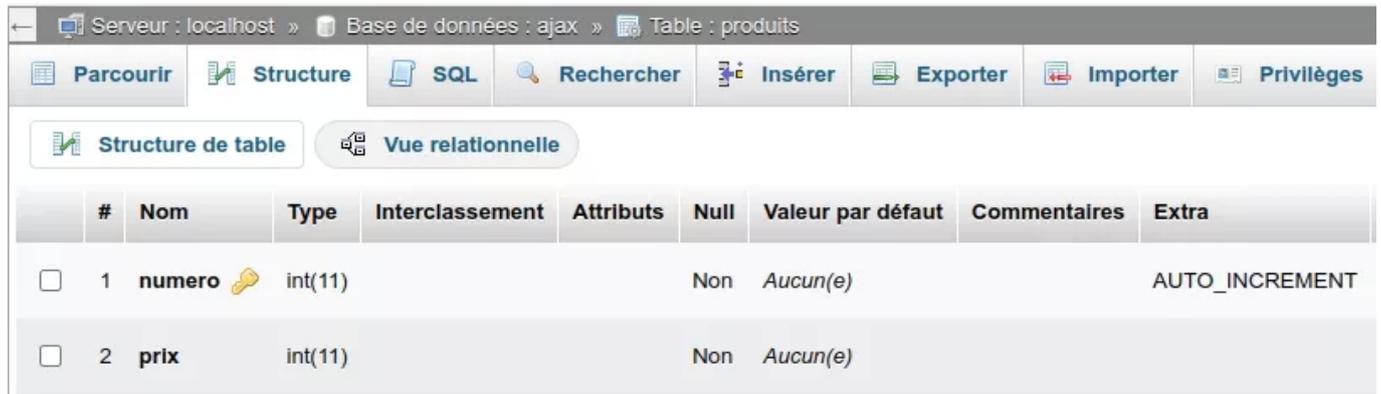


Lire et écrire une table SQL avec AJAX

AJAX (Asynchronous JavaScript And XML) permet d'exécuter des requêtes puis de capter les réponses sans recharger l'ensemble de la page. Grâce au JavaScript qui manipule le **DOM** avec précision, les données pourront être modifiées au coeur de la page Web sans rechargement.

Considérons une table SQL toute simple contenant une clé primaire *numero* auto-incrémentée et un champ numérique *prix*.
But du jeu : cliquer sur un bouton pour augmenter la valeur du *prix*, puis l'afficher.



The screenshot shows a database management interface with the following table structure:

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/>	1	numero 	int(11)		Non	Aucun(e)		AUTO_INCREMENT
<input type="checkbox"/>	2	prix	int(11)		Non	Aucun(e)		

Méthode traditionnelle en PHP

Pour incrémenter le prix à chaque clic, nous utilisons un formulaire HTML et un code PHP :

traditionnel.php :

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="utf-8">
  <title>PHP et MySQL</title>
</head>

<body>
  <h1>Cliquez pour augmenter le prix</h1>
  <form method="post">
    <button type="submit" name="clic">Cliquez</button>
  </form>
  <?php
  $mysqli = new mysqli("localhost", "root", "root", "ajax");
  if (isset($_POST['clic'])) {
    $requete = "UPDATE produits SET prix = prix + 1 WHERE numero = 1";
    $resultat = $mysqli->query($requete);
    if ($resultat) {
      echo "<p>Le prix a augmenté.</p>";
    }
  }
  $requete = "SELECT * FROM produits WHERE numero = 1";
  $resultat = $mysqli->query($requete);
  $ligne = $resultat->fetch_assoc();
  echo "<p>Le prix du numéro " . $ligne["numero"] . " est : " . $ligne["prix"] . " €</p>";
  ?>
</body>

</html>
```

Ici la soumission du formulaire est captée par la fonction `isset()` qui vérifie l'existence d'une variable envoyée par le

formulaire. Les requêtes SQL mettent à jour la valeur du prix puis affichent le prix actuel. Ce code PHP fonctionne parfaitement, façon « Web 2.0 », mais il a l'inconvénient de recharger toute la page pour s'exécuter.

Méthode AJAX

AJAX permet de faire la même chose sans recharger toute la page. Nous allons tout d'abord créer deux scripts PHP indépendants. Le premier script permet d'incrémenter le prix avec une simple requête SQL :

augmentePrix.php :

```
<?php
$mysqli = new mysqli("localhost", "root", "root", "ajax");
$requete = "UPDATE produits SET prix = prix + 1 WHERE numero = 1";
$mysqli->query($requete);
```

Le deuxième script lit la première ligne de la table *produits* puis encode le résultat au format JSON. Cela permet d'envoyer les valeurs de plusieurs variables, à savoir *numero* et *prix*.

lirePrix.php :

```
<?php
$mysqli = new mysqli("localhost", "root", "root", "ajax");
$requete = "SELECT * FROM produits WHERE numero = 1";
$resultat = $mysqli->query($requete);
$ligne = $resultat->fetch_assoc();
echo json_encode($ligne);
```

Le code AJAX ci dessous incrémente le prix suite au clic sur le bouton puis affiche le nouveau prix :

ajax.html :

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="utf-8">
  <title>AJAX</title>
</head>

<body>
  <h1>Cliquez pour augmenter le prix</h1>
  <button onclick="incrimente()">Cliquez</button>
  <p id="textel"></p>
  <p id="texte2"></p>
  <script>
    function incrimente() {
      // On incrémente le prix dans la table produits
      let incr = new XMLHttpRequest();
      incr.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
          document.getElementById("textel").innerText = "Le prix a augmenté.";
        }
      };
      incr.open("POST", "augmentePrix.php", true);
      incr.send();

      // On lit le prix dans la table, puis on l'affiche
      let lire = new XMLHttpRequest();
      lire.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
          var objJSON = JSON.parse(this.responseText);

          document.getElementById("texte2").innerText = "Le prix du numero " +
```

```

objJSON.numero + " est : " + objJSON.prix + " €";
    }
    };
    lire.open("POST", "lirePrix.php", true);
    lire.send();
}
</script>
</body>

</html>

```

Le principe est de créer un objet requête avec `new XMLHttpRequest()` puis de capter la réponse de la requête, de manière désynchronisée, sans bloquer la page, grâce à `XMLHttpRequest.onreadystatechange` :

- `XMLHttpRequest.readyState` prend la valeur 4 quand les opérations sont terminées.
- `XMLHttpRequest.status` - prend la valeur 200 quand la requête a réussi.

Pour finir, je vous propose le même code AJAX, simplifié, grâce à jQuery :

ajaxjQuery.html :

```

<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="utf-8">
  <title>AJAX et jQuery</title>
  <script src="https://code.jquery.com/jquery-latest.js"></script>
</head>

<body>
  <h1>Cliquez pour augmenter le prix</h1>
  <button id="incremente">Cliquez</button>
  <p id="textel"></p>
  <p id="texte2"></p>
  <script>
    $("#incremente").click(function () {
      $.ajax({
        url: "augmentePrix.php",
        method: "POST",
        success: function () {
          $("#textel").text("Le prix a augmenté.");
        }
      });
      $.ajax({
        url: "lirePrix.php",
        method: "POST",
        success: function (data) {
          var objJSON = JSON.parse(data);
          $("#texte2").text("Le prix du numero " + objJSON.numero + " est : " +
objJSON.prix + " €");
        }
      });
    });
  </script>
</body>

</html>

```

Afficher des données au fur et à mesure d'une saisie avec AJAX

Pour illustrer l'efficacité d'AJAX et de ses interactions avec MySQL, je propose de filtrer les communes de France au fur et à mesure de la saisie dans une zone de texte.



Tapez les 3 premières lettres de votre ville

BORD-SAINT-GEORGES
BORDEAUX
BORDEAUX-EN-GATINAIS
BORDEAUX-SAINT-CLAIR
BORDERES
BORDERES-ET-LAMENSANS
BORDERES-LOURON
BORDERES-SUR-L'ECHEZ
BORDES
BORDES
BORDES-DE-RIVIERE
BORDEZAC
BORDS

Importer **les 36000 communes de France** dans la base *ajax* puis tapez le code PHP suivant :

communes.php :

```
<?php
$mysqli = new mysqli("localhost", "root", "", "ajax");
$requete = "SELECT ville_nom FROM communes WHERE ville_nom LIKE '" . $_GET["saisie"] . "%' ";
$resultat = $mysqli->query($requete);
while ($ligne = $resultat->fetch_assoc()) {
    echo $ligne['ville_nom'] . "<br>";
}
```

Ce code affiche les villes dont le nom commencent par une saisie de l'utilisateur. La saisie passe par l'URL `http://localhost/communes.php?saisie=bord` puis elle est récupérée dans la variable `$_GET[« saisie »]`. La requête SQL filtre les noms de villes qui commencent par cette saisie.

A partir de 3 lettres tapées par l'utilisateur, le code AJAX envoie la saisie, pour chaque touche pressée, avec la méthode GET. Dès que la page reçoit les réponses de la requête, les noms des villes s'affichent immédiatement.

communes.html :

```
<!DOCTYPE html>
<html lang="fr">

<head>
    <meta charset="utf-8">
    <title>AJAX et jQuery</title>
    <script src="https://code.jquery.com/jquery-latest.js"></script>
</head>

<body>
    <h1>Tapez les 3 premières lettres de votre ville</h1>
    <input type="text" id="ville" name="ville">
    <p id="texte"></p>
    <script>
        $("#ville").keyup(function () {
```

```
let saisie = $(this).val();
if (saisie.length >= 3) {
    $.ajax({
        url: "communes.php?saisie=" + saisie,
        method: "GET",
        success: function (data) {
            $("#texte").html(data);
        }
    });
}
});
</script>
</body>

</html>
```

L'intérêt de l'AJAX est de créer des formulaires dynamiques qui ne nécessitent pas de rechargement de page à chaque touche pressée.