



GET-запити

У мовах C, C++, Pascal вхідні дані можна було вводити в консолі.

А як же це зробити в мові PHP?

Ми ж відкриваємо сайт у браузері, а не в консолі.

У браузері теж можна задавати
вхідні дані.

Є кілька способів.

Перший спосіб: додати параметри до назви сторінки в адресному рядку браузера.



debian

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
```

localhost/mysite/index.php?
name=ivan&year=2000

Це називається GET-параметри.

А такий запит називається GET-
запитом.

При такому запиті в PHP
створиться масив `$_GET` із двома
елементами:

```
echo $_GET["name"]; // "ivan"
```

```
echo $_GET["year"]; // "2000"
```

Після 1-ї лабораторної роботи
ваш файл **index.php** має вигляд:

```
<?php  
    require_once("header.php");  
    require_once("left_menu.php");  
?>
```

Content...

```
<?php  
    require_once("footer.php");  
?>
```


Треба позбутися всього HTML-коду в `index.php` і винести content в окремий файл.

Оскільки `header.php`, `left_menu.php`
та `footer.php` однакові на всіх
сторінках, їх прийнято виносити в
окрему папку `layout`.

Те, що різне на кожній сторінці
(контент), прийнято поміщати в
папку `views`.

Тож контент головної сторінки
буде у файлі `views/main.php`.

Тепер `index.php` набуде вигляду:

```
<?php
    require_once("layout/header.php");
    require_once("layout/left_menu.php");

    require_once("views/main.php");

    require_once("layout/footer.php");
?>
```

Наступне завдання — створити сторінку “Про сайт”.

Можна було б створити новий файл
під назвою **about.php**:

```
<?php  
    require_once("layout/header.php");  
    require_once("layout/left_menu.php");  
  
    require_once("views/about.php");  
  
    require_once("layout/footer.php");  
?>
```

Але тоді в `index.php` і `about.php`
дублюється багато однакового
коду.

Тож прийнято звертатись до всіх сторінок сайту через єдиний файл — **index.php**.

Він називається точкою входу на сайт (entry point).

Як же розрізняти, яку саме сторінку на сайті хоче відкрити користувач?

Зазвичай, це прийнято визначати GET-параметром **action**:

localhost/mysite/index.php?
action=main

localhost/mysite/index.php?
action=about

Тоді в коді `index.php` перевіряємо:

```
if($_GET["action"] == "main") {  
    require_once("views/main.php");  
}  
else if($_GET["action"] == "about") {  
    require_once("views/about.php");  
}
```

Але виникає недолік.

Якщо параметр **action** не вказано

localhost/mysite/index.php

або

localhost/mysite/

Тоді масив \$_GET буде порожнім.
А при зверненні до неіснуючого
елементу масиву

```
if($_GET["action"] == "main")
```

на екрані з'явиться зауваження

```
NOTICE Undefined index: action
```

Щоб уникнути цього, спершу
будемо перевіряти, чи існує
взагалі в масиві **\$_GET** елемент
action.

Для цього є функція **isset**, яка
повертає **true**, якщо змінна чи
елемент масиву існує.

```
if(isset($_GET["action"]) &&  
$_GET["action"] == "main")
```

```
if(isset($_GET["action"]) &&
$_GET["action"] == "main") {
    require_once("views/main.php");
}
else if(isset($_GET["action"]) &&
$_GET["action"] == "about") {
    require_once("views/about.php");
}
```

Але тоді, якщо **action** не вказано, не виконається жоден **if**. На сторінці не буде контенту.

Додамо в кінці **else**:

```
if(isset($_GET["action"]) &&
    $_GET["action"] == "main") {
    require_once("views/main.php");
}
else if(isset($_GET["action"]) &&
    $_GET["action"] == "about") {
    require_once("views/about.php");
}
else {
    require_once("views/main.php");
}
```

Також цей **else** спрацює тоді, коли користувач відкриє неіснуючу сторінку.

localhost/mysite/index.php?
action=123

Але одержаний код має ще один
недолік.

Якщо потрібно додати на сайт
нову сторінку, наприклад,
action=registration,
треба вставити ще один блок
else if.

І так для кожної майбутньої
сторінки.


```
if(isset($_GET["action"]) &&
    $_GET["action"] == "main") {
    require_once("views/main.php");
}
else if(isset($_GET["action"]) &&
    $_GET["action"] == "about") {
    require_once("views/about.php");
}
else if(isset($_GET["action"]) &&
    $_GET["action"] == "registration") {
    require_once("views/registration.php");
}
else {
    require_once("views/main.php");
}
```

Такий код можна оптимізувати,
щоб він займав лише один блок
if...else:

```
if(isset($_GET["action"]) && ...) {  
    require_once(...);  
}  
else {  
    require_once("views/main.php");  
}
```

Для цього знадобиться функція `file_exists`, яка перевіряє існування файлу.

Тепер пункт меню “Головна” на сайті буде вказувати на сторінку

```
<a href="index.php?action=main">
```

А пункт “Про нас” — на сторінку

```
<a href="index.php?action=about">
```