

PHP - Post-Redirect-Get (PRG)

In PHP, PRG stands for "Post/Redirect/Get". It is a commonly used technique that is designed to prevent the resubmission of a form after it's been submitted. You can easily implement this technique in PHP to avoid duplicate form submissions.

Usually a HTML form sends data to the server with the POST method. The server script fetches the data for further processing like adding a new record in a backend database, or running a query to fetch data. If the user accidentally refreshes the browser, there is a possibility of the same form data being resubmitted again, possibly leading to loss of data integrity. The PRG approach in PHP helps you avoid this pitfall.

Example

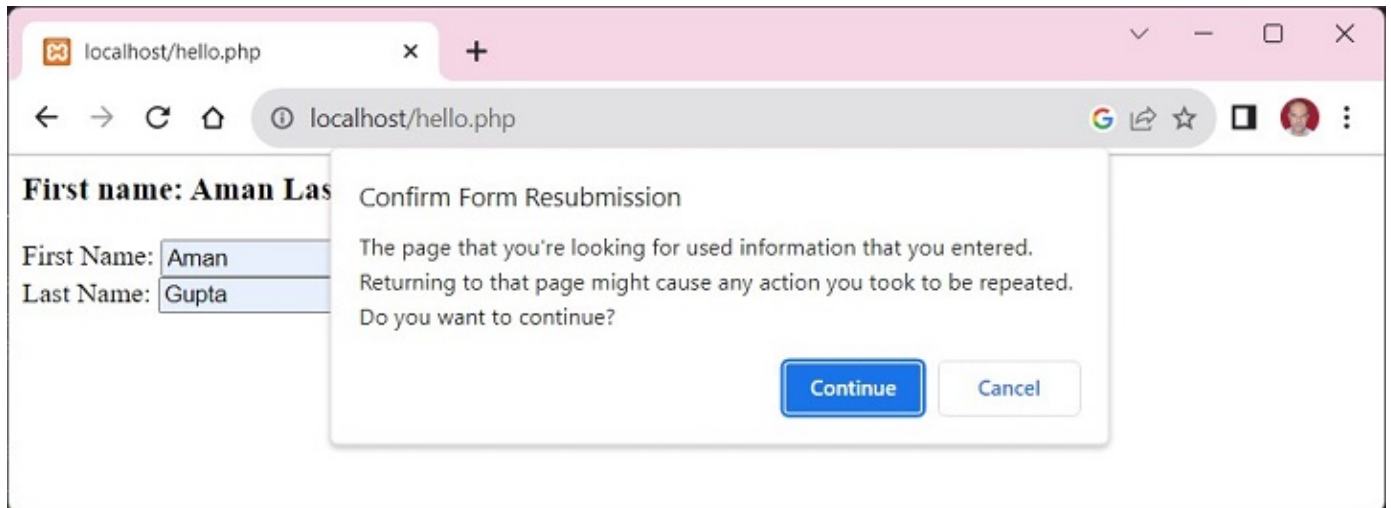
To start with, let us consider the following PHP script that renders a simple HTML form, and submits it back to itself with POST method. When the user fills the data and submits, the backend script fetches the data, renders the result, and comes back to show the blank form again.

```
<?php
    if (isset($_POST["submit"])) {
        if ($_SERVER["REQUEST_METHOD"] == "POST")
            echo "First name: " . $_REQUEST['first_name'] . " " . "Last Name: " . $_R
    }
?>

<html>
<body>
    <form action="<?php echo $_SERVER['PHP_SELF'];?>" method="post">
        First Name: <input type="text" name="first_name"> <br/>
        Last Name: <input type="text" name="last_name" />
        <button type="submit" name="submit">Submit</button>
    </form>
</body>
</html>
```

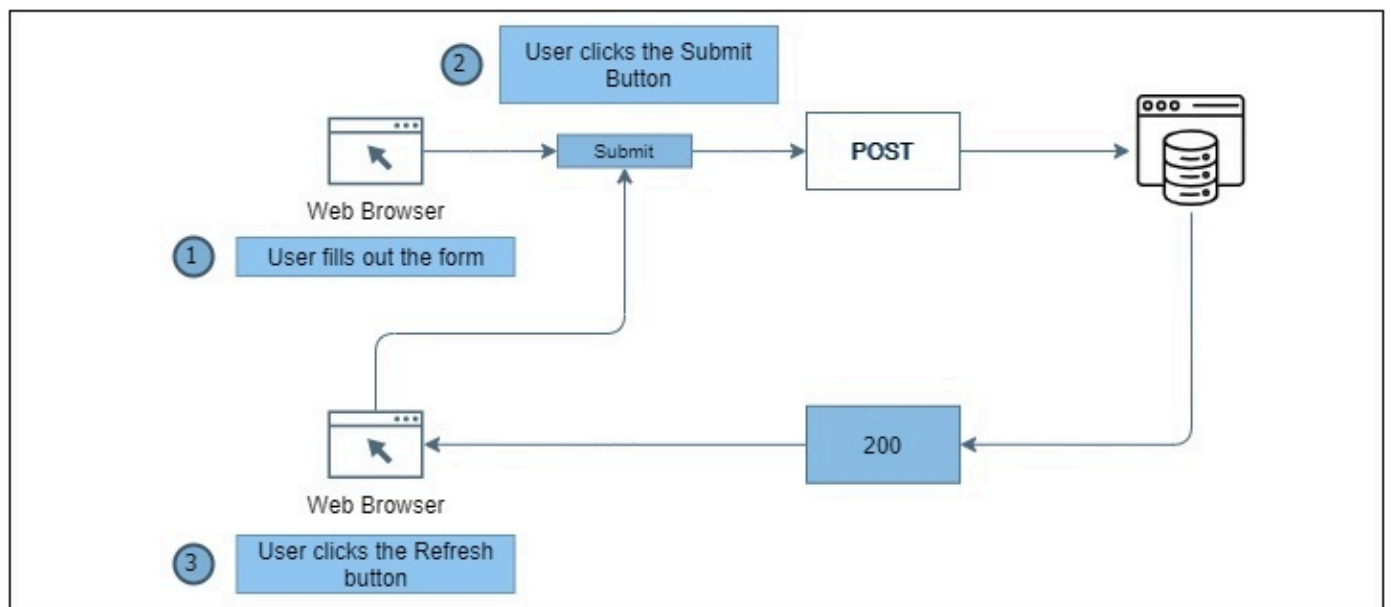
Assuming that the server is running, the above script is placed in the document root folder and visited in the browser.

Fill the data and submit. The browser echoes the result, and re-renders the form. Now if you try to refresh the browser page, a warning pops up as shown below –



If you press **Continue**, the same data is posted again.

The problem can be understood with the following figure –



Following steps are taken in the PHP script to avoid the problem –

- The PHP script before the HTML form starts a new session.
- Check if the form has been submitted with POST method.
- If so, store the form data in session variables
- Redirect the browser to a result page. In our case, it is the same page. With the exit command, to terminate this script to make sure no more code gets executed.
- If PHP finds that the REQUEST method is not POST, it checks if the session variables are set. If so, they are rendered along with the fresh copy of form.
- Now even if the form is refreshed, you have successfully averted the possibility of resubmission.

Example

Here is the PHP code that uses the PRG technique –

</>

Open Compiler

```
<?php
    session_start();
    if (isset($_POST["submit"])) {
        $_SESSION['fname'] = $_POST['first_name'];
        $_SESSION['lname'] = $_POST['last_name'];
        header("Location: hello.php");
        exit;
    }
    if (isset($_SESSION["fname"])) {
        echo "First name: " . $_SESSION['fname'] . " " . "Last Name: " . $_SESSION['lname'];
        unset($_SESSION["fname"]); unset($_SESSION["lname"]);
    }
?>

<html>
<body>
    <form action="<?php echo $_SERVER['PHP_SELF'];?>" method="post">
        First Name: <input type="text" name="first_name"> <br />
        Last Name: <input type="text" name="last_name" />
        <button type="submit" name="submit">Submit</button>
    </form>
```

```
</body>
```

```
</html>
```