

PHP Unit 3- MySQL, Email, and Web Services

Outlines:

Data Access with MySQL, Email, and Web Services:

Using MySQL with PHP:

- Connect to MySQL and select the database
- Execute simple queries
- Retrieve query results
- Count returned records
- Update records
- Add, view, and delete records with PHP

Email:

- Understand email
- Receive email with PHP
- Send email

Web Services:

- Introduction to web services
- Web services model
- Platform elements

Data Access with MySQL, Email, and Web Services

Using MySQL with PHP

MySQL is a popular open-source database management system, commonly used with PHP. In this chapter, we will explore how to use MySQL with PHP.

Using MySQL with PHP We can:

- Connect to MySQL and select the database.

- Execute simple queries
- Retrieve query results
- Count returned records
- Update records
- Add, view, and delete records with PHP

Connecting to MySQL and Selecting the Database

To connect to MySQL, we use the `mysqli_connect()` function in PHP. This function takes four arguments: the server name, username, password, and database name. Here's an example:

Syntax

```
resource mysqli_connect (server, username, password)
```

Example

```
$con = mysqli_connect("localhost", "my_user", "my_password");
```

PHP mysqli_close()

PHP **mysqli_close()** function is used to disconnect with MySQL database. It returns *true* if connection is closed or *false*.

Syntax

```
bool mysqli_close(resource $resource_link)
```

Code

```
<?php
$host = 'localhost';
$user = '';
$pass = '';
$conn = mysqli_connect($host, $user, $pass);
if(! $conn )
{
    die('Could not connect: ' . mysqli_error());
}
echo 'Connected successfully';
```

```
mysqli_close($conn);  
?>
```

Once we have connected to the database, we need to select the database we want to work with:

```
mysqli_select_db($con, "my_db");
```

Executing Simple Queries

The **mysqli_query()** function accepts a string value representing a query as one of the parameters and, executes/performs the given query on the database.

- To execute a simple query with MySQL, we use the `mysqli_query()` function in PHP.
- This function takes two arguments: the database connection and the SQL query. Here's an example:

```
$result = mysqli_query($con, "SELECT * FROM my_table");
```

Example:

Retrieving Query Results

To retrieve the results of a query, we use the `mysqli_fetch_array()` function in PHP. This function returns an array of the rows in the result set. Here's an example:

```
while ($row = mysqli_fetch_array($result)) {  
    echo $row['column_name'];  
}
```

Counting Returned Records

To count the number of records returned by a query, we use the `mysqli_num_rows()` function in PHP. Here's an example:

```
$num_rows = mysqli_num_rows($result);  
echo "There are " . $num_rows . " rows in the result set.";
```

Updating Records

To update records in a MySQL database, we use the `mysqli_query()` function with an `UPDATE` statement. Here's an example:

```
mysqli_query($con, "UPDATE my_table SET column_name = 'new_  
value' WHERE id = 1");
```

Adding, Viewing, and Deleting Records with PHP

To add a record to a MySQL database, we use the `mysqli_query()` function with an `INSERT` statement. Here's an example:

To view a record from a MySQL database, we use a `SELECT` statement:

```
mysqli_query($con, "INSERT INTO my_table (column1, column2)  
VALUES ('value1', 'value2')");
```

```
$result = mysqli_query($con, "SELECT * FROM my_table WHERE  
id = 1");  
$row = mysqli_fetch_array($result);  
echo $row['column_name'];
```

Full Example:

```
<?php  
$host = 'localhost';  
$user = 'root';  
$pass = '';  
$dbname = 'test';  
  
$conn = mysqli_connect($host, $user, $pass,$dbname);  
if(!$conn){  
    die('Could not connect: '.mysqli_connect_error());  
}
```

```

echo 'Connected successfully<br/>';

$sql = 'INSERT INTO user (u_name, u_password, u_city) VALUES

    if(mysqli_query($conn, $sql))
        echo "Record Inserted successfully";
    else
        echo "Could not Insert record: ". mysqli_error($conn);

mysqli_close($conn);

```

Output:

```

Connected successfully.
Record inserted successfully

```

To delete a record from a MySQL database, we use the `mysqli_query()` function with a `DELETE` statement. Here's an example:

```

mysqli_query($con, "DELETE FROM my_table WHERE id = 1");

```

Full Example:

```

<?php
$host = 'localhost';
$user = 'root';
$pass = '';
$dbname = 'test';

$conn = mysqli_connect($host, $user, $pass,$dbname);
if(!$conn){
    die('Could not connect: '.mysqli_connect_error());
}
echo 'Connected successfully<br/>';

$sql = 'DELETE FROM user WHERE username="Naruto"';

```

```

if(mysqli_query($conn, $sql))
    echo "Record Deleted successfully";
else
    echo "Could not Delete record: ". mysqli_error($conn);

mysqli_close($conn);

```

PHP MySQL Order By

PHP `mysql_query()` function is used to execute select query with order by clause.

- The order by clause is used to fetch data in ascending order or descending order on the basis of column.

Let's see the query to select data from emp4 table in ascending order on the basis of name column.

```
SELECT * FROM users order by name
```

Let's see the query to select data from emp4 table in descending order on the basis of name column.

```
SELECT * FROM users order by name desc
```

Example:

```

<?php
$host = 'localhost';
$user = 'root';
$pass = '';
$dbname = 'test';

$conn = mysqli_connect($host, $user, $pass, $dbname);
if (!$conn) {
    die('Could not connect: ' . mysqli_connect_error());
}
echo 'Connected successfully<br/>';

$sql1 = 'SELECT * FROM users ORDER BY u_name';

$result = mysqli_query($conn, $sql1);

if (mysqli_num_rows($result) > 0) {

```

```
// output data of each row
while ($row = mysqli_fetch_assoc($result)) {
    echo "Name: " . $row["u_name"] . " City: " . $row["u_city"]
}
} else {
    echo "0 results";
}
```

```
Connected successfully
Name: Itachi City: Konoha
Name: Kakashi City: Konoha
Name: Kisame City: Konoha
Name: Naruto City: Konoha
Name: Sakura City: Konoha
Name: Sasuke City: Konoha
```

Email

Overview of email and PHP mail() function:

- Email is an electronic communication method that allows individuals to send and receive messages, files, and other digital content between computers and other electronic devices.
- The PHP mail() function is a built-in function in PHP that enables developers to send emails from their PHP scripts. The basic syntax of the mail() function is as follows: *mail(to, subject, message, headers, parameters);*

where:

to : The recipient's email address.

subject : The subject of the email.

message : The content of the email, which can include HTML and text.

headers : Optional additional headers that can be added to the email, such as CC, BCC, and Reply-To.

parameters : Optional parameters that can be passed to the underlying mail system, such as the path to a log file.

The PHP mail() function uses the Simple Mail Transfer Protocol (SMTP) to send emails, which is the standard protocol for sending email on the internet. The PHP mail() function is relatively simple to use, but has several limitations, including limited support for HTML and attachments, lack of error reporting, and potential security risks if not used properly.

Configuring PHP for sending email:

To configure PHP for sending email, the following steps need to be followed:

1. Install a mail server: To send email from PHP, you need to have a mail server installed on your system. The most commonly used mail servers for PHP are Postfix, Sendmail, and Exim.
2. Configure the PHP.ini file: The PHP configuration file (php.ini) needs to be configured to specify the location of the mail server and other email-related settings.
3. Specify the sendmail_path setting: The sendmail_path setting in the php.ini file specifies the path to the sendmail or equivalent executable on your system. For example: sendmail_path = "/usr/sbin/sendmail -t -i"
4. Set the SMTP server: If you want to use an external SMTP server to send email, you need to set the SMTP server in the php.ini file. For example: SMTP = smtp.example.com
5. Set the sendmail_from setting: The sendmail_from setting specifies the email address that should be used as the "From" address in emails sent by PHP. For example: sendmail_from = "noreply@example.com"
6. Restart the web server: After making changes to the php.ini file, you need to restart your web server to ensure that the changes take effect.

Creating a PHP script to send email:

The following is a simple example of a PHP script that sends an email using the PHP mail() function:

```
<?php
$to = "recipient@example.com";
$subject = "Test Email";
$message = "This is a test email sent from a PHP script.";

$headers = "From: sender@example.com" . "\r\n" .
"CC: cc_recipient@example.com";
```



```
mail($to, $subject, $message, $headers);  
echo "Email sent successfully.";
```

- In this example, the PHP mail() function is used to send an email to recipient@example.com with a subject of "Test Email" and a message of "This is a test email sent from a PHP script." The \$headers variable is used to specify the "From" and "CC" headers for the email.
 - After the email is sent, the script outputs "Email sent successfully." to indicate that the email was sent successfully.
 - It is important to note that when sending emails from PHP scripts, it is essential to validate the data entered by the user before sending the email to prevent security vulnerabilities and other issues.
-

Introduction to Web Services:

- A Web service is a **method of communication between two electronic devices over a network**
- It integrates the web-based application using the **REST, SOAP, WSDL**, and **UDDI** over the network. For example, Java web service can communicate with .Net application.
- Web services are designed for application to application interaction.

Web Service Components

There are three major web service components.

- SOAP (Simple Object Access Protocol)
 - UDDI (Universal Description, Discovery and Integration)
 - WSDL (Web Services Description Language)
-

SOAP

SOAP stands for Simple Object Access Protocol.

SOAP is a XML-based protocol for accessing web services.

SOAP is a W3C recommendation for communication between applications.

SOAP is XML based, so it is platform independent and language independent. In other words, it can be used with Java, .Net or PHP language on any platform.

WSDL

WSDL stands for Web Services Description Language.

WSDL is a xml document containing information about web services such as method name, method parameter and how to access it.

WSDL is a part of UDDI. It acts as a interface between web service applications.

WSDL is pronounced as wiz-dull.

UDDI

UDDI stands for Universal Description, Discovery and Integration.

UDDI is a XML based framework for describing, discovering and integrating web services.

UDDI is a directory of web service interfaces described by WSDL, containing information about web services.

Uses of Web Services

- Web services are used for reusing the code and connecting the existing program.
- Web services can be used to link data between two different platforms.
- It provides interoperability between disparate applications.

There are two popular formats for request and response **XML** and **JSON**.

XML Format: XML (**Extensible Markup Language**, it is the popular form as request and response in web services. Consider the following XML code:

JSON Format: JSON is a readable format for structuring data. It is used for transiting data between server and web application.

Introduction of SOAP and REST Web Services

- SOAP and REST are two popular web service technologies.
- SOAP (Simple Object Access Protocol) is an XML-based messaging protocol used to exchange data between different machines on a network.
- REST (Representational State Transfer) is an architectural style that uses standard protocols like HTTP/HTTPS to create a lightweight, scalable, and maintainable web service.
- While SOAP is more heavyweight and requires more resources, REST is lightweight and has better performance. Both have their advantages and disadvantages and are used in different scenarios.

SOAP Web Services

- **SOAP** stands for **Simple Object Access Protocol** is a **network platform** used in a web service to exchange or communicate data between two different machines on a network.
- It uses the XML format of data to transfer messages over the HTTP protocol.
- In Web services, SOAP allows the user request to interact with other programming languages.
- In this way, it provides a way to communicate between applications running on different platforms (Operating system), with programming languages and technologies used in web service.

REST Web Service

- REST stands for REpresentational State Transfer.
- It is an architectural style that provides standards between computer systems on a web service.
- It is a set of constraints used to create a lightweight, scalable, and maintainable web service that facilitates easy communication.
- It is a set of constraints used to create a lightweight, scalable and maintainable web service that makes easy communication with each other.

- All web services are based on the REST; hence it is called a RESTful service.
- The purpose of developing a RESTful web service is to make the web service more effective.
- The Roy Fielding develops it in 2000, who also developed the HTTP protocol.
- It does not depend on a specific protocol to use, but it's used HTTP/HTTPS.

HTTP Method of REST web service

GET: It is used to fetch information from the resource server.

POST: It is used to create or insert new information on the resource server.

DELETE: It is used to remove the records or information from the resource server.

PUT: It is used to manipulate or update the existing records on the server.

Advantage of REST

- REST web services are fast as compared to SOAP because it has no restriction like SOAP. It consumes less bandwidth and resources.
- It is an architectural style for creating lightweight, scalable and maintainable web service.
- It is a language and platform-independent web service that can be written in any programming language and run on Windows, Linux and Mac.
- REST web service helps the client machine access different formats of data such as HTML, JSON, XML, etc. from the webserver.

Difference Between SOAP and REST Web Service

SOAP Web Service	REST Web Service
It stands for Simple Object Access Protocol.	It stands for REpresentational State Transfer.
It is XML based messaging protocol.	It is not a protocol. It is an architectural style for distributed hypermedia system.

SOAP Web Service	REST Web Service
It needs more bandwidth and resources for better web performance.	REST requires less bandwidth and resources as compared to SOAP.
SOAP enforces XML as a message format.	It is not specifically applied that the message format must be XML or JSON, etc.
It has not great performance as compared to REST.	It has better performance as compared to SOAP, less CPU intensive, lesser code, etc.
SOAP defines its security.	REST inherits security measures from the underlying transport.
It does not support error handling.	It has built-in error handling.
SOAP is a heavyweight XML protocol that requires more coding to send a message.	It is a lightweight, scalable and maintainable.
It cannot be cached.	It can be cached.
SOAP messages are wrapped in an envelope and sent to any transport mechanism such as SMTP, FTP, HTTP or any protocol.	It relies on the HTTP protocol for communication between two machines.

DTD (Document Type Definition)

- DTD stands for **Document Type Definition**.
- It is used to define the legal structure and elements of an XML document.
- A DTD specifies the allowed elements, attributes, and entities that can be used in an XML document.

Advantages of DTD

- It can be either internal or external. Internal DTD is defined within the XML document, whereas External DTD is defined in a separate file and linked to the XML document.
- DTD provides a clear and concise way to define the structure of an XML document.
- It allows for the validation of an XML document, which ensures that the document is well-formed and matches the specified structure.
- DTD allows for the reuse of defined structures across multiple documents, resulting in consistent formatting and structure.
- It can help maintain consistency and accuracy within the document and ensure that no invalid data is entered.

Disadvantages of DTD

- DTD has limitations in defining complex structures, and it can be difficult to maintain for large and complex documents.
- It is less flexible than other schema languages, as it does not support advanced features like data types or inheritance.
- DTD is not able to define or check data types, such as integers or dates.
- It cannot handle namespaces, which are used to avoid naming conflicts between different XML vocabularies.

In summary, DTD is a simple and widely-used way of defining the structure and elements of an XML document. It is useful for small to medium-sized documents that do not require complex validation or data typing. However, for larger or more complex documents, other schema languages like XML Schema or Relax NG may be more appropriate.

Differences between SAX and DOM

DOM	SAX
Creates a walkable object tree in memory	Treats XML as flowthrough string data
Can get or set the children, parents, and text content of each part of the tree	Good for repetitive tasks applied to all elements of a certain type

DOM	SAX
Works best if you have a complete XML document available	Passes through a document once from top to bottom
Can save DOM objects to containers as well as write them out as text	Cannot "go back" and do things based on inputs later in the document
Resource-intensive with large documents	Lightweight and easier to learn