    "error": {
      "code": "0"
    },
    "resultset": {
      "count": "1",
      "fetch-size": "1",
      "record": [
        {
          "fields": {
            "City": "Columbus",
            "Comments": "",
            "Latitude": "",
            "Longitude": "",
            "Phone Number": "",
            "Photo": "",
            "Postal Code": "",
            "State": "OH",
            "Store ID": "",
            "Store Name": "Test",
            "Street 1": "100 Test Avenue",
            "Street 2": "",
            "Street 3": ""
          },
          "mod-id": "0",
          "record-id": "13227"
        }
      ]
    }
  }
}
```

```
    ]
  }
}
}
```

1.10. DeleteRecord : supprimer un enregistrement

Pour la suppression d'un enregistrement le numéro de l'enregistrement doit être fourni comme paramètre « **RecordID** ». Il n'est pas possible de mettre à jour plusieurs enregistrements à la fois, sinon via un script.

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store Details",
  "Account": "xfm",
  "Password": "latte",
  "Action": "DeleteRecord",
  "RecordID": 13227
}
```

Pour l'action de suppression, Filemaker se contente de retourner le code d'erreur, qui devrait être 0 si l'enregistrement a pu être supprimé avec succès.

```
{
  "fmresultset": {
    "error": {
      "code": "0"
    },
    "resultset": {
      "count": "0",
      "fetch-size": "0"
    }
  }
}
```

1.11. Ajout d'un tri à une action de recherche

Le paramètre « **SortFields** » permet de demander un tri basé sur un ou plusieurs champs, chacun **ascendant** ou **descendant**. Ce paramètre peut contenir plusieurs critères.

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "FindRecords",
  "Max": "50",
  "Skip": "150",
  "Requests": [
    {
      "Fields": {
        "State": "VA",
        "City": "Richmond"
      },
      "Type": "find"
    }
  ],
  "SortFields": [
    {
      "FieldName": "State",
      "SortOrder": "Ascend"
    },
    {
      "FieldName": "City",
      "SortOrder": "Descend"
    }
  ]
}
```

1.12. Invocation de scripts Filemaker

Pour déclencher l'exécution d'un script dans la BD Filemaker dans une requête, on ajoute des paramètres au corps JSON. L'utilisation de scripts peut servir à plusieurs types de besoins dont, à titre d'exemple :

- Aller chercher des données dans d'autres systèmes auxquelles vous n'avez pas accès mais auxquels le serveur Filemaker a accès, pour des raisons de sécurité (ex : pare-feu).
- Modifications à plusieurs tables Filemaker sous forme de transaction qui doit être annulée si l'exécution n'est pas complétée avec succès.
- Transmettre un gros document comme paramètre pour sauvegarde sur un disque auquel notre application n'a pas accès.

Il y a 3 moments possibles pour l'exécution du script :

1. Après l'exécution de l'action de recherche et du tri, s'il y en a un :

- On peut passer le nom du script à exécuter comme paramètre « **Script** » avec les actions

```
FindAnyRecord
FindAllRecords
FindRecords
GetRecord
UpdateRecord
InsertRecord
DeleteRecord
```

- On peut optionnellement inclure un « **ScriptParam** » pour passer un paramètre pour le script.

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "FindRecords",
  "Script": "monScript",
  "ScriptParam": "monParametreDeScript",    (optionnel, avec le Script)
  "Fields": {
    "City": "Columbus",
    "State": "OH",
    "Store Name": "Test",
    "Street 1": "100 Test Avenue"
  }
}
```

2. Avant l'exécution de l'action de recherche et de tri, s'il y en a un :

- On peut passer le nom du script à exécuter comme paramètre « **ScriptPreFind** » avec les actions :

```
FindAnyRecord
FindAllRecords
FindRecords
```


- On peut optionnellement inclure un « **ScriptPreFindParam** » pour passer un paramètre pour le script.

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "FindRecords",
  "ScriptPreFind": "monScript",
  "ScriptPreFindParam": "monParametreDeScript",      (optionnel, avec le ScriptPreFind)
  "Fields": {
    "City": "Columbus",
    "State": "OH",
    "Store Name": "Test",
    "Street 1": "100 Test Avenue"
  }
}
```

3. Après l'exécution de l'action de recherche, s'il y en a une, mais avant le tri :

- On peut utiliser le nom du script à exécuter comme paramètre « **ScriptPreSort** » avec les actions :

```
FindAnyRecord
FindAllRecords
FindRecords
```

- On peut optionnellement inclure un « **ScriptPreSortParam** » pour passer un paramètre pour le script.

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "FindRecords",
  "ScriptPreSort": "monScript",
  "ScriptPreSortParam": "monParametreDeScript",      (optionnel, avec le ScriptPreSort)
  "Fields": {
    "City": "Columbus",
    "State": "OH",
    "Store Name": "Test",
    "Street 1": "100 Test Avenue"
  }
}
```

1.13. GetValueLists : Obtenir une liste de valeurs Filemaker

La requête pour obtenir une liste de valeurs passe par la présence d'un ou plusieurs champs dans un modèle qui sont liés à une liste de valeurs et prend cette forme :

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "nomDeLaBdCible",
  "Layout": "NomDuModeleCible",
  "Account": "compteFilemaker",
  "Password": "motDePasseDuCompteFilemaker",
  "Action": "GetValueLists"
}
```

Cette requête peut être utile, voire nécessaire, pour obtenir une liste de valeurs « acceptables » pour d'autres actions.

L'allure du résultat de cette requête prend 2 formes selon qu'il existe un seul ou plusieurs champs avec liste de valeurs dans le modèle ciblé par la requête :

- Un seul champ => 1 seule liste de valeurs :

```
{
  "VALUELIST": {
    "NAME": "Xojo_datesFacturation",
    "VALUE": [
      {
        "DISPLAY": "07/31/2022",
        "text": "07/31/2022"
      },
      {
        "DISPLAY": "08/31/2022",
        "text": "08/31/2022"
      },
      {
        "DISPLAY": "09/30/2022",
        "text": "09/30/2022"
      },
      {
        "DISPLAY": "10/31/2022",
        "text": "10/31/2022"
      },
      {
        "DISPLAY": "11/30/2022",
        "text": "11/30/2022"
      }
    ]
  }
}
```

- Plus d'un champ => plusieurs listes de valeurs; voici un exemple avec 2 :

```
{
  "VALUelist": [
    {
      "NAME": "Xojo_datesFacturation",
      "VALUE": [
        {
          "DISPLAY": "07/31/2022",
          "text": "07/31/2022"
        },
        {
          "DISPLAY": "08/31/2022",
          "text": "08/31/2022"
        },
        {
          "DISPLAY": "09/30/2022",
          "text": "09/30/2022"
        },
        {
          "DISPLAY": "10/31/2022",
          "text": "10/31/2022"
        },
        {
          "DISPLAY": "11/30/2022",
          "text": "11/30/2022"
        }
      ]
    },
    {
      "NAME": "Version",
      "VALUE": [
        {
          "DISPLAY": "Dev",
          "text": "Dev"
        },
        {
          "DISPLAY": "Prod",
          "text": "Prod"
        },
        {
          "DISPLAY": "Staging",
          "text": "Staging"
        }
      ]
    }
  ]
}
```

1.14. Conseils pour la création des requêtes

1. Une excellente connaissance de Filemaker est nécessaire pour une utilisation efficace de CRUD_WebAPI.
2. Il est très facile de créer des requêtes fonctionnelles mais qui contribuent à une charge trop importante au serveur FMP, donc il est important de formuler les requêtes en appliquant certains principes de base :
 - 2.1. Inclure sur les modèles utilisés que les champs dont l'information est utile pour l'application client afin de minimiser le volume de data échangé.
 - 2.2. Créer plus de modèles pour une même table ayant chacun le nombre minimum de champs pour différentes requêtes car le nombre de modèles d'une base de données n'affecte pas la performance du serveur.
 - 2.3. Planifier les requêtes en utilisant les paramètres « MAX » et « SKIP » pour limiter le nombre de résultats à retourner par le serveur et utiliser au besoin une approche de pagination.
 - 2.4. Sur les modèles utiliser au minimum les champs multivalués et les champs liés par une relation; utiliser plutôt plusieurs requêtes spécifiques et bien ciblées.
 - 2.5. S'assurer que les champs servant comme critères de recherche ou de tri soient indexés dans Filemaker (gros impact sur la performance!)
3. Travailler avec un œil attentif aux changements dans la BD Filemaker utilisée :
 - 3.1. Supprimer ou renommer des champs dans un modèle utilisé engendrera des erreurs dans des requêtes qui fonctionnaient bien auparavant.
 - 3.2. Retirer des champs dans un modèle utilisé peut provoquer des erreurs dans le traitement des résultats d'une requête qui fonctionnait bien.
 - 3.3. Ajouter des champs dans un modèle peut entraîner une augmentation imprévue dans la quantité de données véhiculée entre l'application client et le serveur Filemaker.
 - 3.4. Si la BD de l'application Filemaker est utilisée par des usagers via l'application Filemaker Pro client ou Web Publishing ou application mobile, il peut être prudent de dédier des modèles pour chaque besoin, permettant ainsi certaines modifications nécessaires sans s'affecter mutuellement.

2. Codes d'erreur Filemaker

Les codes d'erreur Filemaker pour les différentes versions peuvent être consultées en ligne.

Filemaker 16 : https://help.claris.com/archive/help/16/fmp/fr/index.html#page/FMP_Help/error-codes.html

Filemaker 17 : https://help.claris.com/archive/help/17/fmp/fr/index.html#page/FMP_Help/error-codes.html

Filemaker 18 : https://help.claris.com/archive/help/18/fmp/fr/index.html#page/FMP_Help/error-codes.html

Filemaker 19 : <https://help.claris.com/archive/fm19/fr/pro-help/content/error-codes.html?Highlight=erreurs>

Voici la liste des codes d'erreur que Filemaker 19 peut retourner dans la réponse JSON à une requête transmise via CRUD_WebAPI.

Code	Description
-1	Erreur inconnue
0	Pas d'erreur
1	L'utilisateur a annulé l'action
2	Erreur de mémoire
3	Commande non disponible (par exemple, système d'exploitation ou mode incorrect)
4	Commande inconnue
5	Commande incorrecte (par exemple, un calcul n'a pas été indiqué pour une action de script Définir rubrique)
6	Fichier accessible en lecture seule
7	Mémoire insuffisante
9	Privilèges d'accès insuffisants
10	Données requises manquantes
11	Nom incorrect
12	Ce nom existe déjà
13	Le fichier ou l'objet est déjà utilisé
14	En dehors de l'interv.
15	Division par zéro impossible
16	Echec de l'opération, réessayez (par ex. une requête utilisateur)
17	Echec de tentative de conversion d'un jeu de caract. étrangers en UTF-16
18	Le client doit fournir les données du compte pour poursuivre
19	La chaîne contient des caractères autres que A-Z, a-z, 0-9 (ASCII)
20	Commande/opération annulée par le script déclenché
21	Requête non prise en charge (par ex. lors de la création d'un lien en dur sur un syst. de fichiers qui ne prend pas en charge ce type de lien)
100	Fichier manquant
101	Enregistrement manquant
102	Rubrique manquante
103	Lien manquant
104	Script manquant
105	Modèle manquant
106	Table manquante
107	Index manquant
108	Liste de valeurs manquante

109	Jeu de privilèges d'accès manquant
110	Tables liées manquantes
111	Valeur de rubrique incorrecte
112	Fenêtre manquante
113	Fonction manquante
114	Référence de fichier manquante
115	Jeu de menus manquant
116	Objet de modèle manquant
117	Source de données manquante
118	Le thème est manquant
130	Les fichiers sont endommagés ou manquants et doivent être réinstallés
131	Les fichiers de pack de langues sont manquants
200	L'accès à l'enregistrement est interdit
201	Impossible de modifier la rubrique
202	L'accès à la rubrique est interdit
203	Aucun enregistrement à imprimer, ou bien un mdp est nécessaire pour imprimer
204	Aucun accès aux rubriques dans l'ordre de tri
205	Impossible de créer de nvx enregistrements ; l'importation peut remplacer les données existantes
206	Impossible de modifier le mot de passe ou bien le fichier ne peut pas être modifié
207	Impossible d'accéder au schéma de BDD ou bien le fichier ne peut pas être modifié
208	Nombre de caractères insuffisant dans le mot de passe
209	Le nv mdp doit être différent du mdp existant
210	Compte utilisateur inactif
211	Mot de passe expiré
212	Compte utilisateur ou mot de passe non valide
214	Tentatives de connexion trop nombreuses
215	Impossible de dupliquer les droits Admin
216	Impossible de dupliquer un compte Invité
217	L'utilisateur ne dispose pas de droits suffisants pour modifier le compte admin
218	Le mdp et le mdp de confirmation ne correspondent pas
219	Impossible d'ouvrir le fichier ; vous devez être un utilisateur sous licence ; contactez le chef d'équipe
300	Fichier verrouillé ou en cours d'utilisation
301	L'enregistrement est déjà utilisé par un autre utilis.
302	La table est déjà utilisée par un autre utilis.
303	Le schéma de base de données est déjà utilisé par un autre utilis.
304	Le modèle est déjà utilisé par un autre utilisateur
306	L'ID de modification d'enregistrement ne corresp. pas
307	La transact. ne peut pas être verrouillée en raison d'une erreur de communication avec l'hôte
308	Le thème est déjà utilisé et verrouillé par un autre utilis.
400	Les critères de recherche sont vides
401	Aucun enregistrement ne correspond à cette requête
402	Ceci n'est pas une rubrique clé pour une référence externe
404	Ordre de tri incorrect

405	Le nombre d'enregistrements indiqué dépasse le nombre d'enregistrements pouvant être ignorés
406	Les critères de remplacement et de renumérotation sont incorrects
407	Une ou les deux rubriques clés manquent (lien incorrect)
408	Le type de la rubrique indiquée est incorrect pour ce type d'opération
409	Ordre d'importation incorrect
410	Ordre d'exportation incorrect
412	Version de FileMaker Pro incorrecte pour récupérer le fichier
413	Le type de la rubrique indiquée est incorrect
414	Résultat impossible à afficher sur le modèle
415	Un ou plusieurs enregistrements liés requis ne sont pas disponibles
416	Une clé primaire est requise dans la table source de données
417	Le fichier n'est pas une source de données prise en charge
418	Echec interne lors de l'opération INSERER dans une rubrique
500	La date indiquée ne corresp. pas aux options d'entrée définies
501	L'heure indiquée ne corresp. pas aux options d'entrée définies
502	Le numéro indiqué ne corresp. pas aux options d'entrée définies
503	La valeur de la rubrique n'est pas comprise dans la fourchette de valeurs définies
504	La valeur de la rubrique ne correspond pas à la valeur définie dans les options d'entrée
505	La valeur de la rubrique ne correspond à aucune donnée existante contenue dans le fichier des options d'entrée
506	La valeur de la rubrique n'est pas incluse dans la liste des valeurs définies
507	La valeur de la rubrique n'est pas conforme aux critères de contrôle de calcul des options d'entrée
508	Valeur incorrecte saisie en mode Recherche
509	La rubrique requiert une valeur correcte
510	La valeur liée est vide ou non disponible
511	La valeur de la rubr. dépasse la taille de rubrique max.
512	Enregistrement déjà modifié par un autre utilis.
513	Pas de validation définie, mais les données ne peuvent pas être affichées entièrement dans la rubr.
600	Une erreur d'impr. s'est produite
601	L'en-tête et le pied de page combinés sont plus longs qu'une page
602	Le corps de la page ne tient pas sur une page pour la dispos. en col. active
603	Connexion d'imprimante perdue
700	Le fichier ne possède pas le bon format d'import.
706	Le fichier EPS est dépourvu d'images de prévisualis.
707	Le traducteur graphique est introuvable
708	Impossible d'importer le fichier, ou bien un moniteur gérant les couleurs est nécessaire pour effect. cette opér.
711	Le traducteur d'importation est introuvable
714	Les privilèges d'accès associés à votre mot de passe ne vous permettent pas d'effectuer cette opér.
715	Feuille de calcul ou plage nommée Excel spécifiée manquante
716	Une requête SQL utilisant les instructions DELETE, INSERT ou UPDATE n'est pas autorisée dans l'import. ODBC
717	Les informations XML/XSL sont insuffisantes pour procéder à l'import. ou à l'export.
718	Erreur lors de l'analyse du fichier XML (à partir de Xerces)

719	Erreur lors de la transform. du fichier XML en fichier XSL (à partir de Xalan)
720	Erreur lors de l'export. le format souhaité ne prend pas en charge les rubr. multivaluées
721	Une erreur inconnue s'est produite dans l'analyseur ou le transformateur
722	Imposs. d'importer des données dans un fichier dépourvu de rubr.
723	Vous ne disposez pas des droits nécess. pour ajouter des enregist. ou les modifier dans la table cible
724	Vous ne disposez pas des droits nécess. pour ajouter des enregist. à la table cible
725	Vous ne disposez pas des droits nécess. pour modifier les enregist. dans la table cible
726	Le fichier src a plus d'enregist. que la table cible ; tous les enregist. n'ont pas été importés
727	La table cible a plus d'enregist. que le fichier source ; tous les enregist. n'ont pas été mis à jour
729	Des erreurs se sont produites pendant l'importation. Des enregist. n'ont pas pu être importés
730	Version d'Excel non prise en charge ; convertissez le fichier au format Excel actuel et réessayez
731	Le fichier servant à l'import. ne contient aucune donnée
732	Impossible d'insérer ce fichier car il contient d'autres fichiers
733	Une table ne peut être importée en elle-même
734	Ce type de fichier n'a pu être affiché sous la forme d'une image
735	Ce fichier ne peut pas être affiché sous forme d'image. Il sera inséré, puis affiché sous forme de fichier
736	Les données à exporter dans ce format sont trop nombreuses et seront tronquées
738	Le thème que vous tentez d'importer existe déjà
800	Impossible de créer le fichier sur le disque
801	Impossible de créer un fichier temporaire sur le disque
802	Impossible d'ouvrir le fichier
803	Le fichier est mono-utilisateur ou bien l'hôte est introuvable
804	Le fichier ne peut pas être ouvert en lecture seule dans son état actuel
805	Le fichier est endommagé
806	Le fichier ne peut être ouvert avec cette version d'un client FileMaker
807	Le fichier n'est pas un fichier FileMaker Pro ou est sérieusement endommagé
808	Impossible d'ouvrir le fichier car les privilèges d'accès sont endommagés
809	Le disque/volume est saturé
810	Le disque/volume est verrouillé
811	Un fichier temporaire ne peut être ouvert en tant que fichier FileMaker Pro
812	La capacité de l'hôte a été dépassée
813	Erreur de synchr. d'enregistrements sur le réseau
814	Impossible d'ouvrir le fichier, car le nombre maximal de fichiers ouverts est atteint
815	Impossible d'ouvrir le fichier de référence externe
816	Impossible de convertir le fichier
817	Impossible d'ouvrir le fichier, car il n'appartient pas à cette solution
819	Impossible d'enregistrer une copie locale d'un fichier distant
820	Fichier en cours de fermeture
821	Déconnexion forcée par l'hôte
822	Fichiers FileMaker Pro introuvables ; réinstallez les fichiers manquants
823	Impossible de définir le fichier en mode mono-utilisateur, des invités sont connectés
824	Le fichier est endommagé ou n'est pas un fichier FileMaker Pro
825	Le fichier n'est pas autorisé à référencer le fichier protégé

826	Le chemin d'accès indiqué pour le fichier est non valide
827	Le fichier n'a pas été créé parce que la source est vide de données ou est une référence
850	Le chemin est non valide pour le système d'exploitation
851	Impossible de supprimer un fichier externe du disque
852	Impossible d'écrire dans un fichier se trouvant dans le stockage externe
853	Un ou plusieurs conteneurs n'ont pas pu être transférés
870	Impossible de modifier le fichier parce qu'un autre utilisateur effectue actuellement des modifications
871	Erreur apparue pendant le chargement du modèle Core ML
872	Le modèle Core ML n'a pas été chargé, car il contenait un paramètre d'entrée ou de sortie non pris en charge
900	Erreur générale de vérif. orthographique
901	Le dictionnaire principal n'est pas installé
903	Cette commande ne peut pas être utilisée dans un fichier partagé
905	Cette commande nécessite une rubrique pour être active
906	Le fichier en cours n'est pas partagé ; la commande ne peut être utilisée que si le fichier est partagé
920	Impossible d'init. le correcteur orthographique
921	Impossible de charger le dictionnaire de l'utilis. pour modification
922	Dictionnaire de l'utilis. introuvable
923	Dictionnaire de l'utilis. en lecture seule
951	Une erreur inattendue s'est produite (*)
952	Jeton FileMaker Data API non valide (*)
953	Limite de données transmises par FileMaker Data API et OData dépassée (*)
954	Grammaire XML non prise en charge (*)
955	Aucun nom de base de données (*)
956	Le nombre maximal de sessions de BDD ou Admin API a été dépassé (*)
957	Commande en conflit (*)
958	Paramètre manquant (*)
959	La technologie de Publication Web personnalisée est désactivée
960	Paramètre non valide
1200	Erreur de calcul générique
1201	Paramètres trop peu nombreux dans la fonction
1202	Paramètres trop nombreux dans la fonction
1203	Fin de calcul non conforme
1204	Nombre, chaîne, nom de rubrique ou parenthèse ouvrante
1205	Commentaire non terminé par « */ »
1206	La chaîne doit se terminer par un guillemet
1207	Parenthèses non équilibrées
1208	Opérateur manquant, fonction introuvable ou parenthèse ouvrante non nécessaire
1209	Nom (nom de rubrique ou de modèle) manquant
1210	Fonction plug-in ou action de script déjà enregistrée
1211	L'utilis. de listes n'est pas autorisée dans cette fonction
1212	Entrez ici un opérateur (+, -, *,).
1213	Cette variable a déjà été définie dans la fonction Définir

1214	Un paramètre de fonction contient une expression dans laquelle une rubrique est requise.
1215	Ce paramètre est incorrect pour la fonction Obtenir
1216	Seules les rubriques de type Statistique sont autorisées comme premier argument dans Recapitulatif
1217	Rubrique de regroupement incorrecte
1218	Evaluation du nombre impossible
1219	Impossible d'utiliser une rubrique dans sa propre formule
1220	Le type de la rubrique doit être de type normal ou calculé
1221	Le type de données doit être un nombre, une date, une heure ou un horodatage
1222	Impossible d'enregistrer le calcul
1223	La fonction à laquelle il fait référence n'est pas encore mise en œuvre
1224	La fonction à laquelle il fait référence n'existe pas
1225	La fonction à laquelle il fait référence n'est pas prise en charge dans ce contexte
1300	Le nom indiqué ne peut pas être utilisé
1301	Un paramètre de la fonction importée ou collée porte le même nom que la fonction du fichier
1400	L'initialisation du pilote ODBC a échoué ; assurez-vous que les pilotes ODBC sont correctement installés.
1401	Echec de l'allocation d'environnement (ODBC)
1402	Echec de la libération d'environnement (ODBC)
1403	Echec de la déconnexion (ODBC)
1404	Echec de l'allocation de connexion (ODBC)
1405	Echec de la libération de connexion (ODBC)
1406	Echec de la vérification de SQL API (ODBC)
1407	Echec de l'allocation d'instruction (ODBC)
1408	Erreur étendue (ODBC)
1409	Erreur (ODBC)
1413	Echec du lien de communication (ODBC)
1414	L'instruction SQL est trop longue
1415	Déconnexion en cours (ODBC)
1450	Action nécessitant une extension du privilège PHP (*)
1451	Action nécessitant une gestion à distance du fichier actif
1501	Echec de l'authentification SMTP
1502	Connexion refusée par le serveur SMTP
1503	Erreur avec SSL
1504	Le serveur SMTP requiert le cryptage de la connexion
1505	L'authentification spécifiée n'est pas prise en charge par le serveur SMTP
1506	L'envoi du ou des emails a échoué
1507	Impossible de se connecter au serveur SMTP
1550	Impossible de charger le plug-in, ou plug-in non valide
1551	Impossible d'installer le plug-in ; impossible de suppr. un plug-in existant ou d'écrire sur le dossier ou le disque.
1552	-
1559	Renvoyé par les plug-ins ; consultez la documentation fournie avec ce plug-in
1626	Le protocole n'est pas pris en charge
1627	Echec de l'authentification

1628	Une erreur s'est produite avec SSL
1629	Connexion expirée ; la valeur du délai d'expir. est de 60 secondes
1630	Format d'URL incorrect
1631	Echec de la connexion
1632	Le certificat a expiré
1633	Le certificat est auto-signé
1634	Une erreur de vérification du certificat s'est produite
1635	Connexion non cryptée
1700	La ressource n'existe pas (*)
1701	L'hôte est actuellement incapable de recevoir des demandes (*)
1702	Les informations d'authentification n'ont pas été fournies au bon format ; vérifiez la valeur de l'en-tête d'autorisation (*)
1703	Nom d'utilisateur, mot de passe ou jeton JWT (JSON Web Token) non valide (*)
1704	La ressource ne prend pas en charge le verbe HTTP spécifié (*)
1705	L'en-tête HTTP requis n'a pas été spécifié (*)
1706	Ce paramètre n'est pas pris en charge (*)
1707	Le paramètre requis n'a pas été spécifié dans la demande (*)
1708	Valeur de paramètre non valide (*)
1709	Opération non valide pour l'état actuel de la ressource (*)
1710	L'entrée JSON n'est pas syntaxiquement correcte (*)
1711	La licence de l'hôte a expiré (*)
1712	Le fichier de clé privée existe déjà ; supprimez-le et réexécutez la commande (*)
1713	La requête API n'est pas prise en charge par ce système d'exploitation (*)
1714	Nom du groupe externe non valide (*)
1715	La connexion avec un compte de serveur externe n'est pas activée (*)

Addendum

Cette section vise à présenter d'autres types de requêtes, plus avancées mais présentant certains défis additionnels.

1. GetLayoutNames : Obtenir une liste des modèles dans la BD

La requête pour obtenir une liste des modèles peut être utile pendant le développement lorsqu'on n'a pas un accès direct à la base de données avec un client Filemaker :

```
{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Account": "xfm",
  "Password": "latte",
  "Action": "GetLayoutNames"
}
```

Le résultat de cette requête a cette allure :

```
{
  "fmresultset": {
    "error": {
      "code": "0"
    },
    "resultset": {
      "count": "2",
      "fetch-size": "2",
      "record": [
        {
          "fields": {
            "LAYOUT_NAME": "Store List"
          },
          "mod-id": "11",
          "record-id": "1"
        },
        {
          "fields": {
            "LAYOUT_NAME": "Store Details"
          },
          "mod-id": "16",
          "record-id": "3"
        }
      ]
    }
  }
}
```

2. GetLayoutView : Obtenir des informations sur un modèle dans la BD

La requête pour obtenir la liste des champs d'un modèle ainsi que des informations techniques sur ces champs peut aussi être utile pendant le développement lorsqu'on n'a pas un accès direct à la base de données avec un client Filemaker :

```
{
```

```
"Host": "localhost",
"Protocol": "http",
"Database": "starbucks-us",
"Layout": "Store List",
"Account": "xfm",
"Password": "latte",
"Action": "GetLayoutView"
}
```

Le résultat de cette requête a cette allure :

```
{
  "FMPXMLLAYOUT": {
    "ERRORCODE": "0",
    "LAYOUT": {
      "DATABASE": "starbucks-us",
      "FIELD": [
        {
          "NAME": "Store ID",
          "STYLE": {
            "TYPE": "EDITTEXT",
            "VALUELIST": ""
          }
        },
        {
          "NAME": "Store Name",
          "STYLE": {
            "TYPE": "EDITTEXT",
            "VALUELIST": ""
          }
        },
        {
          "NAME": "Street 1",
          "STYLE": {
            "TYPE": "EDITTEXT"
          }
        },
        {
          "NAME": "Street 2",
          "STYLE": {
            "TYPE": "EDITTEXT"
          }
        },
        {
          "NAME": "Street 3",
          "STYLE": {
            "TYPE": "EDITTEXT"
          }
        },
        {
          "NAME": "City",
          "STYLE": {
            "TYPE": "EDITTEXT"
          }
        },
        {
          "NAME": "State",
          "STYLE": {
            "TYPE": "EDITTEXT"
          }
        }
      ]
    }
  }
}
```

```

    }
  },
  {
    "NAME": "Postal Code",
    "STYLE": {
      "TYPE": "EDITTEXT"
    }
  },
  {
    "NAME": "Phone Number",
    "STYLE": {
      "TYPE": "EDITTEXT"
    }
  },
  {
    "NAME": "Latitude",
    "STYLE": {
      "TYPE": "EDITTEXT"
    }
  },
  {
    "NAME": "Longitude",
    "STYLE": {
      "TYPE": "EDITTEXT"
    }
  }
],
"NAME": "Store List"
},
"PRODUCT": {
  "BUILD": "04/10/2018",
  "NAME": "FileMaker Web Publishing Engine",
  "VERSION": "17.0.1.146"
},
"VALUELISTS": ""
}
}

```

3. GetScriptNames : Obtenir une liste des scripts dans la BD

La requête pour obtenir une liste des scripts peut aussi être utile pendant le développement lorsqu'on n'a pas un accès direct à la base de données avec un client Filemaker :

```

{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",
  "Account": "xfm",
  "Password": "latte",
  "Action": "GetScriptNames"
}

```

Le résultat de cette requête a cette allure :

```

{
  "fmresultset": {

```

```

    "error": {
      "code": "0"
    },
    "resultset": {
      "count": "1",
      "fetch-size": "1",
      "record": [
        {
          "fields": {
            "SCRIPT_NAME": "Photo Container Set"
          },
          "mod-id": "0",
          "record-id": "0"
        }
      ]
    }
  }
}

```

4. GetContainer : Obtenir l'objet dans un champ dans la BD

La requête pour obtenir un objet (ex : image, PDF, ...) enregistré dans un champ Filemaker ressemblerait à ceci :

```

{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Account": "xfm",
  "Password": "latte",
  "Action": "GetContainer",
  "ContainerURL": "/fmi/xml/cnt/500px-Albertsons_(logo).svg.png?-db=starbucks-us&-
}

```

Dans cette requête, la valeur de ContainerURL aura été obtenue par une requête préalable utilisant un modèle contenant un champ de type « container ». Le résultat de cette requête est un objet dont le type se trouvera dans l'entête « Content-Type ».

5. SetContainer : Insérer un objet dans un champ dans la BD

La requête pour insérer un objet dans un champ de type container fait partie d'un processus plus complexe car l'insertion devra se faire en invoquant un script qui se chargera d'effectuer l'insertion et le fichier devra avoir été déplacé au préalable dans un répertoire ou être autrement accessible au serveur Filemaker.

Dans l'exemple présent, la requête effectue une recherche pour l'enregistrement visé qui aura été trouvée dans une recherche préalable qui nous aura permis d'en identifier dans le résultat son « RecordID ». La requête invoque un script en lui donnant le URL pour obtenir l'image à insérer dans le champ container.

```

{
  "Host": "localhost",
  "Protocol": "http",
  "Database": "starbucks-us",
  "Layout": "Store List",

```

```
"Account": "xfm",
"Password": "latte",
"Action": "GetRecord",
"RecordID": 12898,
"ScriptPreSort": "Photo Container Set",
"ScriptPreSortParam": "https://upload.wikimedia.org/wikipedia/commons/thumb/b/b6
}
```

Dans cette requête, la valeur de ContainerURL aura été obtenue par une requête précédente dans un modèle contenant un champ « container ». Le résultat de cette requête est un objet dont le type se trouvera dans l'entête « Content-Type ».

La réponse aura cette allure, où on voit bien le même RecordID et le champ container « Photo » qui nous présente un URL pour obtenir le document qu'on vient d'insérer :

```
{
  "fmresultset": {
    "error": {
      "code": "0"
    },
    "resultset": {
      "count": "1",
      "fetch-size": "1",
      "record": [
        {
          "fields": {
            "Store ID": "1015248",
            "Store Name": "Albertsons - Rio Rancho 923",
            "Street 1": "7800 Enchanted Hills Blvd NE",
            "Street 2": "",
            "Street 3": "",
            "City": "Rio Rancho",
            "State": "NM",
            "Postal Code": "87144",
            "Phone Number": "505-771-7082",
            "Latitude": "35.323337554931641",
            "Longitude": "-106.57643890380859",
            "Comments": "",
            "Photo": "VfmiVxmVcntV500px-Albertsons_(logo).svg.png?-db=starbuc"
          },
          "mod-id": "4",
          "record-id": "12898"
        }
      ]
    }
  }
}
```