



DEVELOPPEMENT MODULE ET INTEGRATION PRESTASHOP



GEORGES REMI

SOMMAIRE

1. Expression des besoins	
1.1 Contexte, domaine, processus métier	2
1.2 Les différents acteurs impliqués dans le projet	2
1.3 Etude de l'existant, diagnostic	2
1.4 Description de la demande, objectifs, bénéfices attendus	2
1.5 Contraintes ou exigences (matérielles, techniques, délais, budget, ...)	3
2. Conception, Spécifications techniques	
2.1 Description de la solution	3
2.2 Outils logiciels de la solution	6
2.3 Architecture matérielle et logicielle de la solution.....	6
2.4 Besoins techniques, ressources.....	7
3. Développement	
3.1 Dossier de programmation codes sources documentés et commenté	7
3.2 Difficultés rencontrées	16
4. bilan	17

1 Expression des besoins

1.1 Contexte, domaine, processus métier :

Fondée en 2009 par une équipe diversifiée de professionnels passionnés, Ether Création se distingue par sa composition hétérogène, réunissant des experts en informatique, des spécialistes du marketing, et des consultants en stratégie. L'entreprise a été créée à l'initiative de Arthur Revenaz qui a apporté son expertise complémentaire pour développer une agence spécialisée dans la création de modules personnalisés et la transformation numérique.

Positionnée dans le secteur du e-commerce, Ether Création offre une gamme de services allant de la création de modules sur mesure à la refonte technique et graphique, en passant par des conseils stratégiques en marketing et en fonctionnalités. L'entreprise se distingue également par sa spécialisation dans PrestaShop (front et back office), avec une particularité étant la connexion avec les ERP, CRM et PIM.

Sur la plateforme add-ons.prestashop.com, Ether Création propose plus de 50 modules, démontrant ainsi son engagement continu envers l'écosystème Prestashop. La qualité du support client est une priorité, avec un taux de satisfaction dépassant les 95%.

Avec trois agences situées à Angers (4 salariés), Tours (1 salarié) et Bourg-en-Bresse (3 salariés), Ether Création compte au total 8 salariés.

1.2 Les différents acteurs impliqués dans le projet sont les suivants :

Je suis l'acteur principal dans le projet qui consiste à développer le plus loin et le plus poussé possible un site d'e-commerce. Je serai épaulé de toute l'équipe en cas de besoin mais sur tout de Arthur Leroux qui a une solide expérience dans le développement.

Les utilisateurs sont les internautes qui visiteront le site web. Ils peuvent être des clients potentiels, mon objectif en tant que développeur est de fournir une expérience optimale pour les utilisateurs

1.3 Etude de l'existant, diagnostic :

Il n'existe pour le moment pas de site e-commerce pour le demandeur.

1.4 Description de la demande, objectifs, bénéfices attendus :

La demande de mon entreprise consiste à développer un site d'e-commerce portant sur la vente de joaillerie pour un client.

L'objectif de cette demande est économique, permettrai au demandeur de générer un chiffre.

1.5 Contraintes ou exigences (matérielles, techniques, délais, budget, ...) :

J'utilise Postman, pour tester l'API du fournisseur au préalable.

Je fais ces tests pour savoir comment manipuler une API et les requêtes de celle-ci.

Notre fournisseur est un grossiste

Ensuite, nous travaillons avec un éditeur de code (Visual Studio Code) pour développer et tester nos modules PHP.

Bien que les délais ne soient pas stricts, nous disposons de 6 semaines de stage pour approfondir et pousser le développement du produit demandé.

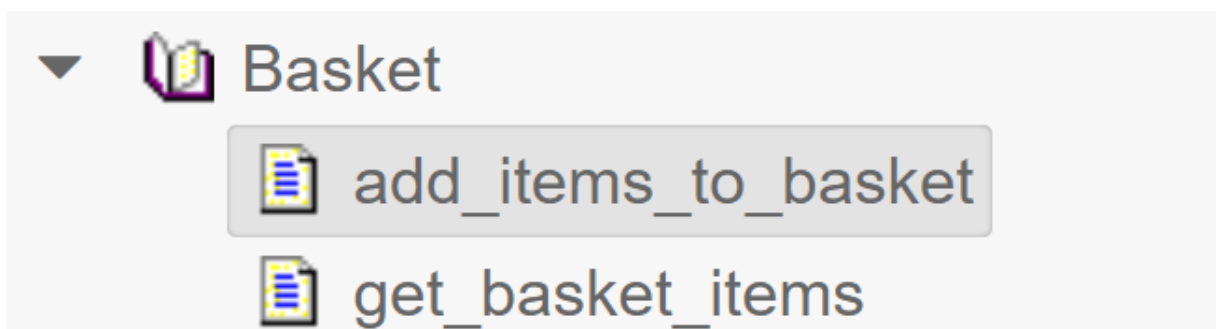
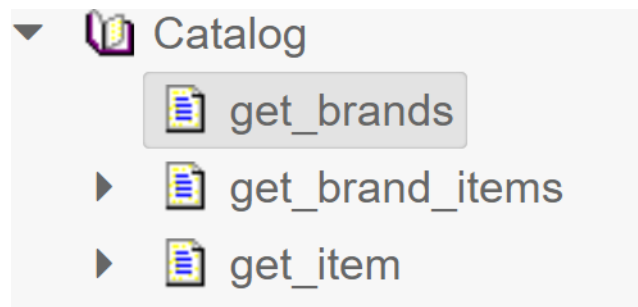
2 Conception, Spécifications Techniques

2.1 Description de la solution :

La première étape de la solution implique de tester notre API sur Postman ou en local via WampServer pour avoir un meilleur affichage graphique. Voir les tests à partir de la page 10.

Notre API possède diverse requête permettant le bon fonctionnement d'un site d'e-commerce. Nous avons accès à toutes ces requêtes car le fournisseur donne un guide de son API.

Nous travaillons actuellement avec les requêtes suivantes :



get_brands permet l'accès aux marques du catalogue avec les paramètres suivant :

Parameter	Required
uid	yes
pid	yes
lid	yes
key	yes
api_version	yes
request	yes

Get_brand_items permet d'afficher tous les produits d'une catégorie de marque spécifique avec les paramètres suivant :

Parameter	Required
uid	yes
pid	yes
lid	yes
key	yes
api_version	yes
request	yes
id_brand	yes

Et get_item permet d'afficher les détails d'un produit avec les paramètres suivant :

Parameter	Required
uid	yes
pid	yes
lid	yes
key	yes
api_version	yes
request	yes
id_product	yes

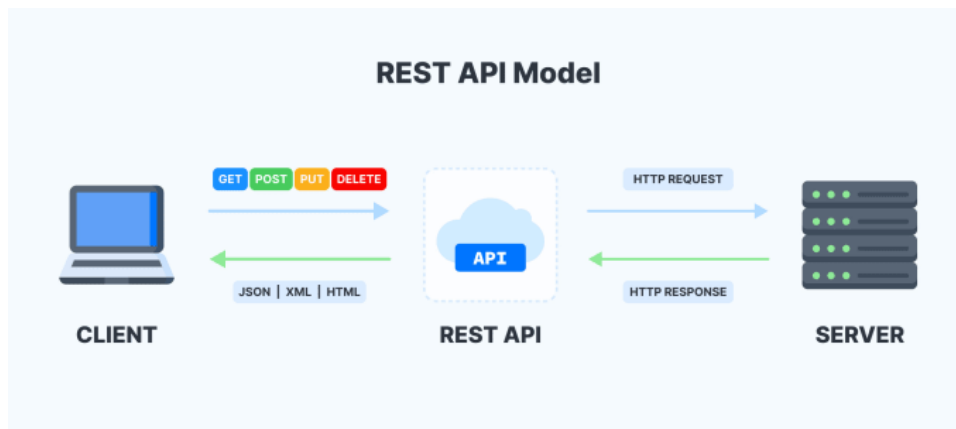
add_items_to_basket permet d'ajouter des produits et des détails client dans le panier API.

Parameter	Required
uid	yes
pid	yes
lid	yes
key	yes
api_version	yes
request	yes
items	yes
id_product	yes
qty	yes

get_basket_items permet d'afficher les éléments du panier API.




Parameter	Required
uid	yes
pid	yes
lid	yes
key	yes
api_version	yes
request	yes
id_basket	yes

Schéma de l'utilisation d'une API.



Définition API : Une API (application programming interface ou « interface de programmation d'application ») est une interface logicielle qui permet de « connecter » un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités.

2.2 Outils logiciels de la solution :

	Nous permet de tester notre API, crée des collections de requête.
	Visual studio code comme outil de développement.
	Pour crée un server en local

2.3 Architecture matérielle et logicielle de la solution :

L'architecture de la solution du développement du site d'e-commerce comprend les éléments matériels et logiciels suivants :

- Matériel : Un ordinateur ou un serveur local sera utilisé pour effectuer le développement du site
- Logicielle : Postman, Visual studio code, WampServer.

2.4 Besoins techniques, ressources (humaines, matérielles, logicielles et budgétaires, coûts) :

Besoins techniques :

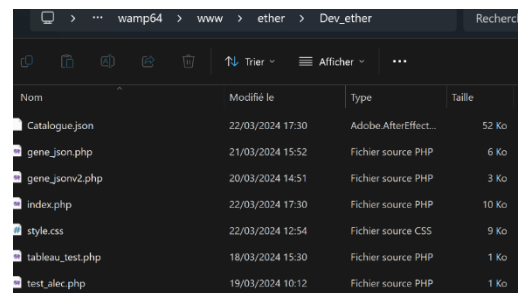
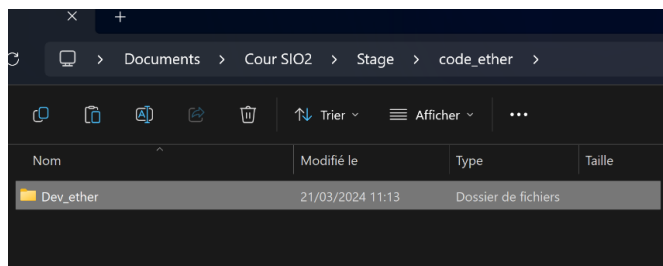
Compétences en développement web : Il est essentiel d'avoir des connaissances en développement web, particulièrement dans l'utilisation de PHP.

- Développeur web : Un développeur expérimenté sera responsable du développement de ce site.
- Environnement de développement local : L'utilisation de Wamp64 ou d'un environnement similaire sera nécessaire pour exécuter Apache, MySQL et PHP localement.


3 Développement





3.1 Dossier de programmation codes sources documentés et commentés

3.1.1 Racine du code



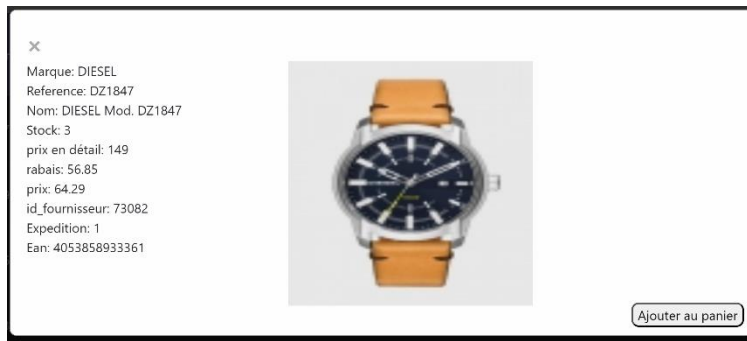
3.1.2 Affichage de notre catalogue

Rechercher un produit 

Marque	Reference	Nom	Stock	prix en détail	rabais	prix	id_fournisseur	Expedition	Ean	Images
DIESEL	DZ1847	DIESEL Mod. DZ1847	3	149	56.85	64.29	73082	1	4053858933361	
DIESEL	DZ4529	DIESEL Mod. GRIFFED	6	289	60.71	113.56	6058	2	4053858415430	
DIESEL	DZ4596	DIESEL Mod. GRIFFED	29	289	60.71	113.56	6058	2	4064092127430	
DIESEL	DZ1763	DIESEL Mod. RASP	13	179	60.48	70.75	6058	2	4053858696549	

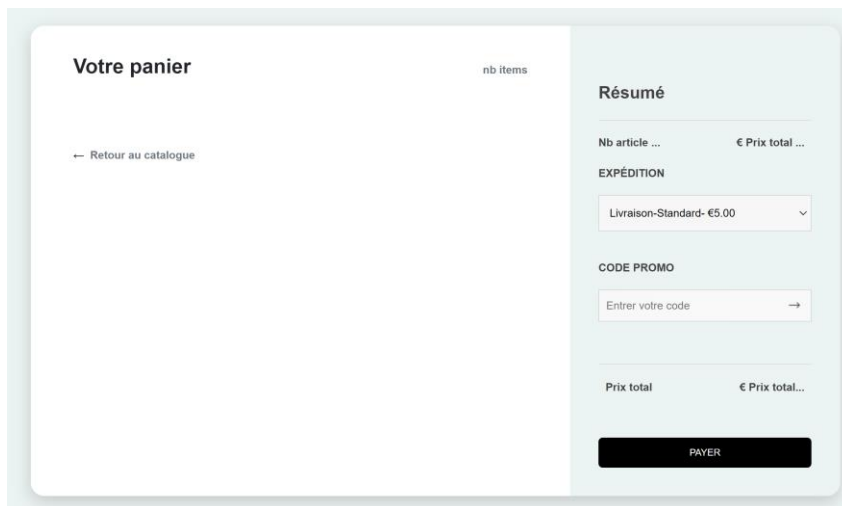
On a un catalogue de montres contenant toutes ses informations, on a input text qui va nous permettre de filtrer le tableau après avoir renseigné le détail d'un produit ou d'une marque. De plus on peut filtrer par ordre croissant ou décroissant le tableau en cliquant sur l'en-tête du tableau.

Après clic sur une ligne de notre tableau il y a un modal qui s'affiche



Celui-ci nous affiche les détails du produit, par la suite nous allons créer un carrousel pour afficher toutes les photos du produit et aussi l'ajout du produit dans un panier.

3.1.3 Affichage de notre panier (pas encore finalise)



3.1.4 Fonctionnalité de notre catalogue

Notre filtre fonctionne grâce à du JQuery, notre code fait en sorte qu'il supprime chaque ligne du tableau si les caractères renseignés dans l'input ne correspondent pas.

```
<script>
$(document).ready(function() {
  $("#myInput").on("keyup", function() {
    var value = $(this).val().toLowerCase();
    $("#myTable tr").filter(function() {
      $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)
    });
  });
});
</script>
```

Le système sur les en-têtes fonctionne grâce à un TableSorter.

Il s'agit d'un plugin JQuery permettant de trier un tableau HTML par ordre croissant et décroissant après clic sur l'en-tête.

```
<script>
$(function() {
  $(".tablesorter").tablesorter();
});
</script>
```

On a simplement besoin d'insérer ce code et d'appeler le plugin dans le Head de notre code.

Ce JQuery me permet de créer mon modal, il va renseigner le nom des en-têtes avec la valeur de la ligne sélectionnée, de plus on a retiré l'en-tête image pour afficher seulement sa valeur et par la suite créer notre carrousel. Voir page 6 le visuel du modal.

```
$(document).ready(function() {
  $("#myTable tr").click(function() {
    var rowInfo = '';
    $(this).find('td').each(function(i) {
      var header = $("#myTables th").eq(i).text(); // Récupère l'en-tête correspondant
      var cellData;
      if ($(this).find('img').length) { // regarde si la cellule contient une img
        var imgSrc = $(this).find('img').attr('src'); // récupère le src de l'image
        cellData = ''; // Crée l'image dans le modal
      } else {
        cellData = $(this).text();
      }
      if (header !== 'Images') { // Ajout de cette condition pour exclure l'en-tête 'Images'
        rowInfo += header + ': ' + cellData + '<br />'; // Ajoute l'en-tête devant la donnée
      } else {
        rowInfo += cellData + '<br />'; // Ajoute seulement la donnée de l'image
      }
    });

    // Contenu du modal
    var modalContent = "<div id='myModal' class='modal'><div class='modal-content'><span class='close'></span><p style='display: flex;'>"
    + "\r\n" + rowInfo + "</p><button class='btn_modal' id='btnpanier'>Ajouter au panier</button></div></div>";

    $("body").append(modalContent); // Ajout du modal à la page

    var modal = document.getElementById("myModal"); // Affichage du modal
    modal.style.display = "block";

    $(".close").click(function() { // Fermeture du modal après clic
      modal.style.display = "none";
      $("#myModal").remove();
    });
  });
});
```

3.1.5 Test des requêtes de notre API

Pour avoir pu en arriver ici, j'ai du tout d'abord m'entraîner à manipuler une API.

Tests réaliser :

On crée un tableau avec les paramètres obligatoire de notre méthode « get_brands »

Et on encode le résultat en Json, on récupère ce code Json pour l'insérer dans Postman (dans la partie Value), pour nous permettre d'afficher le résultat attendu. Ce code nous permet d'afficher toutes les marques.

→ Résultat ci-dessous de la requête sur Postman

```
$test = [  
  "uid"      =>"{{uid}}",  
  "pid"      =>"{{pid}}",  
  "lid"      =>"{{lid}}",  
  "key"      =>"{{key}}",  
  "api_version" =>"1.1.0",  
  "request"  =>"get_brands",  
];
```

```
echo json_encode($test);
```

```
{  
  "success": true,  
  "rc": 0,  
  "message": "",  
  "num_rows": 1290,  
  "rows": [  
    {  
      "id_brand": "45",  
      "group": "MONTRES",  
      "category": "MONTRES EN STOCK",  
      "name": "DIESEL",  
      "min_qty": "0",  
      "extra_discount_from_euro": 0,  
      "extra_discount_percentage_value": 0  
    },  
    ...  
  ]  
}
```

On refait le même système mais avec la requête get_brand_items. Ce code nous permet d'afficher tous les items avec l'id de la marque N°45.

→ Partie du résultat ci-dessous de la requête sur Postman

```
$test2 = [  
  "uid"      =>"{{uid}}",  
  "pid"      =>"{{pid}}",  
  "lid"      =>"{{lid}}",  
  "key"      =>"{{key}}",  
  "api_version" =>"1.1.0",  
  "request"  =>"get_brand_items",  
  "id_brand"  =>"45",  
];
```

```
echo json_encode($test2);
```

```
{  
  "success": true,  
  "rc": 0,  
  "message": "",  
  "num_rows": 62,  
  "rows": [  
    {  
      "id_product": "569469",  
      "brand": "DIESEL",  
      "reference": "DZ1847",  
      "brand_name": "DIESEL",  
      "name": "DIESEL Mod. DZ1847",  
      "stock": "3",  
      "weight": "300",  
      "retail_price": 149,  
      "price": 64.29  
    },  
    ...  
  ]  
}
```

→ Partie du résultat ci-dessous de la requête sur Postman

```
$test3 = [  
  "uid"      =>"{{uid}}",  
  "pid"      =>"{{pid}}",  
  "lid"      =>"{{lid}}",  
  "key"      =>"{{key}}",  
  "api_version" =>"1.1.0",  
  "request"  =>"add_items_to_basket",  
  "items"    =>[  
    "id_product" =>"569469",  
    "qty"        =>"1",  
  ],  
];  
  
echo json_encode($test3);
```

La requête n'a pas pu fonctionner car je n'ai pas les droits. Elle m'aura permis l'ajout de produit au panier.

```
"success": false,  
"rc": -11,  
"message": "One or more errors occurred while processing your request",  
"items": [  
  {  
    "success": false,  
    "rc": -6,  
    "message": "Field id_product is mandatory",  
    "id_basket": 0,  
    "id_product": 0,  
    "qty_requested": 0,  
    "qty_available": 0,  
    "minimum_qty": 0,  
    "qty_reserved": 0,  
    "retail_price": 0,  
    "discount": 0,  
    "price": 0,  
    "extra_discount": 0,  
    "net_price": 0  
  },  
]
```

3.1.6 Exploitation de notre API

On nous demande de créer une fonction PHP permettant d'afficher tous les produits de toutes les marques en passant par la requête `get_brand_items`.

A l'aide du code CURL que Postman nous fournisse, nous allons appeler l'API dans VS Code directement et par la suite afficher nos résultats dans notre serveur local.

Voici ci-dessous une partie de notre fonction CURL :

```
function curl($accespoint, $POST = null)
{
    $url = "https://dev.dropshippingb2b.com/api/";
    $table = [
        "uid"      => "82870",
        "pid"      => "9",
        "lid"      => "4",
        "key"      => "0yiBBiY95P44A9Sbbbq2w59RwC12e0mxvXu4B6NOArzZpA8I15E96v9Fcj7KPh7",
        "api_version" => "1.1.0",
        "request"   => $accespoint,
        "display_brand_name" => true,
        "display_reference" => true,
        "display_name"    => true,
        "display_stock"   => true,
        "display_weight"  => true,
        "display_retail_price" => true, //Tableau contenant les parametres de nos requêtes
        "display_discount" => true,
        "display_price"    => true,
        "display_speed_shipping" => true,
        "display_ean"      => true,
        "display_image_path" => true,
        "display_attributes" => true
    ];
    if (!is_null($POST)) { //Si $POST n'est pas nulle, ajout d'un élément au tableau $table
        $table["id_brand"] = $POST; //La clé "id_brand" est ajoutée au tableau $table avec la valeur de $POST
    }
    curl_setopt_array($curl, array(
        CURLOPT_URL => "https://dev.dropshippingb2b.com/api/", //Appel de notre API

        CURLOPT_HEADER => false,
        CURLOPT_VERBOSE => false,
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_ENCODING => "",
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_TIMEOUT => 0,
        CURLOPT_POST => true,
        CURLOPT_SSL_VERIFYPEER => false,
        CURLOPT_FOLLOWLOCATION => true,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => 'POST',
        CURLOPT_POSTFIELDS => $data,
    ));

    $response = curl_exec($curl);
    curl_close($curl);

    return $response;
}
```

Pour pouvoir réaliser cette demande nous avons besoin tout d'abord de récupérer chaque id_brand :

Affichage dans Postman après avoir utilisé notre requête get_brands.

```
"success": true,
"ic": 0,
"message": "",
"num_rows": 1290,
"rows": [
  {
    "id_brand": "45",
    "group": "MONTRES",
    "category": "MONTRES EN STOCK",
    "name": "DIESEL",
    "min_qty": "0",
    "extra_discount_from_euro": 0,
    "extra_discount_percentage_value": 0
  },
  {
    "id_brand": "65",
    "group": "MONTRES",
    "category": "MONTRES EN STOCK",
    "name": "SEIKO",
    "min_qty": "0",
    "extra_discount_from_euro": 0,
    "extra_discount_percentage_value": 0
  }
]
```

```
$response = curl('get_brands');

$data = json_decode($response, true);

$tid_brand = [];

if (isset($data['rows'])) { //Vérifie si la cle rows existe dans le tableau $data
    foreach ($data['rows'] as $brand) { //Parcour de chaque row
        $id_brand = $brand['id_brand']; //Récupération de l'id de la marque
        $tid_brand[] = $id_brand; //Insertion de chaque id dans le tableau $tid_brand
    }
} else {
    echo "Aucune marque trouvée";
}

var_dump($tid_brand);
```

Dans ce code, nous parcourons le tableau rows qui contient tout les id_brand, dès que notre foreach atteint l'index « id_brand » il l'ajoute dans notre tableau préalablement crée, puis on affiche le tableau avec un var_dump.

Après l'exécution de notre code, nous pouvons confirmer que les id_brand sont bien dans notre tableau.

```
[Running] php "c:\wamp64\www\ether\D
C:\wamp64\www\ether\D
array(1290) {
  [0] =>
  string(2) "45"
  [1] =>
  string(2) "65"
  [2] =>
  string(3) "101"
  [3] =>
  string(3) "144"
  [4] =>
  string(3) "153"
  [5] =>
```

On test le tableau contenant les id_brand avec le code suivant :

```
$temp = [];  
  
$count = 0;  
  
foreach ($tid_brand as $testid) {  
    if ($count > 5) {          // affichage de 5 marques pour éviter que le code mette trop de temps à s'exécuté  
        break;  
    }  
    $count++;  
  
    $reponse2 = curl('get_brand_items', $testid); //appel de la requete  
    $data2 = json_decode($reponse2, true); //traite les données  
    $temp[] = $data2; //stock les données dans un tableau  
}  
var_dump($temp);
```

Ce code nous permet de parcourir notre tableau contenant les id_brand, on appelle la requête get_brand_items grâce au CURL, et affiche les données de chaque id_brand appelé.

```
[Running] php "c:\wamp64\www\ether\De  
C:\wamp64\www\ether\Dev_ether\tempCod  
array(6) {  
  [0] =>  
    array(5) {  
      'success' =>  
        bool(true)  
      'rc' =>  
        int(0)  
      'message' =>  
        string(0) ""  
      'num_rows' =>  
        int(62)  
      'rows' =>  
        array(62) {  
          [0] =>  
            array(17) {  
              ...  
            }  
          [1] =>  
            array(17) {
```

Le code fonctionne mais nous permet pas encore d'accéder aux éléments qu'il y a dans l'index rows (contient les produits de nos marques). Voir le code ci-dessous pour réaliser cette tâche.

A partir de maintenant nous allons utiliser notre server local pour avoir un meilleur visuel de notre code. Voir page 6 le visuel.

```
{
  "success": true,
  "rc": 0,
  "message": "",
  "num_rows": 62,
  "rows": [
    {
      "id_product": "569469",
      "brand": "DIESEL",
      "reference": "DZ1847",
      "brand_name": "DIESEL",
      "name": "DIESEL Mod. DZ1847",
      "stock": "3",
      "weight": "300",
      "retail_price": 149,
      "discount": "56.85",
      "price": 64.29,
      "id_supplier": 73082,
      "speed_shipping": "1",
      "ean": "4053858933361",
      "currency": "€",
      "images": [
        {

```

Résultat de la requête get_brand_items avec un id_brand défini sur Postman

Le code suivant nous permet de parcourir l'index rows et de renseigner les caractéristiques de nos marques dans un tableau

```
foreach ($temp as $info) {
  if (isset($info['rows'])) {
    foreach ($info['rows'] as $echo) {
      if (isset($echo['images'])) {
        $image = "";
        foreach ($echo['images'] as $img) {
          $image = $img['icon_path']; //Parcour de l'index image puis on stock l'url des images dans une variable..
          break;
        }
      } else {
        $image = "";
      }
      $infobrands = [
        "Marque" => $echo['brand'],
        "Reference" => $echo['reference'],
        "Nom" => $echo['name'],
        "Stock" => $echo['stock'],
        "prix en détail" => $echo['retail_price'], //Parcour du rows pour afficher chaque détails de nos marques
        "rabais" => $echo['discount'],
        "prix" => $echo['price'],
        "id_fournisseur" => $echo['id_supplier'],
        "Expedition" => $echo['speed_shipping'],
        "Ean" => $echo['ean'],

        "Images" => $image,
      ];
      $info_brand[$echo['id_product']] = $infobrands; //Ajout des détails des marques dans un tableau
    }
  }
}
```


Nous avons créé un tableau HTML que nous avons rendu dynamique grâce à notre PHP.

```
<table id="myTables" class="tablesorter">
  <thead style="cursor: pointer">
    <tr>
      <?php foreach ($infobrand as $key => $value) : ?>
        <th><?= $key ?></th> <!--Parcour de mon tableau $infobrand pour afficher les en-tetes -->
      <?php endforeach ?>
    </tr>
  </thead>
  <tbody id="myTable">
    <?php foreach ($tinfo_brand as $infobrand) : ?>
      <tr>
        <?php foreach ($infobrand as $key => $value) : ?>
          <?php if ($key === 'Images') : ?> <!-- Verifie si il y a une image, si oui il met le lien dans la balise img -->
            <td></td>
          <?php else : ?>
            <td><?= $value ?></td>
          <?php endif; ?>
        <?php endforeach ?>
      </tr>
    <?php endforeach ?>
  </tbody>
</table>
```

Voici le résultat graphique :

Marque	Reference	Nom	Stock	prix en détail	rabais	prix	id_fournisseur	Expedition	Ean	Images
DIESEL	DZ1847	DIESEL Mod. DZ1847	3	149	56.85	64.29	73082	1	4053858933361	
DIESEL	DZ4529	DIESEL Mod. GRIFFED	6	289	60.71	113.56	6058	2	4053858415430	

3.2 Difficultés rencontrées (bug, reste à faire) :

- Manipulation de l'API : Intégrer une API dans nos projets a été un défi initial. Comprendre sa documentation et ses fonctionnalités m'a demandé du temps.

- Trouver de la documentation pertinente sur internet : Trouver des ressources pertinentes et fiables sur Internet pour résoudre des problèmes spécifiques, utiliser des forums de discussion spécialisés (ex : openclassrooms, stack overflow) des tutoriels en ligne (ex : YouTube) et même l'utilisation de l'IA (ex : copilot, chatgpt, blackbox ai) pour trouver des solutions à mes problèmes de programmation.

- Pour la partie de l'ajout d'un produit au panier je dois utiliser la techno Ajax mais je n'ai pas les connaissances nécessaires et le temps d'apprendre celle-ci pour mener à bien cette tâche. De plus, la requête du fournisseur dédié à l'ajout du panier n'est pas fonctionnel.

BILAN

Durant mon stage chez Ether Création, j'ai eu l'opportunité de travailler plus profondément dans le domaine du développement web et d'explorer en détail la manipulation d'une API. Travailler sur l'intégration et l'exploitation de cette API m'a permis d'acquérir une compréhension approfondie des requêtes fournies, ainsi que de perfectionner mes compétences en PHP. Ce fut une expérience enrichissante qui m'a également ouvert les portes vers l'apprentissage de nouveaux concepts, tels que l'utilisation de CURL pour l'envoi de requêtes HTTP.

L'une des composantes les plus stimulantes de ce stage a été d'approfondir mes connaissances sur des outils de développement essentiels, comme Postman. Ce logiciel, bien que nouveau pour moi au début du stage, est devenu rapidement un outil indispensable dans mon travail quotidien. Grâce à lui, j'ai pu tester et valider mes requêtes API de manière plus efficace.

En résumé, mon stage chez Ether Création a été une expérience enrichissante sur les plans professionnel et personnel. Non seulement j'ai pu approfondir mes connaissances techniques et acquérir de nouvelles compétences, mais j'ai également eu l'occasion de me confronter à des défis stimulants qui m'ont permis de progresser en tant que développeur web. Je suis reconnaissant envers toute l'équipe pour leur soutien et leur guidance tout au long de cette expérience.