

PROJET DE GESTION D'UNE CAVE A VIN



Cave à vin

Catalogue des vins Autres gestionnaires

Cave à vin



© 2023 Zakaria BELKACEMI : zakaribel.fr

Table des matières

1 – Compétences couvertes	
2 – Résumé du projet	
3 – Cahier des charges	
4 – Maquette de présentation	
5 – Gestion des vins	
5.1 L'onglet 'Catalogue des vins'	
5.1.1 Comment saisir un nouveau vin	
5.1.2 Comment apporter des modification sur un vin	
5.1.3 Comment supprimer un vin	
5.2.1 Gestion du tableau 'Pays'	
5.2 Gestion du tableau 'Région'	
5.2.3 Gestion du tableau 'Appellation'	
5.2.4 Gestion du tableau 'Couleur'	
DOSSIER TECHNIQUE	
6 - Schéma de la base de données	
7 – Les tables	

7.1 La structure des tables.....	
7.1.1 La table « VIN »	
7.1.2 La table « REGION ».....	
7.1.3 La table « PAYS »	
7.1.4 La table « APPELLATION »	
7.1.5 La table « COULEUR ».....	
7.2 L'accès aux tables	
7.3 Syntaxe pour accéder aux données dans les tables	
7.4 Les 'clés' des tables	
8 – Le Programme	
8.1 L'emploi des fichiers JavaScript.....	
8.1.1 Analyse du fichier init_api.js	
9 - Test	

1. Liste des compétences couvertes par le projet

Ce projet permet de mettre en évidence plusieurs compétences techniques dans le titre professionnel développeur web - web mobile. Les compétences sont inclut la capacité de maquetter une application afin de permettre au client et au professionnel de visualiser une ébauche du projet et de pouvoir se mettre d'accord quant au visuel. La compétence de réaliser une interface utilisateur web statique et adaptable, créer une interface visuelle tout en la rendant compatible avec différents appareils et tailles d'écran. Enfin, le développement d'une interface utilisateur web dynamique, ce qui implique la création d'éléments interactifs au sein de l'interface pour une expérience utilisateur plus riche et fluide.

2. Résumé du projet

Ce site de gestion de cave à vin est réalisé dans le cadre d'un projet de développement de la partie front-end d'une application web & web mobile. Un client me demande de réaliser un site pour gérer sa cave à vin. L'utilisateur doit pouvoir rentrer dans la base de donnée, créé auparavant, les bouteilles de sa cave à vin.

Il doit pouvoir visualiser le contenu de sa cave à un instant donné, avec toutes les concernant ses vins, mais également la possibilité d'ajouter de nouvelles bouteilles ou d'en retirer. Le travail consiste donc en la création d'un site de gestion de cave à vin.

L'objectif de ce projet et de satisfaire la demande de client souhaitant pouvoir gérer et enrichir la base de données de ses vins. Le site facilitera la recherche de vins spécifiques en fonction de divers critères tels que le nom du vin, la région viticole, l'appellation, et la couleur.

Il y a plusieurs avantages de ce projet, il permettra le gain du temps, l'efficacité et améliorer la gestion des stocks.

Maintenant, l'idée va être de réaliser un CRUD (Create - créer, Request- lire, Update – mettre à jour, Delete- supprimer) pour pouvoir mettre à jour les données. En plus, je vais utiliser l'API Rest et les demandes AJAX format JSON pour la mise à jour dynamique de l'interface utilisateur sans recharger la

page entière dans le navigateur. Les applications fonctionnent plus rapidement et sont plus réactives aux actions de l'utilisatrice ou de l'utilisateur.

3 - Cahier des charges

Un client nous a commandé une application qui lui permettra de gérer sa cave. Il voudra gérer son stock et gérer aussi les accords mets/vins.

Chaque vin appartient à un pays, une région, une appellation et un cépage sous la forme de références. Un vin a une couleur mais une appellation peut avoir plusieurs couleurs de vin.

Un vin peut s'accorder avec plusieurs mets et inversement. Le gestionnaire de la cave peut gérer les références, les accords mets/vins, les stocks et peut consulter les événements en rapport avec la dégustation d'un vin.

Le sommelier peut gérer les accords mets/vins et les stocks.

4 – Maquette de présentation

La maquette ci-dessous donne une idée de la présentation du site de gestion d'une cave à vin.



5 – Utilisation du programme de Gestion des vins

Ce programme permet la gestion d'une cave à vin. La page d'accueil se présente sous la forme suivante :



La barre de navigation est constituée de 3 onglets :

- Accueil
- Catalogue des vins
- Autres gestionnaires

5.1 L'onglet 'Catalogue des vins'

En cliquant sur cette onglet on accède à l'ensemble des vins répertoriés dans cette application comme le montre la saisie d'écran ci-dessous :

Code Vin	Nom du Vin	Appellation	Région	Couleur	Culture	Commentaires	Actions
1	Barbera d'Alba	Alsace	Abruzzes	Blanc	Conventionnelle	Un vin italien frais et fruité, mettant en valeur le cépage Barbera caractéristique du Piémont	
2	Priorat	Priorat DOCA	Catalogne	Rouge	Biologique	Un vin espagnol intense et concentré, issu des terrasses escarpées de la région du Priorat	
3	Brunello di Montalcino	Brunello Di Montalcino DOGC	Toscane	Rouge	Biodynamique	Un vin toscan élégant et complexe, vieilli pendant de longues périodes avant d'être mis en bouteille	
5	Chianti Classico	Chianti Classico DOGC	Toscane	Rouge	Biologique	Un vin italien fruité et équilibré, issu principalement du	

On retrouve la dénomination des vins sous forme d'un tableau constitué de plusieurs colonnes.

La première colonne « Code Vin » est le code généré par l'application au fur et à mesure de la saisie des vins. Ce code sera incrémenté automatiquement.

La deuxième colonne « Nom du vin » contient le nom du vin. Ce vin sera saisi au moment de sa création dans l'application.

La troisième colonne « Région » indique l'origine du vin. La région d'origine sera choisie au moment de la création du vin dans l'application.

La quatrième colonne « Couleur » donne le couleur du vin. Là encore, la couleur sera choisie au moment de la création du vin dans l'application.

La cinquième colonne « Culture » permet d'ajouter, si besoin, des précisions sur le type de culture.

La sixième colonne « Commentaires » permet de mettre des précisions que l'on souhaite avoir à propos du vin.

5.1.1 Comment saisir un nouveau vin

A partir de la page « Liste des vins » (accessible en cliquant sur 'Catalogue des vins' dans la barre de navigation), cliquez sur le bouton « +Nouveau » :

The screenshot shows the 'Cave à vin' application interface. At the top, there's a header with the Afpa logo and the session dates. Below the header, a 'Liste d' menu is visible. A modal form is open, allowing users to add a new wine entry. The form includes fields for 'Appellation', 'Couleur', 'Région', 'Nom du vin', 'Culture', and 'Commentaires'. A '+ Nouveau' button is highlighted with a blue circle. The background shows a table of existing wine entries with columns for 'Code Vin', 'Nom du Vin', 'Appellation', 'Couleur', 'Région', 'Culture', and 'Commentaires'.

Il suffit alors de renseigner les différents champs contenus dans la fenêtre. Dans le champ 'vin' on saisira le nom du vin que l'on souhaite ajouter dans l'application.

Pour les champs 'Appellation', 'Région' et 'Couleur', le fait de cliquer sur la petite flèche présente en dessous de chaque champ affichera une liste. Il suffira de faire dérouler la liste jusqu'à la dénomination qui nous intéresse et de cliquer dessus pour la sélectionner comme le montre la saisie d'écran.

This image shows a close-up of the 'Appellation' dropdown menu. The menu is open, displaying a list of wine appellations including Alsace, Priorat DOCA, Brunello Di Montalcino DOGC, Chianti Classico DOGC, Alsace, Bourgogne, Rioja DOCA, Marlborough, Chateaufeuf du Pape, Kanonkop Estate Pinotage, Barossa Valley, Barossa Valley, Mosel, Cahors AOC, Amarone Della Valpoltisela DOGC, Sauternes, Barolo DOGC, Champagne, and Margaux. The 'Appellation' label is at the top of the dropdown.

Fermer Enregistrer

Pour terminer la saisie, cliquez sur « Enregistrer ». Le nouveau 'vin' est maintenant enregistré et présent dans la liste.

Remarque importante :

La saisie d'un nouveau vin est conditionnée au renseignement préalable de certaines tables de l'application. Les tableaux « Pays », « Région », « Appellation » et « Couleur » devront être renseignés avant la saisie d'un nouveau vin.

Cette partie sera abordée dans le chapitre 2.2 de la présente notice d'utilisation.

Une fois un vin ajouté, une fenêtre apparaît comme suit :



L'utilisateur est à présent invité à cliquer sur la flèche afin d'y constater le vin fraîchement ajouté tout en bas de page :

Cave à vin Catalogue des vins Autres gestionnaires

Liste des vins

Entrez un mot clé.

+ Nouveau

Code Vin	Nom du Vin	Appellation	Région	Couleur	Culture	Commentaires	Actions
1	Barbera d'Alba	Alsace	Abruzzes	Blanc	Conventionnelle	Un vin italien frais et fruité, mettant en valeur le cépage Barbera caractéristique du Piémont	  
2	Priorat	Priorat DOCA	Catalogne	Rouge	Biologique	Un vin espagnol intense et concentré, issu des terrasses escarpées de la région du Priorat	  
3	Brunello di Montalcino	Brunello Di Montalcino DOGC	Toscane	Rouge	Biodynamique	Un vin toscan élégant et complexe, vieilli pendant de longues périodes avant d'être mis en bouteille	  
5	Chianti Classico	Chianti Classico DOCG	Toscane	Rouge	Biologique	Un vin italien fruité et équilibré, issu principalement du	  

5.1.2 Comment apporter des modifications sur un vin

A partir de la page « La liste des vins » (accessible en cliquant sur 'Catalogue des vins' dans la barre de navigation), il y a 3 boutons en face de chaque ligne contenant le nom d'un vin.

Cliquez sur le bouton représentant un crayon :



Une fenêtre s'ouvre :

Cave à vin

Liste des vins

Entrez un mot clé.

+ Nouveau

Modifier l'Appellation

Alsace

Modifier la Couleur

Rouge

Modifier la Région

Abruzzes

Modifier le Nom du vin

Barbera d'Alba













Modifier la Culture

Conventionnelle

Modifier le Commentaire

Un vin italien frais et fruité, mettant en valeur le cépage Barbera caractéristique du Piémont

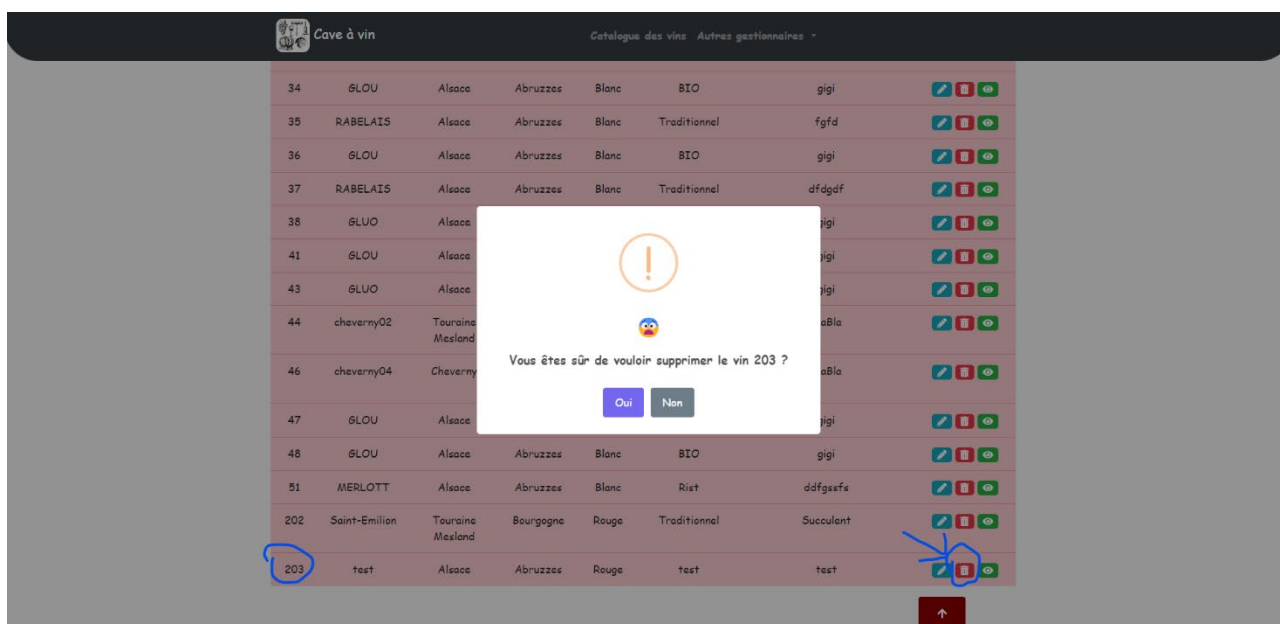
Fermer Enregistrer

Code Vin	Nom du Vin	Appellation	Région	Couleur	Culture	Commentaires	Actions
1	Barbera d'Alba	Alsace	Abruzzes	Blanc	Conventionnelle	Un vin italien frais et fruité, mettant en valeur le cépage Barbera caractéristique du Piémont	  
2	Priorat	Priorat DOCA	Catalogne	Rouge	Biologique	Un vin espagnol intense et concentré, issu des terrasses escarpées de la région du Priorat	  
3	Brunello di Montalcino	Brunello Di Montalcino DOGC	Toscane	Rouge	Biodynamique	Un vin toscan élégant et complexe, vieilli pendant de longues périodes avant d'être mis en bouteille	  
5	Chianti Classico	Chianti Classico DOCG	Toscane	Rouge	Biologique	Un vin italien fruité et équilibré, issu principalement du	  

Apporter les modifications désirées et cliquez sur le bouton « Enregistrer ». La modification est alors faite.

5.1.3 Comment supprimer un vin

A partir de la page « La gestion des vins » (accessible en cliquant sur 'Catalogue des vins' dans la barre de navigation), il y a 3 boutons en face de chaque ligne contenant le nom d'un vin.



Le message suivant apparaît :

Cliquez sur le bouton «OUI ». La suppression est alors effectuée. Si l'on ne souhaite pas effectuer la suppression il suffit de cliquer sur le bouton « NON ». Le message suivant apparaîtra :



Vin numéro 203 supprimé

OK

5.2 L'onglet « Autres gestionnaires »

Comme abordé à la fin du chapitre 2.2.1 de la présente notice, il est important de procéder aux renseignements de certaines tables avant la saisie d'un nouveau vin.

Pour cela il est nécessaire d'aller sur l'onglet 'Autres gestionnaires' présent dans la barre de navigation.

L'action de cliquer sur cet onglet ouvre une liste ou apparaît toutes les tables nécessaires au fonctionnement de l'application.



5.2.1 Gestion du tableau 'Pays'

Avant toute saisie d'un nouveau vin, la table 'Pays' doit être renseignée. Pour cela il suffit de cliquer sur 'Pays' présent dans la liste comme montré ci-dessous :

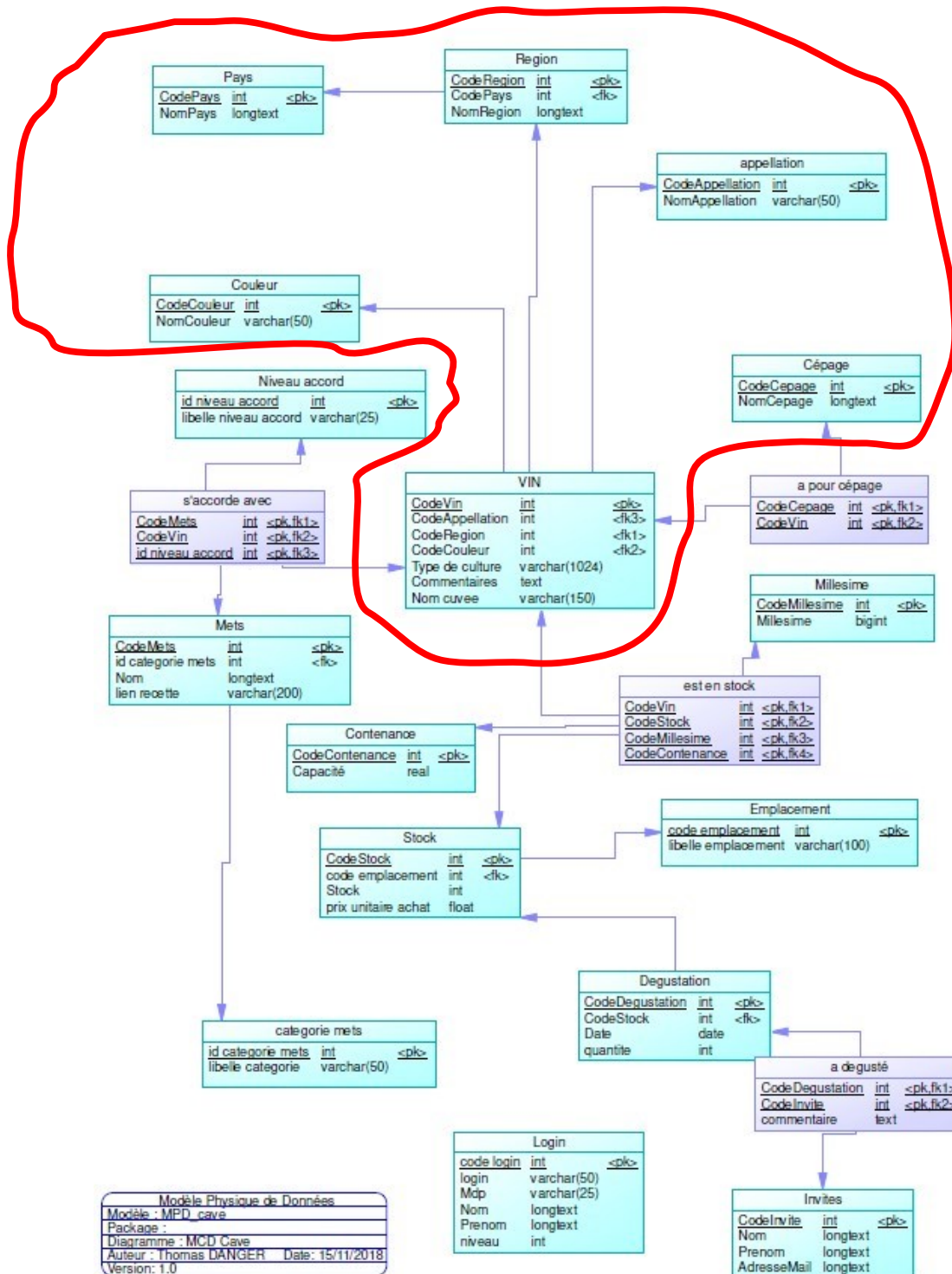


L'action de cliquer sur 'Pays' va ouvrir une nouvelle page qui ressemblera à celle vu pour la gestion des vins. A savoir que 'Pays' ainsi que tous les autres gestionnaires auront exactement le même fonctionnement que la gestion des vins. Fonctionnement abordé précédemment.

DOSSIER TECHNIQUE

6 - Schéma de la base de données

Le schéma ci-dessous est le M.C.D (Modèle Conceptuel des données) issu de l'analyse du projet avec la méthode « MERISE » qui est l'acronyme pour « Méthode d'étude et de réalisation informatique pour les systèmes d'entreprise ».



Le M.C.D. va mettre en évidence les relations existantes entre les différentes entités.

Ainsi, les tables qui seront dans un premier temps utilisées sont les tables :

- PAYS
- REGION
- APPELLATION
- COULEUR
- VIN

On remarque que la table vin est générée en faisant appel aux contenus de nombreuses autres tables.

7 – Les tables

Les tables seront consultées à partir d'une API (Application Programming Interface ou Interface de programmation d'application en français).

Concrètement, ce terme désigne un ensemble de protocoles à travers lequel un site web ou un logiciel met tout ou partie de sa **base de données** à la disposition des utilisateurs. Cette interface de programmation constitue, entre autres, une sorte de portail permettant d'accéder et d'interagir avec les fonctionnalités d'une application existante.

Il existe deux types d'API :

- les API SOAP (Simple Object Access Protocol).

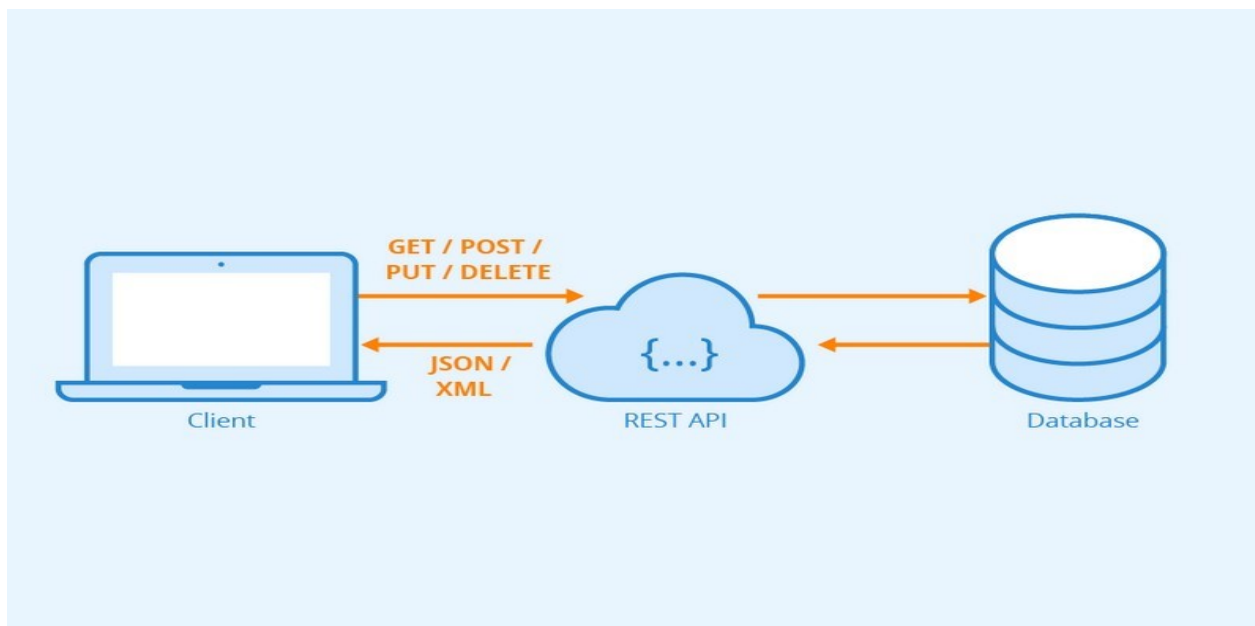
Les **API** qui suivent un protocole SOAP transmettent leurs requêtes sous format XML. Il s'agit d'un ancien protocole d'accès dont l'usage peut s'avérer assez encombrant.

- les API REST (Representationnal State Transfert).

les **APIs** basés sur l'architecture REST, appelés API RESTful, sont beaucoup plus flexibles et rapides que celles utilisant le protocole SOAP.

Comment fonctionne une API REST ?

Comme le montre le schéma ci-dessous un client va faire des requêtes AJAX (Asynchronous JavaScripts And XML) de type « GET/POST/PUT et DELETE » à l'API qui va consulter la « BDD » (base de données). La BDD retournera la réponse à l'API qui à son tour la renverra au client.



La syntaxe « GET » permet de lire les données contenues dans la BDD.

La syntaxe « POST » permet d'écrire une donnée dans la BDD.

La syntaxe « PUT » permet de modifier la donnée dans la BDD.

Enfin, la syntaxe « DELETE » va effacer la donnée de la BDD.

En règle générale, on appelle « CRUD » (Create Read Update Delete) l'ensemble de l'utilisation des syntaxes GET/POST/PUT/DELETE.

Create : POST

Read : GET

Update : PUT

Delete : DELETE

7.1 La structure des tables

7.1.1 La table « VIN »

La table 'VIN' se présente sous la forme suivante :

```
{
  "VIN": {
    "columns": [
      "CODEVIN",
      "NOI_CUVEE",
      "CODEAPPELLATION",
      "CODEREGION",
      "CODECOULEUR",
      "TYPE_DE_CULTURE",
      "COMMENTAIRES"
    ],
    "records": [
      [
        1,
        "Domaine Dirler-Cade ",
        2,
        2,
        2,
        "Traditionnel",
        "Le nez est fruité, boisé, épicé avec des arômes de raisin frais. La bouche est sec et intense sur une belle longueur."
      ],
      [
        2,
        "Domaine Vissoux",
        4,
        25,
        61,
        "Culture raisonné",
        "Ce vin blanc du mill"
      ],
      [
        3,
        "Château Margaux",
        39,
        25,
        1,
        "Traditionnel",
        "Un nez splendide, profond et plein de complexité aromatique. On y retrouve des notes de fruits noirs et de poivre. En bouche, on retrouve toute la puissance du fruit. "
      ],
      [
        15,
        "Paul Bara ",
        4,
        5,
        4,
        "Traditionnel",
        "Au nez ce vin de Champagne révèle des arômes fruités et épicés. En bouche, ce vin de Champagne du domaine Paul Bara allie des saveurs fruitées et acidulées avec une longue persistance."
      ],
      [
        16,
        "Cuvée de Bourgoene".
      ]
    ]
  }
}
```

7.1.2 La table « REGION »

La table 'REGION' se présente sous la forme suivante :

← → ↻ Non sécurisé | fbr.esy.es/DWWM21050/Api/api.php/REGION/

```
{
  "REGION": {
    "columns": [
      "CODEREGION",
      "CODEPAYS",
      "NOMREGION"
    ],
    "records": [
      [
        1,
        1,
        "Val de Loire"
      ],
      [
        2,
        1,
        "Alsace"
      ],
      [
        3,
        1,
        "Aquitaine"
      ],
      [
        4,
        34,
        "Vallée du Rhone"
      ],
      [
        5,
        1,
        "Champagne"
      ],
      [
        7,
        2,
        "Toscane"
      ],
      [
        8,
        2,
        "Piemont"
      ],
      [
        10,
        3,
        "Navarre"
      ],
      [
        11,
        1,
        "Beaujolais"
      ]
    ]
  }
}
```

7.1.3 La table « PAYS »

La table 'PAYS' se présente sous la forme suivante :

```
{
  "PAYS": {
    "columns": [
      "CODEPAYS",
      "NOMPAYS"
    ],
    "records": [
      [
        1,
        "FRANCE"
      ],
      [
        2,
        "ITALIE"
      ],
      [
        3,
        "ESPAGNE"
      ],
      [
        4,
        "BELGIQUE"
      ],
      [
        22,
        "USA"
      ],
      [
        23,
        "PORTUGAL"
      ],
      [
        24,
        "AMERIQUE"
      ],
      [
        30,
        "AUSTRALIE"
      ],
      [
        31,
        "NOUVELLE ZELANDE"
      ],
      [
        32,
        "ALLEMAGNE"
      ],
      [
        33,
        "CANADA"
      ]
    ]
  }
}
```

7.1.4 La table « APPELLATION »

La table 'APPELLATION' se présente sous la forme suivante :

← → ↻ Non sécurisé | fbr.esy.es/DWWM21050/Api/api.php/APPELLATION

```
{
  "APPELLATION": {
    "columns": [
      "CODEAPPELLATION",
      "NOMAPPELLATION"
    ],
    "records": [
      [
        1,
        "Morgon"
      ],
      [
        2,
        "Morgon"
      ],
      [
        3,
        "Chardonnay"
      ],
      [
        4,
        "Chardonnay, Pinot noir"
      ],
      [
        5,
        "Côte de Provence"
      ],
      [
        6,
        "Champagne"
      ],
      [
        33,
        "Rosée"
      ],
      [
        39,
        "Margaux"
      ],
      [
        41,
        "Savagnin"
      ],
      [
        71,
        "Côte-Roannaise"
      ],
      [
        73,
        "Côtes du jura"
      ]
    ]
  }
}
```

7.2 L'accès aux tables

Les tables sont accessibles à partir de l'url suivante :

<http://fbr.esy.es/DWWM21050/Api/api.php/>

A laquelle il faut ajouter le nom de la table que l'on souhaite consulter (en majuscule).

Par exemple, pour consulter la table 'PAYS' on renseignera l'url suivante :

<http://fbr.esy.es/DWWM21050/Api/api.php/PAYS>

Pour consulter la table 'REGION' on renseignera l'url suivante :

<http://fbr.esy.es/DWWM21050/Api/api.php/REGION>

Pour consulter la table 'APPELLATION' on renseignera l'url suivante :

<http://fbrc.esy.es/DWWM21050/Api/api.php/APPELLATION>

Pour consulter la table 'COULEUR' on renseignera l'url suivante :

<http://fbrc.esy.es/DWWM21050/Api/api.php/COULEUR>

Pour consulter la table 'VIN' on renseignera l'url suivante :

<http://fbrc.esy.es/DWWM21050/Api/api.php/VIN>

7.3 Syntaxe pour accéder aux données dans les tables

Ci-dessous un extrait de la table 'VIN' :

```
{
  "VIN": {
    "columns": [
      "CODEVIN",
      "NOM_CUVEE",
      "CODEAPPELLATION",
      "CODEREGION",
      "CODECOULEUR",
      "TYPE_DE_CULTURE",
      "COMMENTAIRES"
    ],
    "records": [
      [
        1,
        "Domaine Dirler-Cade ",
        2,
        2,
        2,
        "Traditionnel",
        "Le nez est fruité, boisé, épicé avec des arômes de raisin frais. La bouche est sec et intense sur une belle longueur"
      ],
      [
        2,
        "Domaine Vissoux",
        4,
        25,
        8,
        "Culture maison",
        "Ce vin blanc du mil51"
      ]
    ]
  }
}
```

La table 'VIN' est constituée de plusieurs « colonnes » comme 'CODEVIN', 'NOM_CUVEE', 'CODEAPPELLATION', 'CODEREGION'....


On pourrait apparenter cette table 'VIN' à une page d'un tableur qui aurait la structure suivante :

	A	B	C	D	E	F	G
1							
2	Table VIN						
3							
4	CODEVIN	NOM_CUVEE	CODEAPPELLATION	CODEREGION	CODECOULEUR	TYPE_DE_CULTURE	COMMENTAIRES
5							
6	1	Domaine Dirler-Cade	2	2	2	Traditionnel	Le nez est fruité, boisé, épicé avec des arômes de raisin frais. La bouche est sec et intense sur une belle longueur.
7							
8	2	Domaine Vissoux	4	25	8	Culture maison	Ce vin blanc du mil51
9							
10
11							

Ainsi, pour accéder à la colonne 'CODEVIN' de la table 'VIN', la syntaxe est :

<http://fbrc.esy.es/DWWM21050/Api/api.php/VIN?columns=CODEVIN>

On obtient alors ceci :



```
{
  "VIN": {
    "columns": [
      "CODEVIN"
    ],
    "records": [
      [
        22
      ],
      [
        29
      ],
      [
        37
      ],
      [
        240
      ],
      [
        1
      ],
      [
        16
      ]
    ]
  }
}
```

List + Column selection

By default all columns are selected. With the "columns" parameter you can select specific columns. Multiple columns should be comma separated. An asterisk ("*") may be used as a wildcard to indicate "all columns":

```
...
GET http://localhost/api.php/categories?columns=name
GET http://localhost/api.php/categories?
columns=categories.name
...
```

Extrait de la page 5 de la documentation de l'API - Documentation_comment_utiliser_API.pdf Par analogie, c'est un peu comme si on avait sélectionné que la colonne 'CODEVIN' de notre

tableur :

2	Table VIN						
3							
4	CODEVIN	NOM_CUVEE	CODEAPPELLATION	CODEREGION	CODECOULEUR	TYPE_DE_CULTURE	COMMENTAIRES
5							
6	1	Domaine Dider-Cade	2	2	2	Traditionnel	Le nez est fruité, boisé, épicé avec des arômes de raisin frais. La bouche est sec est intense sur une belle longueur.
7							
8	2	Domaine Vissoux	4	25	8	Culture maison	Ce vin blanc du mil51
9							
10

On remarque par contre que la table n'est pas ordonnée. En effet, sous 'records' il est affiché une suite de nombres '22', '29', '37', '240'...

Pour obtenir un résultat ordonné, il faut taper la syntaxe suivante :

<http://fbrc.esy.es/DWWM21050/Api/api.php/VIN?columns=CODEVIN&order=CODEVIN,asc>

On notera que l'on a simplement rajouter &order=CODEVIN,asc asc = tri ascendant desc = tri descendant

List + Order

With the "order" parameter you can sort. By default the sort is in ascending order, but by specifying "desc" this can be reversed:

...

GET http://localhost/api.php/categories?order=name,desc

GET http://localhost/api.php/posts?

order[]=icon,desc&order[]=name

...

Extrait de la page 6 de la documentation de l'API - Documentation_comment_utiliser_API.pdf

Ainsi, pour accéder à la colonne 'NOM_CUVEE' de la table 'VIN' triée, la syntaxe est :

http://fbrc.esy.es/DWWM21050/Api/api.php/VIN?columns=NOM_CUVEE&order=NOM_CUVEE,asc

```
{
  "VIN": {
    "columns": [
      "CODEVIN"
    ],
    "records": [
      [
        1
      ],
      [
        2
      ],
      [
        3
      ],
      [
        15
      ],
      [
        16
      ]
    ]
  }
}
```

Il est aussi possible d'avoir plusieurs colonnes qui seront triées sur l'une d'elle. Par exemple, pour afficher que les colonnes 'CODEVIN', 'NOM_CUVEE' et 'APPELLATION' et triées sur la colonne 'APPELLATION', la syntaxe est la suivante :

http://fbrc.esy.es/DWWM21050/Api/api.php/VIN?columns=CODEVIN,NOM_CUVEE,CODEAPPELLATION&order=CODEAPPELLATION,asc

On obtient alors la table sous la forme suivante :

```
{
  "VIN": {
    "columns": [
      "CODEVIN",
      "NOM_CUVEE",
      "CODEAPPELLATION"
    ],
    "records": [
      [
        240,
        "Test 01",
        1
      ],
      [
        29,
        "France",
        1
      ],
      [
        22,
        "Viognier ",
        1
      ],
      [
        37,
        "JE PEUX PAS SUP",
        1
      ],
      [
        1,
        "Domaine Dirler-Cade ",
        2
      ],
      [
        35,
        "cqcq",
        2
      ]
    ]
  }
}
```

On remarque que le trie s'effectue sur le troisième argument.

7.4 Les 'clés' des tables

Chaque table est constituée d'une clé primaire ou « pk » (primary key). Par exemple, la clé primaire ou « pk » de la table 'PAYS' est 'CODEPAYS'. La clé primaire ou « pk » de la table 'REGION' est 'CODEREGION'. La clé primaire ou « pk » de la table 'VIN' est 'CODEVIN'. Et ainsi de suite pour chacune des tables (cf chapitre 3 'Schéma de la base de données').

Certaines tables ne possèdent que des clé primaire « pk » comme les tables « PAYS », « APPELLATION », « COULEUR » et « CEPAGE ».

D'autres tables comme « REGION » possèdent une clé primaire « pk » et une clé étrangère « fk » (foreign key). Pour la table « REGION » la clé primaire « pk » est « CODEREGION » et la clé étrangère « fk » est « CODEPAYS ».

Enfin, certaines tables, comme la table « VIN » possèdent une clé primaire « pk » et plusieurs clés étrangères « fk ». Pour la table « VIN » la clé primaire « pk » est « CODEVIN » et les clés étrangères « fk » sont « CODEREGION », « CODECOULEUR » et « CODEAPPELLATION ».

En quoi consiste une clé étrangère « fk » ?

Le site W3schools.com l'explique très bien. Voici un extrait contenu dans la page :

https://www.w3schools.com/sql/sql_foreignkey.asp

« A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table. »

8 – Le Programme

Le programme se compose de 6 pages HTML :

- index.html (page d'accueil)
- PAYS.html
- REGIONS.html
- APPELLATIONS.html
- COULEURS.html
- VINS.html

Chaque page html est associé à un fichier JavaScript. Ainsi, les pages html suivantes sont associées au fichiers JavaScript suivants :

- pays.html → pays.js + init.js
- regions.html → regions.js + init.js
- appellation.html → appellation.js + init.js
- couleur.html → couleur.js + init.js
- vins.html → vins.js + init.js

8.1 Les pages html

Les pages html intègrent différents liens nécessaires à leurs fonctionnements. Le lien ci-dessous permet la gestion des icônes présentes dans le programme.

```
<link
  rel="stylesheet"
  href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css"
  integrity="sha384-AYmEC3YW5cVb3ZcuHt0A93w35dYTsvhLPVnYs9eStHfGJv0vKxVfELGroGkvsg+p"
  crossorigin="anonymous"
/>
```

Ce lien va permettre de générer les icônes 'voir', 'modifier' et 'supprimer' et auront les représentations suivantes :



Le lien ci-dessous permet la gestion de la librairie JQuery pour les syntaxes écrites en JQuery présentes dans le programme.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

Le lien ci-dessous permet la gestion d'alertes en utilisant la bibliothèque « sweetAlert » dans le programme.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/sweetalert2@11.0.19/dist/sweetalert2.min.css">
```

Les liens ci-dessous permettent l'utilisation de Bootstrap afin d'améliorer la présentation des pages html dans le programme, ainsi que la possibilité de le rendre web responsive .

```
<link
  rel="stylesheet"
  href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
/>
<link
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

8.1 L'emploi des fichiers JavaScript

Chaque page html fera appel à un script JavaScript avec, comme par exemple, la syntaxe utilisée ci-dessous dans la page VINS .html :

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<script src=" ../js/init_api.js"></script>
<script src=" ../js/vins.js"></script>
```

Cela permet, entre autre, de séparer le code JavaScript du code html. Cela permet aussi l'importation de différents modules indépendamment les uns des autres.

8.1.1 Analyse du fichier init_api.js

Le fichier init_api.js est plutôt simple. Il contient toute les variables contenant les différentes adresses de l'API.

```
JS init_api.js X
js > JS init_api.js > ...
    oui zeki, 4 days ago | 1 author (oui zeki)
1  const urlApiAppellation = "https://afpafabrice.space/DWWM22239/Api/api.php/APPELLATION";
2  const urlApiCouleur = "https://afpafabrice.space/DWWM22239/Api/api.php/COULEUR";
3  const urlApiPays = "https://afpafabrice.space/DWWM22239/Api/api.php/PAYS";
4  const urlApiRegion = "https://afpafabrice.space/DWWM22239/Api/api.php/REGION";
5  const urlApiVins = "https://afpafabrice.space/DWWM22239/Api/api.php/VIN";
6
```

Prenons l'exemple de VINS.html :

```
<script src="../js/init_api.js"></script>
<script src="../js/vins.js"></script>
```

La page de la liste des vins est liée à mon programme principal (vins.js) et également lié à init_api.js. De ce fait, vins.js, init_api.js et VINS.html sont tous liés.

C'est comme cela que j'ai accès à ma variable constante « urlApiVins » dans mon programme principal (vins.js).

A présent, prenons l'exemple de l'affichage des données de l'API vins.

Dans ce cas, dans ma fonction « getWines », je vais simplement avoir besoin d'appeler la constante « urlApiVins » :

```
function getWines(urlApiVins) {  
  let requestOptions = {  
    method: "GET",  
    redirect: "follow",  
  };  
  
  fetch(urlApiVins + "?include=COULEUR,REGION,APPELLATION&transform=1", requestOptions)  
    .then((response) => response.json())  
    .then(function (data) {  
      wines = data;  
  
      display();  
      deleteWine();  
      editWine();  
      viewWine();  
    })  
    .catch(function (error) {  
      console.log("Une erreur s'est produite lors de la récupération des données :", error);  
    });  
}
```

Je mets les données que me retourne l'api dans une variable globale « wines »

Ma fonction « display() » va ainsi pouvoir récupérer les données et les traiter afin de les afficher dans ma page VINS.html :

```
//DISPLAY //

function display(){

  wines.VIN.map(function (vinElement) {
    let tr = document.createElement("tr");
    tr.setAttribute('class','data');

    let codeVinTd = document.createElement("td");
    codeVinTd.textContent = vinElement.CODEVIN;

    let nomVinTd = document.createElement("td");
    nomVinTd.textContent = vinElement.NOM_CUVEE;

    let appellationTd = document.createElement("td");
    appellationTd.textContent = vinElement.CODEAPPELLATION;

    let regionTd = document.createElement("td");
    let couleurTd = document.createElement("td");
    let cultureTd = document.createElement("td");
    let commentairesTd = document.createElement("td");
    let actionsTd = document.createElement("td");
    actionsTd.style.width = "150px";

    // ici je cherche à savoir si le code couleur de l'api COULEUR (color.CODECOULEUR)
    // match bien avec le code couleur de l'api VIN (vinElement.CODECOULEUR) et ce, en utilisant la fonction "find"
    let colorObj = vinElement.COULEUR.find((color) => color.CODECOULEUR === vinElement.CODECOULEUR);
    // A savoir que vinElement.COULEUR tout simplement car quand j'ai fait mon call api, j'ai ajouté l'api COULEUR dans mon include
    //c'est pour ça que j'y ai accès

    if (colorObj) {
      // si la fonction find renvoie true (c'est à dire qu'il ya un match) alors on recupere la valeur NOMCOULEUR de l'api COULEUR
      couleurTd.textContent = colorObj.NOMCOULEUR;
    } else {
      couleurTd.textContent = "Couleur non trouvée, désolé !!";
    }

    // à la recherche de THE region :)
    let regionObj = vinElement.REGION.find((region) => region.CODEREGION === vinElement.CODEREGION);
    if (regionObj) {
      regionTd.textContent = regionObj.NOMREGION;
    }
  })
}
```

```

    } else {
      regionTd.textContent = "Region non trouvée sorry!";
    }

    // à la recherche de THE appellation :)
    let appellationObj = vinElement.APPELLATION.find((appellation) => appellation.CODEAPPELLATION === vinElement.CODEAPPELLATION);
    if (appellationObj) {
      appellationTd.textContent = appellationObj.NOMAPPELLATION;
    } else {
      appellationTd.textContent = "appellation non trouvée sorry!!";
    }
  }

  cultureTd.textContent = vinElement.TYPE_DE_CULTURE;
  commentairesTd.textContent = vinElement.COMMENTAIRES;

  actionsTd.innerHTML = `
    <button class="modif btn btn-info btn-sm fas fa-pencil-alt fa-sm"></button>
    <button class="delete btn btn-danger btn-sm fas fa-trash-alt fa-sm"></button>
    <button class="view btn btn-success btn-sm fas fa-eye fa-sm"></button>
  `;

  tr.appendChild(codeVinTd);
  tr.appendChild(nomVinTd);
  tr.appendChild(appellationTd);
  tr.appendChild(regionTd);
  tr.appendChild(couleurTd);
  tr.appendChild(cultureTd);
  tr.appendChild(commentairesTd);
  tr.appendChild(actionsTd);

  table.appendChild(tr);
  let count = 0;

  for (let i = 0; i < table.rows.length; i++) { // je check le nombre de lignes générées en fonction de ce qu'il y a dans l'API
    if (table.rows[i].nodeName === "TR") {
      count++;
    }
  }

  if (count > limit) { // si le nombre de tr dépasse la limite alors j'affiche le bouton qui me permet de scroll tout en bas
    scrollTopButton.style.visibility = "visible";
    scrollDownButton.style.visibility = "visible";
  } else {
    scrollTopButton.style.visibility = "hidden";
    scrollDownButton.style.visibility = "hidden";
  }
}

```










Mes th étant déjà pré codés dans VINS.html et donc non généré par mon js, voici le résultat :

Cave à vin

Catalogue des vins Autres gestionnaires

Liste des vins

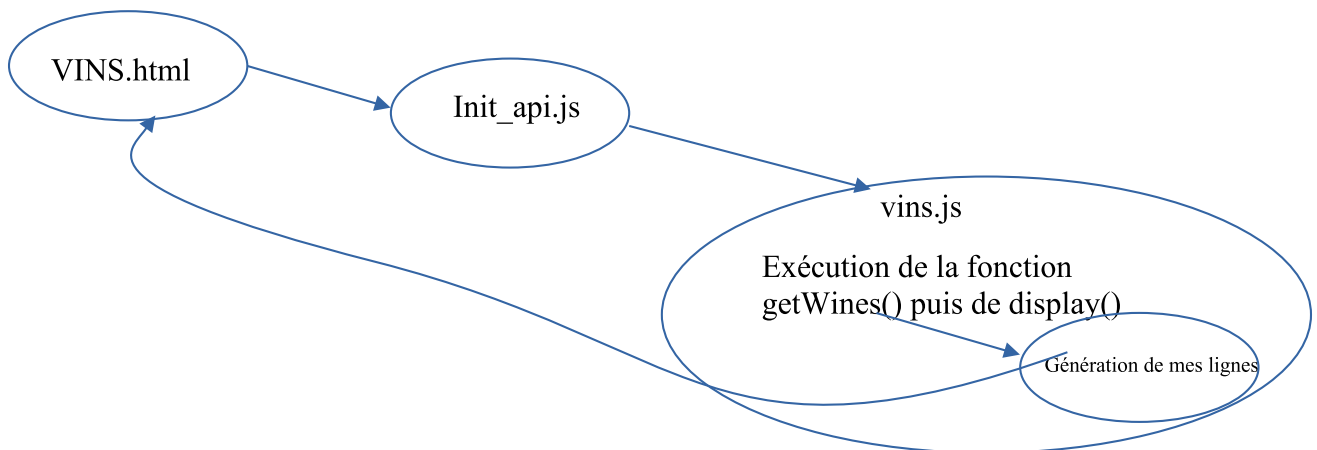
Nouveaux

Code Vin	Nom du Vin	Appellation	Région	Couleur	Culture	Commentaires	Actions
1	Barbera d'Alba	Alba	Abruzzo	Blanc	Conventionnelle	Un vin italien frais et fruité, mettant en valeur le cépage Barbera caractéristique du Piémont	  
2	Priorat	Priorat DOCA	Catalogne	Rouge	Biologique	Un vin espagnol intense et concentré, issu des terroirs escarpés de la région du Priorat	  
3	Brunello di Montalcino	Brunello Di Montalcino DOGC	Toscane	Rouge	Biodynamique	Un vin toscan élégant et complexe, vieilli pendant de longues périodes avant d'être mis en bouteille	  
5	Chianti Classico	Chianti Classico DOGB	Toscane	Rouge	Biologique	Un vin italien fruité et équilibré, issu principalement du	  

En résumé :

L'utilisation du fichier 'init_api.js' permet d'avoir pour chaque page html la même présentation. Ainsi, plutôt que de recréer plusieurs fichier init pour chaque api différente, on crée un seul fichier init_api contenant toutes les adresses de l'api.

Si l'aspect des pages html doit évoluer, il suffira de modifier la variable à laquelle on fait appel dans nos programmes principaux.



Ce fonctionnement est identique pour les pages 'appellation.html', 'couleur.html' et 'regions.html'.

9 - Test

Page/Test	Affichage	Nouveau	Visualisation	Modification	Suppression	Recherche
index	OK	-	-	-	-	-
Pays	OK	OK	OK	OK	OK	OK
Région	OK	OK	OK	OK	OK	OK
Appellation	OK	OK	OK	OK	OK	OK
Couleur	OK	OK	OK	OK	OK	OK
Catalogue des Vins	OK	OK	OK	OK	OK	OK