Web dynamique

Ajax/Json/MVC

1 OBJECTIFS

Passage d'une page statique à dynamique.

Transmission d'informations au format Json avec Ajax.

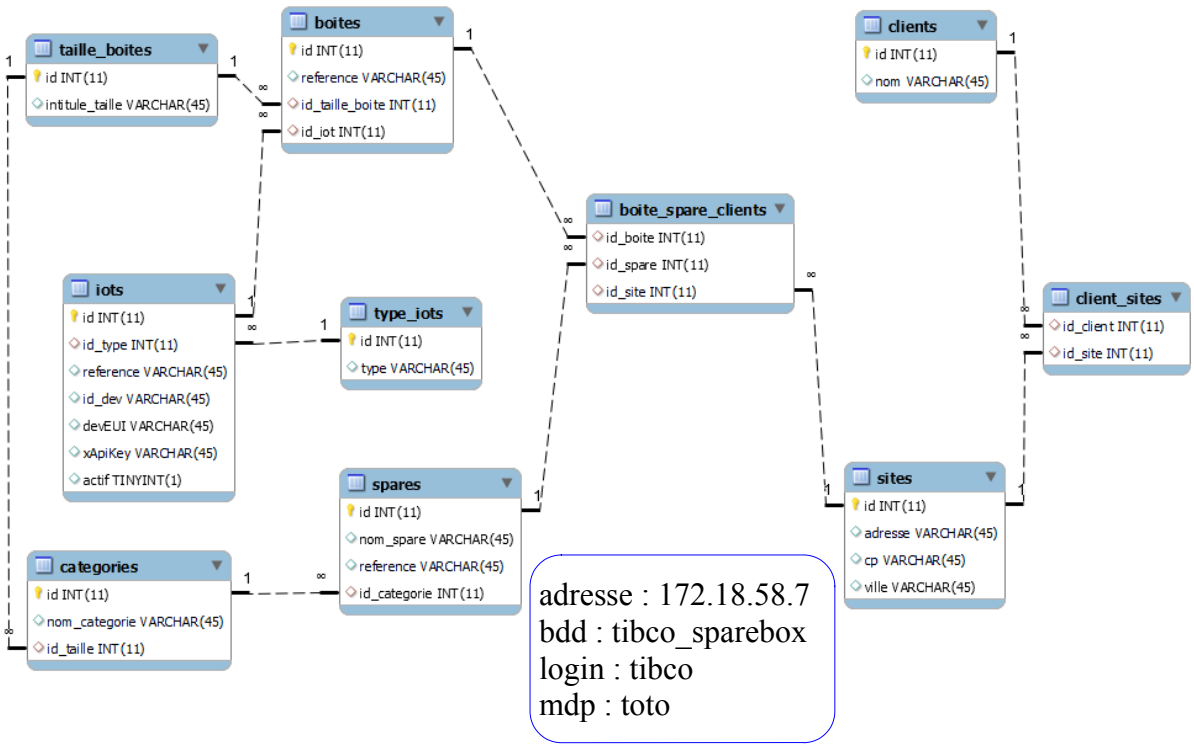
Architecture MVC.

2 SUPPORT

Gestion des sparebox.

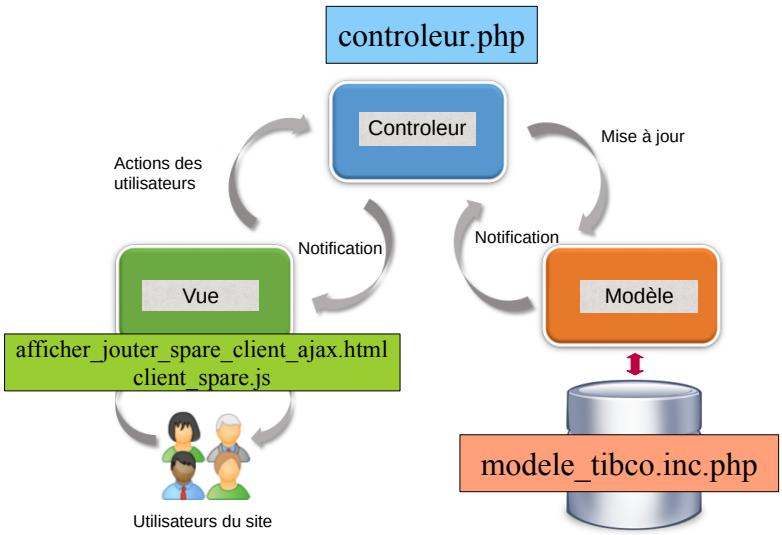
3 ARCHITECTURE CLIENT/SERVEUR

La base de données nommée **tibco_sparebox** possède les tables suivantes:



4 NOTION MVC

Afin de transformer notre maquette en prototype fonctionnel, nous allons structurer notre application selon le modèle MVC.



- La vue est chargée de l'affichage et de la mise en forme des informations cela correspondra à nos fichiers html, css et js
- Le contrôleur reçoit des demandes de la vue et fait appel au modèle pour mettre à jour ou demander des données. Cela correspondra à un fichier que nous appellerons **controleur.php.**
- Le modèle interagit avec la base de données et retourne des informations au contrôleur , le modèle correspondra au fichier **modele_tibco.inc.php**.

4.1 Mise en œuvre

1. ***Décompressez l'archive contenant le projet et verifiez que ce dernier est bien en mode "PHP Built-in Web Server".***
2. ***Testez l'exemple fourni (afficher_ajouter_spare_client_ajax.html) en modifiant le fichier config.inc.php afin de correspondre au serveur de base de données. (en cliquant sur ⊕ une modale apparait dans laquelle la liste des entreprises est remplie).***

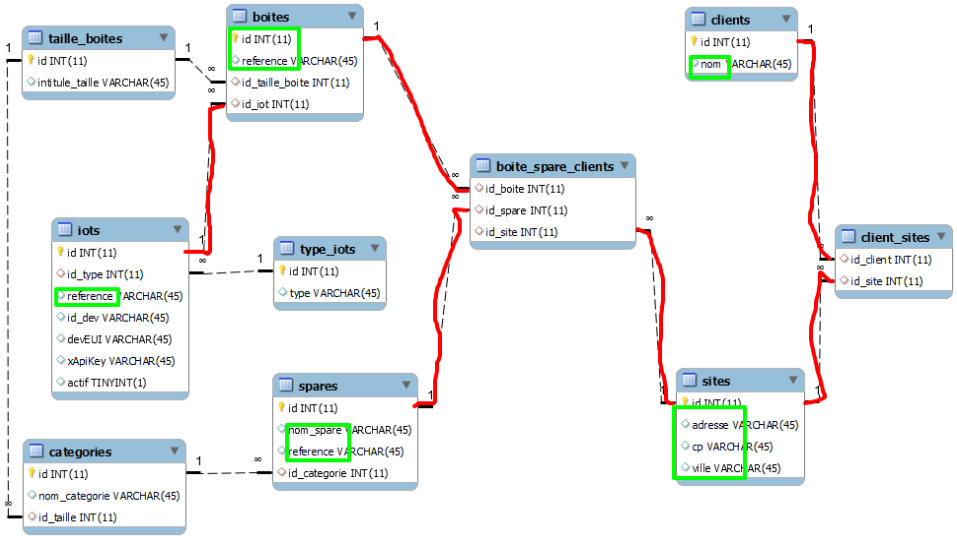
Pour information, La génération de la liste des clients se fait via le dialogue vue<->contrôleur suivant :

vue → contrôleur	contrôleur → vue
`{ "commande": "getListeClients" }`	`[ { "id": "1", "nom": "Auchan" }, { "id": "2", "nom": "Mecachrome" }, { "id": "3", "nom": "La panetiere" } ]`

4.2 Remplissage du tableau des boites.

La requête SQL permettant d'obtenir la référence des boites, la référence des iots, la référence des spares, le nom des spares, le nom des clients et l'adresse, le code postal et la ville des boites associées à des clients ayant une sparebox est fournie dans le code de la fonction ***getDataTableauSpareClient*** du fichier ***modele_tibco.inc.php***.

Sur le schéma ci-dessous les champs à récupérer sont encadrés en vert et les jointures à faire sont en rouge.



Pour la génération du tableau des boites associées à un site, on se propose d'avoir les informations suivantes qui transitent au format json entre la vue et le contrôleur :

vue → contrôleur	contrôleur → vue
{ "commande": "getBoitesSite" }	[{ "0": "B102", "1": "S001", "2": "DELL-INSP-003", "3": "Dell Inspiron 15", "4": "Auchan", "5": "Zac moulin aux moines 72000 Le Mans", "6": "", "DT_RowId": "2" }, { "0": "B001", "1": "L001", "2": "NT-M1-001", "3": "Lecteur code-barre NT-M1", "4": "Mecachrome", "5": "2 rue Saint-Exupéry 31140 Launaguet", "6": "", "DT_RowId": "1" }]

Numéro ligne	Champs
0	référence de la boite
1	référence de l'iot
2	référence du spare
3	nom du spare
4	nom du client
5	adresse, code postal et ville du site

3. ***Complétez la fonction remplirTableauBoiteClient du fichier client_spare.js***
4. ***Complétez controleur.php qui devra faire appel à la fonction getDataTableauSpareClient contenu dans modele_tibco.inc.php afin d'avoir le résultat attendu.***

4.3 Génération de la liste des boites dans la modale.

Pour générer la liste des boites disponibles on propose le dialogue vue<->contrôleur suivant;

vue → contrôleur	contrôleur → vue
{ "commande": "getListeBoites" }	[{ "id": "3", "boite": "B002(Lecteur code-barre NT-M1 NT-M1-003)" }, { "id": "4", "boite": "B103(Imprimante Canon TS6350 TS6350-002)" }, { "id": "5", "boite": "B104(Dell Inspiron 15 DELL-INSP-005)" }]

5. Donnez la requête SQL permettant d'obtenir l'id , la référence des boites ainsi que le nom et la référence du spare de ces boites, tel que id_site est NULL dans la table boite_spare_clients et id_iot de la table boites n'est pas à NULL.

Pour tester si un champ est null ou non null en SQL, la syntaxe est la suivante :

```
nomChamp IS NULL
nomChamp IS NOT NULL
```

6. Complétez la fonction remplirListeBoitesDisponibles du fichier client_spare.js

7. Complétez controleur.php et modele_tibco.inc.php afin d'avoir le résultat attendu, en codant et appelant une fonction genererListeBoiteJson.

4.4 Génération de la liste des sites

Pour générer la liste des sites on propose le dialogue vue<->contrôleur suivant;

vue → contrôleur	contrôleur → vue
{ "commande": "getListeSites", "id": "1" }	[{ "id": "1", "site": "Zac moulin aux moines 72000 Le Mans" }, { "id": "2", "site": "57 rue du Chateau d'eau 33000 Bordeaux" }, { "id": "3", "site": "127 boulevard Lobau 54000 Nancy" }]

8. ***Donnez la requête SQL permettant d'obtenir l'adresse, le code postal et la ville des sites pour le client ayant pour id 1 .***
9. ***Complétez la fonction remplirListeSite du fichier client_spare.js***
10. ***Complétez controleur.php et modele_tibco.inc.php afin d'avoir le résultat attendu, en codant et appelant une fonction genererListeSiteFromIdEntreprise qui prendra en paramètre l'id de l'entreprise dont il faut récupérer les adresses.***

4.5 Mise à jour

Pour la mise à jour des associations on propose le dialogue vue<->contrôleur suivant :

vue → contrôleur	contrôleur → vue
{ "commande": "setBoiteSite", "idBoite": "3", "idSite": "5" }	{ "status": "ok" }
{ "commande": "setBoiteSite", "idBoite": "-1", "idSite": "5" }	{ "status": "pasok" }

11. ***Coder la fonction majBoiteSite dans le fichier client_spare.js***
12. ***Complétez controleur.php et modele_tibco.inc.php afin d'avoir le résultat attendu.***