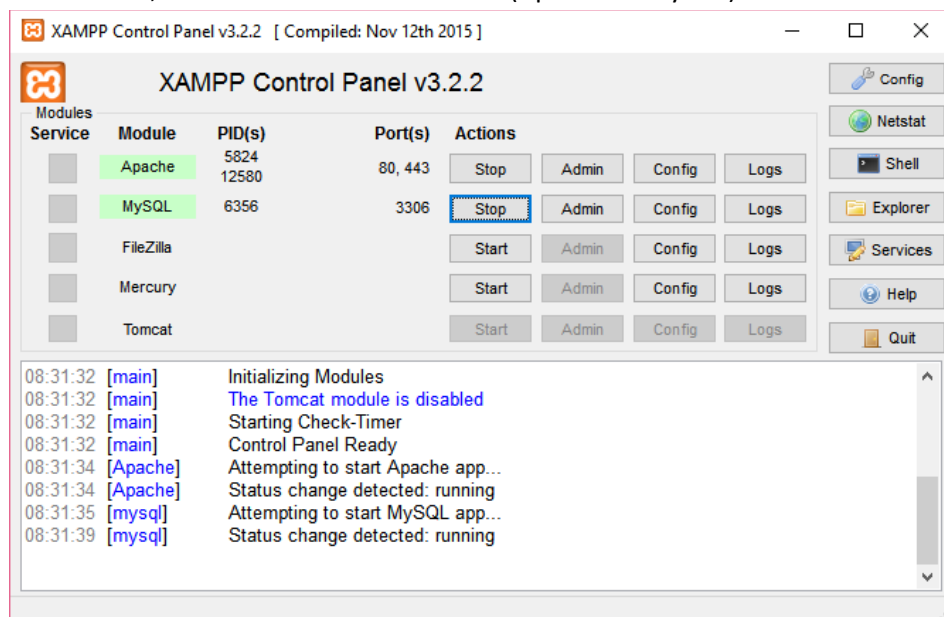


Objectifs :

- Mise en œuvre des concepts de base de HTML, CSS, JavaScript dans une application de gestion,
- Création d'une application web avec le PHP 5,
- Découvrir le PHP Data Objects (PDO),
- Développement des CRUD,
- Création des web services REST,
- Tester les web services avec l'extension de Google chrome « Advanced REST Client »,

Partie 1 : Création de la base de données

1. Démarrer le **XAMP**, et lancer les services suivant (Apache et MySQL) :



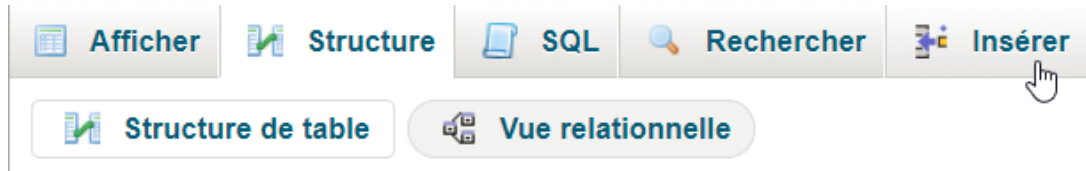
2. Connectez-vous vers : <http://localhost/phpmyadmin/>
3. Cliquez sur «**Bases de données**», et créer la base de données « **school1** » :



4. Sélectionnez la base de données « **school1** », ensuite créer la table « **Etudiant** » suivante :

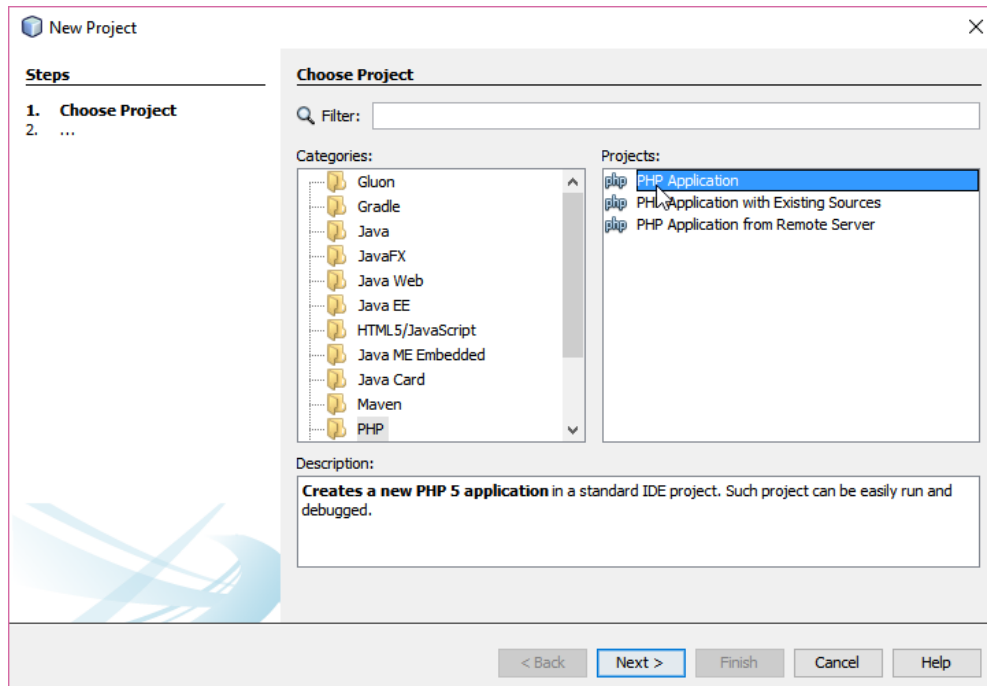
| # | Nom | Type | Interclassement | Attributs | Null | Défaut | Extra |
|----------------------------|---------------|-------------|-----------------|-----------|------|--------|----------------|
| <input type="checkbox"/> 1 | id | int(11) | | | Non | Aucune | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | nom | varchar(50) | | | Non | Aucune | |
| <input type="checkbox"/> 3 | prenom | varchar(50) | | | Non | Aucune | |
| <input type="checkbox"/> 4 | ville | varchar(50) | | | Non | Aucune | |
| <input type="checkbox"/> 5 | sexe | varchar(20) | | | Non | Aucune | |

5. Ajouter un jeu d'enregistrement dans la table « Etudiant », en cliquant sur « Insérer » :

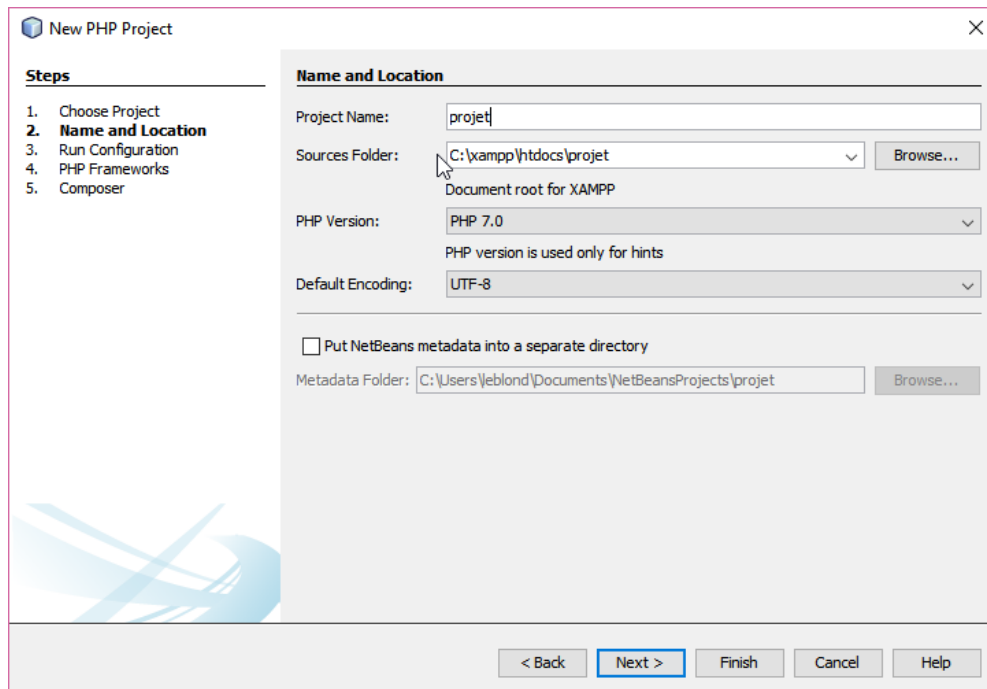


Etape 2 : Développement de l'application web

1. Démarrer « Netbeans », ensuite créer un projet php :

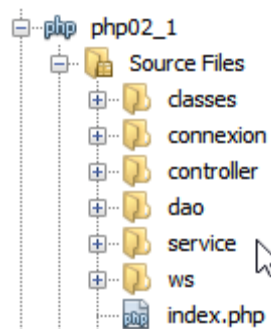


2. Cliquez sur « Next », l'écran suivant apparaît :



Saisissez le nom de votre projet, ensuite enregistrez le dans « **C:\xampp\htdocs\projet** ».

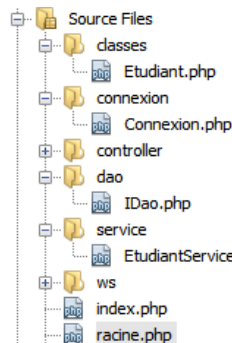
3. Dans votre projet créer les dossiers suivants, pour ce faire cliquez droit sur le projet, ensuite « **New Folder** » :



4. Ajouter le fichier « **racine.php** » dans la racine de votre projet, et définissez la constante « **RACINE** » qui représente le chemin vers la racine de votre projet :

`<?php`

```
define("RACINE", __DIR__);
```



5. Dans le dossier « **classes** », créer la classe « **Etudiant** » suivante :
`<?php`

```
class Etudiant {  
  
    private $id;  
    private $nom;  
    private $prenom;  
    private $ville;  
    private $sexe;  
  
    function __construct($id, $nom, $prenom, $ville, $sexe) {  
        $this->id = $id;  
        $this->nom = $nom;  
        $this->prenom = $prenom;  
        $this->ville = $ville;  
        $this->sexe = $sexe;  
    }  
  
    function getId() {  
        return $this->id;  
    }  
  
    function getNom() {  
        return $this->nom;  
    }  
  
    function getPrenom() {  
        return $this->prenom;  
    }  
  
    function getVille() {  
        return $this->ville;  
    }  
  
    function getSexe() {  
        return $this->sexe;  
    }  
  
    function setId($id) {  
        $this->id = $id;  
    }  
  
    function setNom($nom) {  
        $this->nom = $nom;  
    }  
  
    function setPrenom($prenom) {  
        $this->prenom = $prenom;  
    }  
  
    function setVille($ville) {  
        $this->ville = $ville;  
    }  
  
    function setSexe($sexe) {  
        $this->sexe = $sexe;  
    }  
  
    public function __toString() {  
        return $this->nom . " " . $this->prenom;  
    }  
}
```

6. Dans le dossier « connexion », créer la classe « **Connexion** » suivante :

```
<?php

class Connexion {

    private $connexion;

    public function __construct() {
        $host = 'localhost';
        $dbname = 'school1';
        $login = 'root';
        $password = '';
        try {
            $this->connexion = new PDO("mysql:host=$host;dbname=$dbname", $login, $password);
            $this->connexion->query("SET NAMES UTF8");
        } catch (Exception $e) {
            die('Erreur : ' . $e->getMessage());
        }
    }

    function getConnexion() {
        return $this->connexion;
    }

}
```

7. Dans le dossier « dao », créer l'interface « **IDao** » suivante :

```
<?php
interface IDao {
    function create($o);
    function delete($o);
    function update($o);
    function findAll();
    function findById($id);
}
```

8. Dans le dossier « service », créer la classe « **EtudiantService** » qui implémente l'interface « **IDao** », ensuite redéfinissez les méthodes :

```
<?php

include_once RACINE . '/classes/Etudiant.php';
include_once RACINE . '/connexion/Connexion.php';
include_once RACINE . '/dao/IDao.php';

class EtudiantService implements IDao {

    private $connexion;

    function __construct() {
        $this->connexion = new Connexion();
    }

    public function create($o) {
        $query = "INSERT INTO Etudiant (`id`, `nom`, `prenom`, `ville`, `sexe`) "
            . "VALUES (NULL, '" . $o->getNom() . "', '" . $o->getPrenom() . "', "
            . "' " . $o->getVille() . "', '" . $o->getSexe() . "')";
        $req = $this->connexion->getConnexion()->prepare($query);
        $req->execute() or die('Erreur SQL');
    }

    public function delete($o) {
        $query = "delete from Etudiant where id = " . $o->getId();
        $req = $this->connexion->getConnexion()->prepare($query);
        $req->execute() or die('Erreur SQL');
    }

    public function findAll() {
        $etds = array();
        $query = "select * from Etudiant";
        $req = $this->connexion->getConnexion()->prepare($query);
```

```

$req->execute();
while ($e = $req->fetch(PDO::FETCH_OBJ)) {
    $etds[] = new Etudiant($e->id, $e->nom, $e->prenom, $e->ville, $e->sexe);
}
return $etds;
}

public function findById($id) {
    $query = "select * from Etudiant where id = " . $id;
    $req = $this->connexion->getConnexion()->prepare($query);
    $req->execute();
    if ($e = $req->fetch(PDO::FETCH_OBJ)) {
        $etd = new Etudiant($e->id, $e->nom, $e->prenom, $e->ville, $e->sexe);
    }
    return $etd;
}

public function update($o) {
    $query = "UPDATE `etudiant` SET `nom` = '" . $o->getNom() . "', `prenom` = '" .
        $o->getPrenom() . "', `ville` = '" . $o->getVille() . "', `sexe` = '" .
        $o->getSexe() . "' WHERE `etudiant`.`id` = " . $o->getId();
    $req = $this->connexion->getConnexion()->prepare($query);
    $req->execute() or die('Erreur SQL');
}
}

```

9. Dans le fichier **index.php**, créer l'interface graphique suivante :

Ajouter un nouveau étudiant

Nom :

Prenom :

Ville :

Sexe : ☐ M ☐ F

| ID | Nom | Prenom | Ville | Sexe | Supprimer | Modifier |
|----|----------|---------|-----------|-------|---------------------------|--------------------------|
| 37 | LACHGATR | Mohamed | Marrakech | homme | Supprimer | Modifier |
| 38 | RAMI | Imane | Agadir | femme | Supprimer | Modifier |
| 39 | SAFI | Ali | Rabat | homme | Supprimer | Modifier |

```

<!DOCTYPE html>
<?php
include_once './racine.php';
?>
<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
<body>
<form method="GET" action="controller/addEtudiant.php">

<fieldset>
<legend>Ajouter un nouveau étudiant</legend>

<table border="0">

<tr>
<td>Nom : </td>
<td><input type="text" name="nom" value="" /></td>
</tr>
<tr>
<td>Prenom :</td>
<td><input type="text" name="prenom" value="" /></td>

```

```

        </tr>
        <tr>
            <td>Ville</td>
            <td>
                <select name="ville">
                    <option value="Marrakech">Marrakech</option>
                    <option value="Rabat">Rabat</option>
                    <option value="Agadir">Agadir</option>

                </select>
            </td>
        </tr>
        <tr>
            <td>Sexe </td>
            <td>
                M<input type="radio" name="sexe" value="homme" />
                F<input type="radio" name="sexe" value="femme" />
            </td>
        </tr>
        <tr>
            <td></td>
            <td>
                <input type="submit" value="Envoyer" />
                <input type="reset" value="Effacer" />
            </td>
        </tr>
    </table>

</fieldset>
</form>
<table border="1">
    <thead>
        <tr>
            <th>ID</th>
            <th>Nom</th>
            <th>Prenom</th>
            <th>Ville</th>
            <th>Sexe</th>
            <th>Supprimer</th>
            <th>Modifier</th>
        </tr>
    </thead>
    <tbody>
        <?php
            include_once RACINE . '/service/EtudiantService.php';
            $es = new EtudiantService();
            foreach ($es->findAll() as $e) {
                ?>
                <tr>
                    <td><?php echo $e->getId(); ?></td>
                    <td><?php echo $e->getNom(); ?></td>
                    <td><?php echo $e->getPrenom(); ?></td>
                    <td><?php echo $e->getVille(); ?></td>
                    <td><?php echo $e->getSexe(); ?></td>
                    <td>
                        <a href="controller/deleteEtudiant.php?id=
                            <?php echo $e->getId(); ?>">Supprimer</a> </td>
                    <td><a href="updateEtudiant.php">Modifier</a></td>
                </tr>
            <?php } ?>
        </tbody>
    </table>

</body>
</html>

```

10. Dans le dossier « **controller** », créer les fichiers php suivants :

addEtudiant.php

<?php

```

include_once '../racine.php';
include_once RACINE.'/service/EtudiantService.php';
extract($_GET);

$es = new EtudiantService();
$es->create(new Etudiant(1, $nom, $prenom, $ville, $sexe));

header("location:../index.php");

```

deleteEtudiant.php

```

<?php

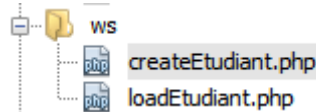
include_once '../racine.php';
include_once RACINE.'/service/EtudiantService.php';
extract($_GET);

$es = new EtudiantService();
$es->delete($es->findById($id));
header("location:../index.php");

```

Etape 3 : Développement des web services

1. Dans le dossier « ws », créer les services suivantes :



createEtudiant.php

```

<?php
if($_SERVER["REQUEST_METHOD"] == "POST"){
    include_once '../racine.php';
    include_once RACINE.'/service/EtudiantService.php';
    create();
}
function create(){
    extract($_POST);
    $es = new EtudiantService();
    $es->create(new Etudiant(1, $nom, $prenom, $ville, $sexe));
}

```

loadEtudiant

```

<?php

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    include_once '../racine.php';
    include_once RACINE . '/service/EtudiantService.php';
    loadAll();
}

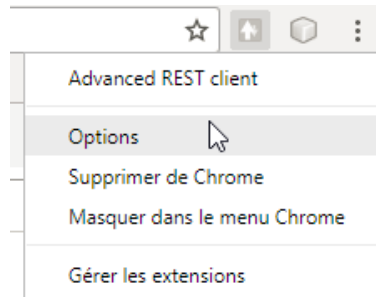
function loadAll() {
    $es = new EtudiantService();
    header('Content-type: application/json');
    echo json_encode(array("etudiants" => $es->getAll()));
}

```

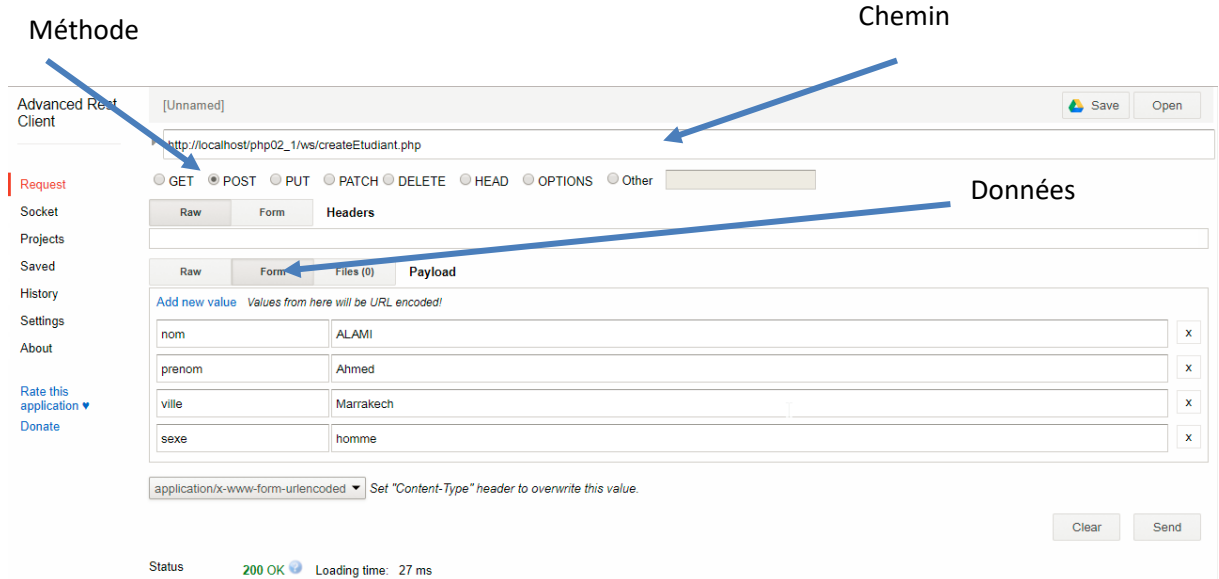

Ajouter la méthode **getAll()** suivante dans la classe « **EtudiantService** »

```
public function getAll() {
    $query = "select * from Etudiant";
    $req = $this->connexion->getConnexion()->prepare($query);
    $req->execute();
    return $req->fetchAll(PDO::FETCH_ASSOC);
}
```

2. Afin de tester les services, ajouter l'extension « **Advanced REST client** » dans google chrome.



Ensuite cliquez sur « Options ». L'interface suivante apparaît :



Cliquez sur « **Request** », ensuite saisissez le chemin vers votre service, la méthode d'envoi, et les données à envoyer. Ensuite cliquez sur « **Send** ».

Vérifier l'ajout dans la base de données.

Ensuite, tester le deuxième service :

http://localhost/php02_1/ws/loadEtudiant.php

☐ GET ☒ POST ☐ PUT ☐ PATCH ☐ DELETE ☐ HEAD ☐ OPTIONS ☐ Other

Raw Form Headers

Raw Form Files (0) Payload

[Add new value](#) Values from here will be URL encoded!

application/x-www-form-urlencoded Set "Content-Type" header to overwrite this value.

Clear Send

Résultat :

| Raw | JSON | Response |
|--|------|----------|
| Copy to clipboard Save as file | | |
| <pre>{ -etudiants: [4] -0: { id: "37" nom: "LACHGATR" prenom: "Mohamed" ville: "Marrakech" sexe: "homme" } -1: { id: "38" nom: "RAMI" prenom: "Imane" ville: "Agadir" sexe: "femme" } -2: { id: "39" nom: "SAFI" prenom: "Ali" ville: "Rabat" sexe: "homme" } -3: { id: "40" nom: "ALAMI" prenom: "Ahmed" ville: "Marrakech" sexe: "homme" } } }</pre> | | |