



# Développement web en

## - Interaction avec les bases de données -

**Groupe des étudiants : CIR2**

Ayoub KARINE

E-Mail :

Numéro de bureau : A3-72

# Front-end VS Back-end

## Front-end (web statique)



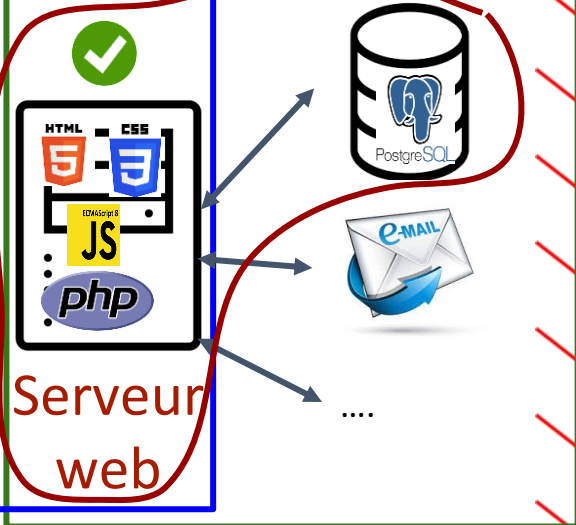
Envoi d'une requête



Internet

Renvoi d'une  
page web  
générée  
dynamiquement

## Back-end (web dynamique)

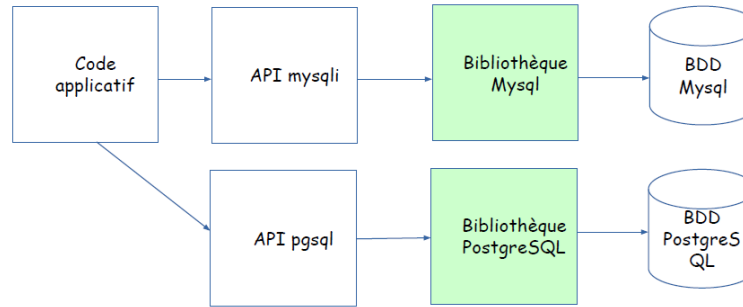


Interprétation du  
code par le serveur

# Types d'interaction avec la BD

- **Approche procédurale**

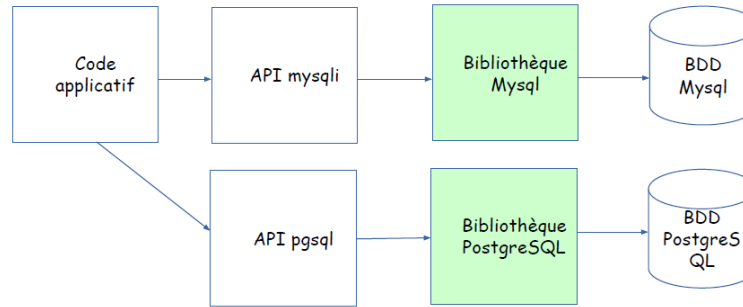
- mysqli
- pgsql
- ...



# Types d'interaction avec la BD

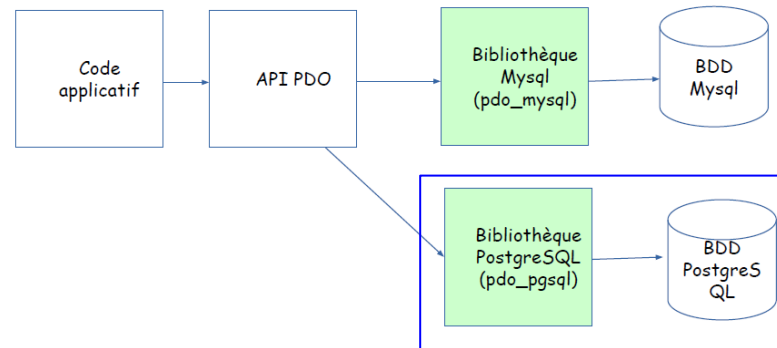
- **Approche procédurale**

- mysqli
- pgsql
- ...



- **Approche orientée objet**

- PDO ( Data Objects) à partir de 5
- ...



# PDO

- PDO utilise 3 classes principales :
  - PDO : qui gère la connexion à la BDD et exécute les requêtes
  - PDOStatement : qui correspond aux requêtes et aux résultats (des requêtes)
  - PDOException : qui correspond aux erreurs de PDO

[https://www.php.net/manual/fr/book.pdo.](https://www.php.net/manual/fr/book.pdo)

# Connexion à la BD

- La connexion à la BD est établie via le constructeur de la classe PDO

```
public PDO::__construct ( string $dsn [, string $username [, string $passwd [, array $options ]]] )
```

- `$dsn` : nom d'hôte où mysql/pgsql est installé
- `$username` : nom d'utilisateur
- `$passwd` : mot de passe
- `$options` : autres options spécifiques à la connexion

# Connexion à la BD

- Exemple avec MySQL

```
<?
$dsn = 'mysql:dbname=testdb;host=127.0.0.1';
$user = 'dbuser';
$password = 'dbpass';

try {
    $conn = new PDO($dsn, $user, $password);
} catch (PDOException $e) {
    echo 'Connexion échouée : ' . $e->getMessage();
}

?>
```

- Exemple avec Postgre

```
<?
$dsn = 'pgsql:dbname=testdb;host=127.0.0.1;port=5432';
$user = 'dbuser';
$password = 'dbpass';

try {
    $conn = new PDO($dsn, $user, $password);
} catch (PDOException $e) {
    echo 'Connexion échouée : ' . $e->getMessage();
}

?>
```

# Accès en lecture pour les requêtes non préparées

- **query** :  
`public PDO::query ( string $statement )`
- **fetch** -> récupère la ligne suivante d'une table :  
`public PDOStatement::fetch ([ int $fetch_style [, int $cursor_orientation = PDO::FETCH_ORI_NEXT [, int $cursor_offset = 0 ]]] ) : mixed`
- **fetchAll** -> retourne un tableau contenant toutes les lignes de la table :  
`public PDOStatement::execute ([ array $input_parameters ] )`

```
<?
$dsn = 'pgsql:dbname=testdb;host=127.0.0.1;port=5432';
$user = 'dbuser';
$password = 'dbpass';

try {
    $conn = new PDO($dsn, $user, $password);
} catch (PDOException $e) {
    echo 'Connexion échouée : ' . $e->getMessage();
}
?>
```

```
<?
    $statement = $conn->query('SELECT * FROM user');
    $result = $statement->fetchAll(PDO::FETCH_ASSOC);
?>
```



# Accès en lecture pour les requêtes préparées

- **prepare** :  
`public PDO::prepare ( string $statement [, array $driver_options = array() ] )`
- **execute** :  
`public PDOStatement::execute ([ array $input_parameters ] )`
- **fetch** -> récupère la ligne suivante d'une table :  
`public PDOStatement::fetch ([ int $fetch_style [, int $cursor_orientation = PDO::FETCH_ORI_NEXT [, int $cursor_offset = 0 ]]] ) : mixed`
- **fetchAll** -> retourne un tableau contenant toutes les lignes de la table :  
`public PDOStatement::execute ([ array $input_parameters ] )`

```
<?
$dsn = 'pgsql:dbname=testdb;host=127.0.0.1;port=5432';
$user = 'dbuser';
$password = 'dbpass';

try {
    $conn = new PDO($dsn, $user, $password);
} catch (PDOException $e) {
    echo 'Connexion échouée : ' . $e->getMessage();
}
?>
```

```
<?
$name = 'Dupont';
$request = 'SELECT * FROM users WHERE name=:name';
$stmt = $conn->prepare($request);
$stmt->bindParam(':name', $name);
$stmt->execute();
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);

?>
```

- Insertion en utilisant une requête non préparée

```
<?
    $sql = "INSERT INTO REGISTRY (name, value) VALUES ('one', 1)";
    if ($conn->query($sql) === TRUE) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
?>
```

- Suppression en utilisant une requête non préparée

- **exec** : `PDO::exec ( string $statement )`

```
$conn->exec("DELETE from MyGuests WHERE id=1");
```

- **Insertion en utilisant une requête préparée**

```
<?
    $stmt = $conn->prepare("INSERT INTO REGISTRY (name, value) VALUES (:name, :value)");
    $name = 'one';
    $value = 1;
    $stmt->bindParam(':name', $name);
    $stmt->bindParam(':value', $value);
    $stmt->execute();
?>
```

- **Suppression en utilisant une requête préparée**

```
<?
    $idval = 1;
    $stmt = $conn->prepare("DELETE from MyGuests WHERE id=:idval");
    $stmt->bindParam(':idval', $idval);
    $stmt->execute();
?>
```