

VisueBaie

Pour vérification mise à jour de GLPI :

« glpi_networkequipment »

« glpi_networkports »

« glpi_computers »

« glpi_peripheral »

« glpi_phones »

« glpi_printers »

« glpi_users »

« glpi_groups »

« glpi_computers_items »

« glpi_sockets »

« glpi_networkportsethernets »

« glpi_networkequipmenttypes »

« glpi_networkports_networkports »

1. Choix des outils nécessaires pour répondre au besoin de l'application
2. Comment sont récupérées les informations depuis GLPI ?
3. Comment fonctionne l'application Laravel ?

1. Choix des outils nécessaires pour répondre au besoin de l'application

- Laravel ([Laravel - The PHP Framework For Web Artisans](#))

- Nous avons décidé d'utiliser le framework PHP Laravel pour une meilleure maintenabilité du code au fil du temps, il permet facilement de structurer (MVC) et de configurer notre code. Avec Laravel, les bibliothèques tierces sont très faciles d'installation et d'utilisation et permet une flexibilité intéressante.
- JQuery ([jQuery](#))
 - Bibliothèque utilisé pour pouvoir utiliser la bibliothèque « JQuery datatable »
- JQuery datatable ([DataTables | Table plug-in for jQuery](#))
 - Bibliothèque tierce utilisant JQuery permettant de générer un tableau entièrement interactif avec la possibilité de filtrer sur toutes les colonnes et de classer par ordre croissant ou décroissant chaque colonne, et cela de manière instantanée.
- Laravel-dompdf ([GitHub - A DOMPDF Wrapper for Laravel](#))
 - Bibliothèque tierce pour Laravel permettant de générer des PDF depuis des vues.

2. Comment sont récupérées les informations depuis GLPI ?

Nous avons décidé d'utiliser une vue SQL pour regrouper toutes les données nécessaires provenant de la base de données GLPI.

La vue nous permet de regrouper les informations dans une seule table ce qui évite de faire beaucoup de requêtes SQLs dans l'application et d'optimiser le code.

La vue est assez longue au niveau de la requête SQL mais est définitive et nécessite d'être invoquée une fois en boucle (on parcourt simplement les enregistrements dans la vue avec un foreach).

Explication du contenu :

Dans nos requêtes SQL, nous avons utilisé des alias pour les tables pour éviter de répéter le nom des tables à chaque fois ce qui est plus pratique et plus court (ex : glpi_networkequipments devient « ne » dans la requête)

Pour obtenir tous les ports réseaux de chaque switch, nous utilisons les tables « glpi_networkequipments », « glpi_networkports » et « glpi_networkequipmenttypes », puis plusieurs jointures entre toutes ces tables nous permet d'obtenir : Index du switch, nom du switch, les ports réseaux.

```

1 SELECT ne.id AS 'Index switch', ne.name AS 'Nom switch', np.id AS 'Index Port'
2 FROM glpi_networkequipments ne
3
4 INNER JOIN glpi_networkports np ON ne.id=np.items_id
5 LEFT JOIN glpi_networkequipmenttypes netype ON netype.id=ne.networkequipmenttypes_id
6
7 WHERE np.itemtype = 'NetworkEquipment' AND netype.name LIKE '%switch%';

```

Pour obtenir les ports réseaux qui sont connectés sur chaque port réseau de chaque switch, on utilise la table « glpi_networkports_networkports », la table « np2 » correspond à la table du port connecté au port du switch.

```

1 SELECT ne.id AS 'Index switch', ne.name AS 'Nom switch', np.id AS 'Index Port',
2 np2.id AS 'Index Port connecté'
3 FROM glpi_networkequipments ne
4 INNER JOIN glpi_networkports np ON ne.id=np.items_id
5 LEFT JOIN glpi_networkequipmenttypes netype ON
6 netype.id=ne.networkequipmenttypes_id
7 LEFT JOIN glpi_networkports_networkports npp ON np.id IN (npp.networkports_id_1,
8 npp.networkports_id_2)
9 LEFT JOIN glpi_networkports np2 ON np2.id IN (npp.networkports_id_1,
10 npp.networkports_id_2) AND np2.id != np.id
11
12 WHERE np.itemtype = 'NetworkEquipment' AND netype.name LIKE '%switch%';

```

Nom switch	Index Port	Index Port connecté
UNIT02	3	16
UNIT03	6	23
UNIT01	1	10
UNIT01	2	11
UNIT13	37	12
UNIT01	13	15
UNIT12	36	14
UNIT04	25	20
UNIT03	5	22
UNIT04	26	27
UNIT08	32	54
UNIT06	30	45
UNIT20	44	50
UNIT17	41	51
UNIT15	39	53
UNIT02	4	NULL
UNIT05	29	NULL
UNIT07	31	NULL
UNIT07	46	NULL
UNIT07	47	NULL

Ici, notre requête SQL fonctionne et récupère bien les bonnes données avec pour chaque port réseau du switch l'index du port qui est connecté à celui-ci (NULL si aucun port connecté).

Maintenant que l'on arrive à récupérer le port du matériel qui est connecté à un port d'un switch, alors l'enchaînement pour retrouver toutes les informations du matériel est assez facile (nom du matériel, l'utilisateur, etc...).

Pour récupérer l'ordinateur derrière un port réseau, on fait une jointure :

« LEFT JOIN glpi_computers c ON c.id = np2.items_id AND np2.itemtype = 'Computer' ». Pour récupérer un téléphone, un périphérique, une imprimante ou bien un matériel réseau (station, borne etc.) on ajoute encore plusieurs jointure avec la table qui correspond au matériel (ex : téléphone = glpi_phones.id, imprimante = glpi_printers.id) et la vérification du type du matériel pour effectuer cette jointure

Pour récupérer la prise associée, on utilise la table « glpi_sockets » et son attribut « networkports_id », nous avons l'id du port du matériel connecté, alors on vérifie via une jointure si glpi.sockets.networkports_id = np2.id. Cependant, le connecteur peut être placé via le matériel directement ou alors sur le switch, il faut donc prévoir ces 2 cas de figure.

Cette jointure permet de le faire :

« LEFT JOIN glpi_sockets s ON s.networkports_id IN (np.id, np2.id) »

Donc on peut très facilement avoir le nom de la prise avec « s.name ».

Pour récupérer un ordinateur s'il est derrière un téléphone, on utilise la table « glpi_computers_items », on regarde si le matériel connecté est un téléphone et si son « id » se retrouve dans la table glpi_computers_id, si oui alors ce téléphone possède un ordinateur.

« LEFT JOIN glpi_computers_items ci ON ci.items_id=np2.items_id AND np2.itemtype='Phone' AND ci.itemtype='Phone' »

Dans la création de la vue SQL, les noms des colonnes ont été modifiés pour plus de facilité dans l'application VisueBaie

« sw_id » → Index du switch

« sw_name » → Nom du switch

« name_rack » → Nom de la baie (côté switch)

« num_member » → Num membre (côté switch)

« sw_port_id » → Index du port du switch

« sw_port_name » → Nom du port du switch

« socket_name » → Nom de la prise (soit côté matériel, soit côté switch)

« socket_id » → Index de la prise

« mat_name » → Nom du matériel connecté
 « mat_type » → Type de matériel connecté
 « mat_id » → Index du matériel connecté
 « sw_port_status » → Status interne sur port réseau (pastille rouge ou verte ou noir)
 « internet_type » → Type de connexion du port réseau du matériel connecté
 « u_id » → Index de l'utilisateur associé au matériel connecté
 « u_name » → Nom de l'utilisateur associé au matériel connecté
 « gt_name » → Nom du groupe technique associé au matériel connecté
 « c_on_phone_pheripheral_id » → Index de l'ordinateur connecté au matériel connecté sur le switch directement
 « c_on_phone_pheripheral_name » → Nom de l'ordinateur connecté au matériel connecté sur le switch directement
 « c_last_boot » → Date du dernier démarrage de l'ordinateur connecté soit sur le switch soit sur un téléphone ou un dock.

Pour obtenir les résultats entrées dans les colonnes si-dessous :

« glpi_networkequipment » → Donne le nom du switch(sw_name), l'index du switch(sw_id) et le nom de la baie(name_rack)(associée avec glpi_networkports)

« glpi_networkports » → Donne le nom du port du switch(sw_port_name) et reconnaît le statut et donne le port du switch par son index(sw_port_id)
 Donne le nom des matériels(mat_name), le type de matériel(mat_type) et l'index du matériel(mat_id),
 Donne le statut du port réseau(sw_port_status), donne l'index de l'utilisateur associé au matériel en fonction de son type(u_id) et le nom de l'utilisateur associé(u_name),
 Donne le groupe technique associé au matériel grâce au type de matériel(gt_name)
 (associée avec glpi_networkports_networkports, glpi_networkequipment, glpi_sockets, glpi_networkportethernets, glpi_computers, glpi_computers_items, glpi_printers, glpi_peripherals, glpi_phones)

« glpi_computers » → Est la table où se trouve les données des ordinateurs avec les utilisateurs associées et la dernière mise en marche(c_last_boot)(associée avec glpi_networkports)

« glpi_peripheral » → Est la table où se trouve les données des matériels de type périphérique avec les utilisateurs associées(associée avec glpi_networkports)

« glpi_phones » → Est la table où se trouve les données des téléphones avec les utilisateurs et les matériels associées(associée avec glpi_networkports)

« glpi_printers » → Est la table où se trouve les données des imprimantes avec les utilisateurs et matériels associées(associée avec glpi_networkports)

« glpi_users » → Est la table où se trouve les données de tout les utilisateurs(associée avec glpi_computers, glpi_peripherals, glpi_phones, glpi_printers, glpi_networkequipments)

« glpi_groups » → Est la table où se trouve les données de tout les groupes techniques(associée avec glpi_computers, glpi_peripherals, glpi_phones, glpi_printers, glpi_networkequipments)

« glpi_computers_items » → Est la table où se trouve les données des matériels connectées à un ordinateur(c_on_phone_peripheral_name, c_on_phone_peripheral_id)(associée avec glpi_networkports, glpi_computers et glpi_peripheraltypes)

« glpi_sockets » → Est la table où se trouve les données des prises avec leur noms(socket_name) et leur index(socket_id).(associée avec glpi_networkports)

« glpi_networkportsethernets » → Donne le type de prise(internet_type)(est associée avec glpi_networkports)

« glpi_networkequipmenttypes » → Donne le type d'équipement(est associée avec glpi_networkequipments)

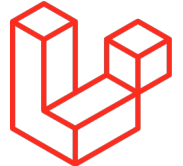
« glpi_networkports_networkports » → Donne les données sur deux ports réseaux connectés entre eux(associée avec glpi_networkports)

Résultat de la vue finale :

Serveur: localhost:3306 » Base de données: glpidb » Vue: glpi_visuebaie										
Parcourir Structure SQL Rechercher Insérer Exporter Opérations										
	sw_id	sw_name	name_rack	num_member	sw_port_id	sw_port_name	socket_name	socket_id	mat_name	mat_type
imer	2	UNIT02			3	1	LR2F25	13	UC694	Computer
imer	3	UNIT03	HDV_LR3	3	6	2	NULL	NULL	I5487	Printer
imer	1	UNIT01	HDV_LR1	1	1	1	NULL	NULL	STA003	Peripheral
imer	1	UNIT01	HDV_LR1	1	2	2	LR3F12	12	0361762056	Phone
imer	16	UNIT13			37	1	LR3C09	16	0361762060	Phone
imer	1	UNIT01	HDV_LR1	1	13	3	LR3F10	10	UC730	Computer
imer	6	UNIT04			25	1	LR47864	15	UC731	Computer
imer	3	UNIT03	HDV_LR3	3	5	1	L3F78	14	I4587	Printer
imer	6	UNIT04			26	2	LR3F09	5	B1024	NetworkEquipmen
imer	9	UNIT06			30	1	NULL	NULL	STA001	Peripheral
imer	2	UNIT02			4	2	LR3C01	1	NULL	NULL
imer	8	UNIT05			29	1	NULL	NULL	NULL	NULL
imer	10	UNIT07			31	1	NULL	NULL	NULL	NULL
imer	10	UNIT07			46	2	NULL	NULL	NULL	NULL
imer	10	UNIT07			47	3	NULL	NULL	NULL	NULL
imer	10	UNIT07			48	4	NULL	NULL	NULL	NULL
imer	11	UNIT08			32	1	NULL	NULL	NULL	NULL

3. Comment fonctionne l'application Laravel ?

Laravel est un framework en PHP basé sur la programmation orienté objet (POO), il fonctionne avec l'architecture MVC (Modèle, vue, contrôleur), c'est une façon d'organiser notre application web plutôt que de tout mélanger (html, css, requête à la bdd etc..).



Modèle : Classe pour interroger notre base de données, il représente en quelque sorte une table BDD sur laquelle on fait des actions (ex : VueBaie est un modèle qui fait référence à notre table/vue « glpi_visuebaie »). Grâce à ce modèle on peut : sélectionner, modifier, supprimer sur la table associée au modèle.

Vue : Correspond à ce que va finalement voir l'utilisateur sur son navigateur, c'est dans une vue que l'on va mettre le HTML, afficher des données récupérées depuis le contrôleur.

Contrôleur : Zone de « traitement », c'est ici que l'on va utiliser le modèle et renvoyer une vue. (ex : VueBaieController avec la méthode « index » qui récupère tous les enregistrements via le modèle « VueBaie » et qui renvoie la vue « home » (vue par défaut qui affiche notre tableau dans VisueBaie).

Fonctionnement de VisueBaie sur Laravel :

Pour obtenir le tableau de VisueBaie avec Laravel, on crée une route dans le fichier « routes/web.php », cette route appelle notre contrôleur qui va pouvoir gérer tout notre « traitement ».

```
Route::get('/', [VueBaieController::class, 'index']);|
```

Dans notre contrôleur « VueBaieController » dans le dossier « app/Http/Controllers/VueBaieController.php »

```
app > Http > Controllers > VueBaieController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\VueBaie;
6  use Illuminate\Http\Request;
7
8  class VueBaieController extends Controller
9  {
10     public function index()
11     {
12         // On récupère les données via le modèle VueBaie
13         $baies = VueBaie::all();
14         // On retourne la vue "home" en "injectant" notre variable $baies dans notre vue pour l'exploiter.
15         return view('home', compact('baies'));
16     }
17 }
```

La ligne 13 permet de récupérer dans un tableau associatif tous nos enregistrements concernant notre vue « glpi_visuebaie » avec notre modèle « VueBaie ».

La ligne 15 permet de renvoyer une vue « home » avec notre tableau que l'on injecte dans notre vue.

Maintenant, on accède à notre vue (du HTML pour faire simple) dans le dossier « resources/views/home.blade.php » **Nom de notre vue.**


```

resources > views > home.blade.php
98 </tr>
99 </thead>
100 <tbody>
101     @foreach($baies as $baie)
102         @php($status['direct'] = !is_null($baie['mat_type']) && is_null($baie['socket_id']))
103         <tr class="{{ $status['direct'] ? 'blue' : '' }}">
104             <td><a target="_blank" href="{{ env('GLPI_URL') }}/front/networkequipment.form.php?id={{ $baie['sw_id'] }}">{{ $baie['sw_name'] }}</a></td>
105             <td>{{ $baie['name_rack'] }}</td>
106             <td>{{ $baie['num_member'] }}</td>
107             <td><a target="_blank" href="{{ env('GLPI_URL') }}/front/networkport.form.php?id={{ $baie['sw_port_id'] }}">{{ $baie['sw_port_name'] }}</a></td>
108             <td><span class="port_status {{ $status['direct'] ? 'blue' : (is_null($baie['mat_type']) ? 'gray' : 'green') }}">{{ $status['direct'] ? 'Direct' : 'Indirect' }}</span></td>
109             <td>
110                 @if(!is_null($baie['socket_id']))
111                     <a target="_blank" href="{{ env('GLPI_URL') }}/front/socket.form.php?id={{ $baie['socket_id'] }}">{{ $baie['socket_name'] }}</a>
112                 @endif
113             </td>
114             <td>{{ $baie['gt_name'] }}</td>
115             <td>
116                 @if(!is_null($baie['mat_id']))
117                     <a target="_blank" href="{{ env('GLPI_URL') }}/front/{{ strtolower($baie['mat_type']) }.form.php?id={{ $baie['mat_id'] }}">{{ $baie['mat_name'] }}</a>
118                 @endif
119             </td>
120             <td>
121                 @if(!is_null($baie['c_on_phone_peripheral_id']))
122                     <a target="_blank" href="{{ env('GLPI_URL') }}/front/computer.form.php?id={{ $baie['c_on_phone_peripheral_id'] }}">{{ $baie['c_on_phone_peripheral_name'] }}</a>
123                 @endif
124             </td>
125             <td>
126                 @if(!is_null($baie['u_id']))
127                     <a target="_blank" href="{{ env('GLPI_URL') }}/front/user.form.php?id={{ $baie['u_id'] }}">{{ $baie['u_name'] }}</a>
128                 @endif
129             </td>
130             <td>{{ !is_null($baie['c_last_boot']) ? \Carbon\Carbon::parse($baie['c_last_boot'])->format('d/m/Y à H\hi') : '' }}</td>
131         </tr>
132     @endforeach
133 </tbody>
134 </table>

```

Il y a du CSS et du javascript (le javascript va nous permettre d'avoir un tableau interactif) et bien sûr le HTML. Au niveau du HTML nous avons un simple tableau « <table> » pour afficher nos enregistrements de notre vue (qu'on a récupéré dans la variable \$baies) nous allons parcourir cette variable avec un foreach. (le @ est spécifique à Blade(moteur de template qu'utilise Laravel))

Donc pour chaque enregistrement de \$baies nous avons un switch avec son port réseau et toutes ses informations (connecté ou non, status, groupe tech, matériel etc...) Pour chaque colonne <td>, on affiche donc les bons éléments par exemple pour obtenir le nom du switch on utilise \$baie['sw_name'] (sw_name est le nom de la colonne dans notre vue SQL), pour obtenir le matériel connecté on utilise \$baie['mat_name'] etc.

Utilisation d'un système de cache :

Le système de cache va nous permettre d'améliorer les performances de façon considérables. Il permet d'éviter à chaque chargement de la page d'aller récupérer toutes les données depuis notre vue SQL ce qui est très lent (environ 10s). Le cache récupère les données et les stocke dans un fichier, en récupérant le cache depuis ce fichier, le temps de chargement devient performant et rapide (environ 530ms plutôt que 10s sans le cache).

Avec Laravel, un système de cache est facile à mettre en place avec la classe « Cache »

```

16 //On récupère les données via le modèle VueBaie
17 $baies = Cache::rememberForever('cache_visuebaies', function () {
18     return VueBaie::all();
19 });


```

Dans notre contrôleur, nous mettons en place un cache où il ne sera jamais « vidé » automatiquement mais seulement avec le bouton dédié à cet effet.



CSV

PDF



Nom actif réseau	Nom Baie	Num membre	Port
UNIT07	HDV_LR1	Mbre 1	01

Précisions :

- Pour afficher une variable dans une vue avec Blade, on utilise {{ \$variable }}
- On peut directement changer l'url du GLPI que l'on souhaite pour les hyperliens via le fichier .env à la racine et GLPI_URL=<url du glpi>

```
6 GLPI_URL=https://glpi.ville-armentieres.fr
```

- Pour changer les informations de connexion à la BDD, tout est dans le .env à la racine avec DB_HOST, DB_USERNAME etc.

```

12 DB_CONNECTION=mysql
13 DB_HOST=10.1.1.50
14 DB_PORT=3306
15 DB_DATABASE=glpi3
16 DB_USERNAME=LireVueBrassage
17 DB_PASSWORD=

```

- Le cache doit être obligatoirement vidé en cas de modification sur GLPI car le site VisueBaie est basé sur les données du cache.
- Le nom de la vue SQL est spécifié dans le modèle « VueBaie », qui est présent dans le répertoire : « app\Models\VueBaie.php ».

```

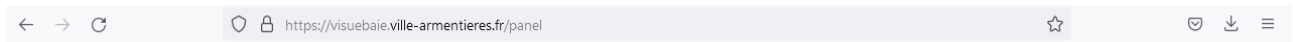
8 class VueBaie extends Model
9 {
10     use HasFactory;
11
12     /**
13      * Précision du nom de la vue dans la base de données
14      */
15     protected $table = 'VueVisueBaie';
16 }

```

- Le système de recherche fonctionne sur toutes les colonnes pour pouvoir rechercher sur plusieurs colonnes, il faut espacer les mots clés par exemple « spécial système », cela va rechercher tous les matériels connectés en spécial qui appartiennent au groupe technique « SYSTEME »

Création d'utilisateurs :

Pour pouvoir accéder au back-office de VisueBaie, on rajoute « /panel » derrière le lien de l'application.



La création d'utilisateurs est faite par les administrateurs de VisueBaie.

A screenshot of the VisueBaie web application's user management interface. The browser address bar shows "10.1.1.51/panel/edit/1". The page header includes the VisueBaie logo and navigation links: "Utilisateurs", "Créer un utilisateur", and "Déconnexion". The main content area is titled "Edition d'un utilisateur admin". It contains a form with the following fields: "Nom d'utilisateur" (a text input field), "Mot de passe" (a text input field with a "Générer un mot de passe" link next to it), and a section "Autoriser l'utilisateur à:" with four checked checkboxes: "Consulter les imprimantes", "Consulter les ordinateurs", "Consulter les utilisateurs", and "Consulter les téléphones". At the bottom of the form is a green button labeled "Modifier l'utilisateur".

Les utilisateurs sont créés avec des droits prédéfinis par les administrateurs, On peut générer un mot de passe aléatoire basé sur huit caractères avec des lettres, des chiffres et des caractères spéciaux.

The screenshot shows a web browser window with the URL `10.1.1.51/panel/create-user`. The page header includes the VisueBaie logo and navigation links: `Utilisateurs`, `Créer un utilisateur`, and `Déconnexion`. The main heading is **Création d'un utilisateur**. The form contains the following fields and options:

- Nom d'utilisateur:** A text input field containing the value `ervan`.
- Mot de passe:** A text input field containing the value `vYVOHfx`. A link `Générer un mot de passe` is visible to the right.
- Autoriser l'utilisateur à:** A section with three checkboxes:
 - ☐ Consulter les imprimantes
 - ☐ Consulter les ordinateurs
 - ☒ Consulter les téléphones

A green button labeled **Créer l'utilisateur** is positioned at the bottom of the form.

At the bottom of the browser window, the Windows taskbar is visible, showing the system clock as 15:42 on 12/02/2014.

Le nom d'utilisateur doit contenir trois caractères et un mot de passe de sept caractères avec obligatoirement des lettres et des chiffres.

This screenshot shows the same user creation page as above, but with validation errors. The `Nom d'utilisateur` field now contains a single character `|`, and the `Mot de passe` field is empty. Red error messages are displayed below each field:

- Below the username field: `The name field must be at least 3 characters.`
- Below the password field: `The password field must be at least 7 characters.`

The other elements, including the navigation links, the **Création d'un utilisateur** heading, the permissions checkboxes (all of which are unchecked), and the **Créer l'utilisateur** button, remain the same. The Windows taskbar at the bottom shows the same date and time (15:42 on 12/02/2014).

On reçoit une notification que le compte a bien été créé avec les droits attribués précédemment.

10.1.1.51/panel/create-user

VisueBaie Utilisateurs [Créer un utilisateur](#) | Déconnexion

Création d'un utilisateur

L'utilisateur enwan a été créé avec succès !

Nom d'utilisateur

Mot de passe [Générer un mot de passe](#)

Autoriser l'utilisateur à :

- ☐ Consulter les imprimantes
- ☐ Consulter les ordinateurs
- ☐ Consulter les utilisateurs
- ☐ Consulter les téléphones

Créer l'utilisateur

15:45 12/02/2024

Recherche multiple :

On a l'option de chercher plusieurs colonnes si on a envie d'être précis comme un ordinateur et un téléphone pour une baie, etc...

CSV PDF

Rechercher UNIT05 UC857

Nom actif réseau	Nom Baie	Num membre	Port	Status	Prise	Groupe Technique	Matériel connecté	PC connecté sur matériel	Utilisateur	Dernière connexion
<u>UNIT05</u>	HDV_LR2	Mbre 2	28	Active	LR2C14	RP	0361762062	<u>UC857</u>	mbouquet	15/02/2024 à 07h44

De 1 à 1 sur 1 lignes (filtré sur 3,387 enregistrements au total)

Cette exemple au dessus démontre qu'en recherchant « UNIT05 UC857 » marche pour trouver précisément nos données.

On est obligé de séparer la donnée mise en premier et la seconde(UNIT05 et UC857 par exemple) par un espace.