

# Accepting User Input

## Web Programming and Testing



Mario Simaremare, S.Kom., M.Sc.

Program Studi Sarjana Sistem Informasi  
Institut Teknologi Del



# Objectives

- The objective of this session is the following:
  - The students are able to elaborate the needs of user interaction.
  - The students are able to distinguish methods to interact with the user.

# Outlines

1. Evolution of web application.
2. Dynamic web.
3. Accepting user input with GET and POST methods.

# Evolution of Web Apps

# Evolution of Web Apps

- Web 1.0:
  - The web is static with push direction.
  - HTML + CSS + JavaScript
- Web 2.0:
  - The web turns into a dynamic space where the user can contribute.
  - Web 1.0 + server-side processing engine + DB.
- Web 3.0
  - The web becomes smarter with suggestion, forecasting, etc.
  - Web 2.0 + AI and the likes.



# Web 2.0 Is Minimal

- In some cases, 1.0 is still there.
  - For documentation, static wiki pages, etc.
- Today's web apps should be at the 2.0 state (minimal).
  - Blogs, official websites, etc.
- Reaching 3.0 would be great.
  - Youtube, Facebook, E-Commerce, etc.



**Key: its users are the content creator.**

# Dynamic Web Application

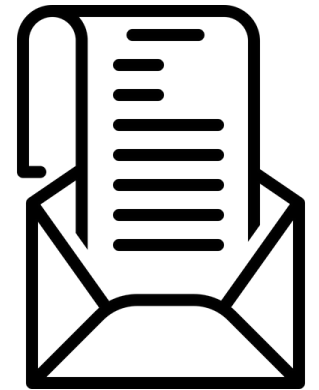
# Dynamic Web Application

- Dynamic web means:
  - Two-way communication.
    - The web provides information that it stores.
    - The user contributes new information to the web apps.
  - Key: user interaction
- The web apps get richer of information.



# Interacting With User

- The question is, how do web apps interact with its users?
  - Or how the user could send information to the web app?
- The HTTP provides several methods:
  - GET, DELETE, POST, PUT, PATCH.
  - HEAD and OPTIONS are not very common in this scenario.
- Requirement to enable user interaction:
  - A backend engine that reads and process the input accordingly.



# GET and POST Methods

- Interactions could be categorized as:
  - Reading or retrieving interaction.
    - Reading must not affect the resource states (idempotent).
    - When the web apps users are asking for a resource.
    - In this case, GET method is used.
  - Writing interaction.
    - When the web apps users are keen to create, modify, or remove a resource.
    - In this case, POST method is used.

**Why PUT, PATCH, and DELETE methods are not used?**

# Retrieving with the GET Method

# GET Method

- GET method is aimed to be used for reading a resource.
  - Since resource retrieval has to be **idempotent**, hence it is qualified to be shareable.
  - Since it is qualified to be shareable, browsers tend to display GET request at the address bar.
- Parameters attached to an HTTP request with GET method are written as in the URL query.

# An HTTP Request with GET Method

```
http://example.com/get.php?param1=tra&param2=lala&param3=parampampam
```

# GET Request Distilled

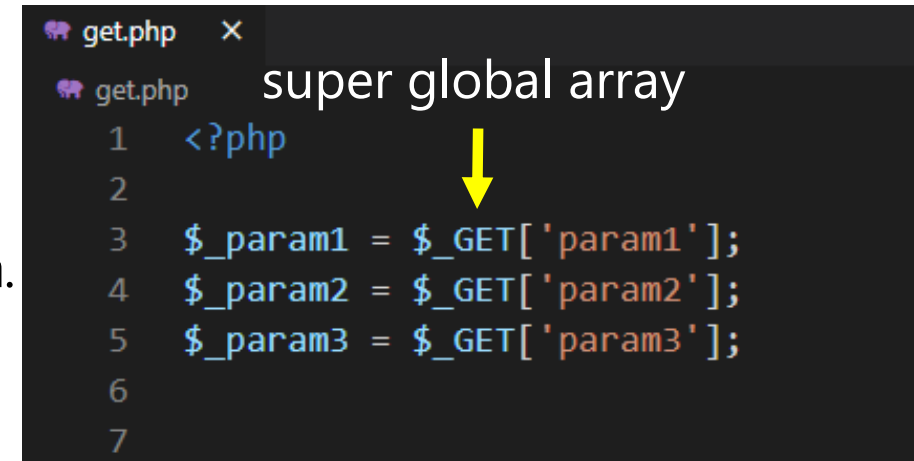
```
http://example.com/get.php?param1=tra&param2=lala&param3=parampampam
```

- The URL consists of:
  - Protocol scheme: http
  - Domain name: example.com
  - Port: 80 (implicit) `http://example.com:88/` ← explicit
  - Path: /get.php
  - Query: param1=tra&param2=lala&param3=parampampam
    - 1<sup>st</sup> parameter named param1 with value tra.
    - 2<sup>nd</sup> parameter named param2 with value lala.
    - 3<sup>rd</sup> parameter named param3 with value parampampam.

# Reading GET Parameters

`http://example.com/get.php?param1=tra&param2=lala&param3=parampampam`

- Let's focus on the query part.
  - 1<sup>st</sup> parameter named param1 with value tra.
  - 2<sup>nd</sup> parameter named param2 with value lala.
  - 3<sup>rd</sup> parameter named param3 with value parampampam.
- In PHP, parameters sent through GET method are accessible via `$_GET` super global array.



```
get.php x
get.php
1 <?php
2
3 $_param1 = $_GET['param1'];
4 $_param2 = $_GET['param2'];
5 $_param3 = $_GET['param3'];
6
7
```

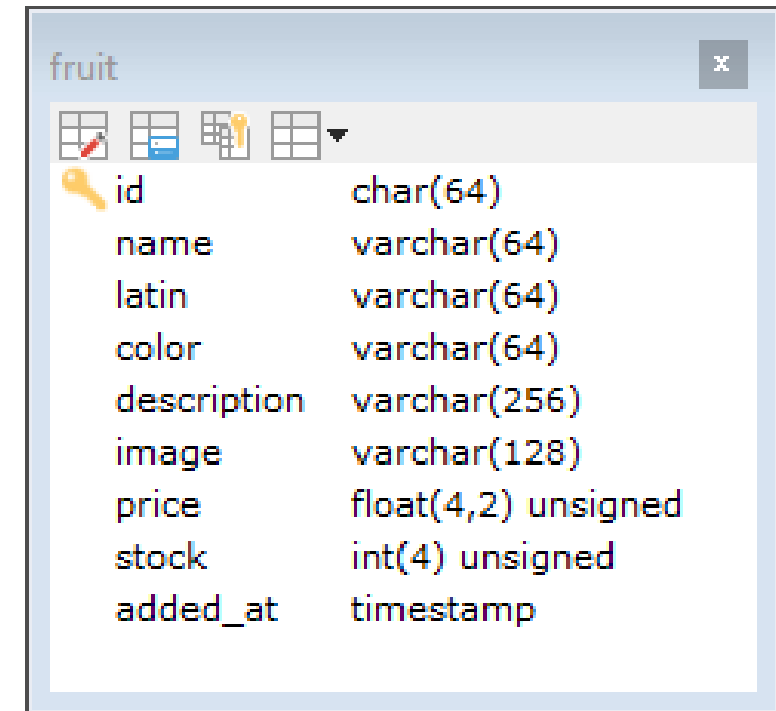
super global array

# GET Method: Getting Dirty



# Case Study: Simple Fruit Shop

- There is an online fruit shop accessible through the Internet.
  - It has information about fruits sold through the website.
  - The information about a fruit is a resource.
- A fruit has some attributes:
  - id, name, latin, color, description, image, price, stock, and added\_at.
  - A fruit is identified by an id.



A screenshot of a database table named 'fruit'. The table has the following attributes and data types:

id	char(64)
name	varchar(64)
latin	varchar(64)
color	varchar(64)
description	varchar(256)
image	varchar(128)
price	float(4,2) unsigned
stock	int(4) unsigned
added_at	timestamp

# Case Study: Simple Fruit Shop

- What we are going to see:
  - How to retrieve a fruit by its identifier field.
  - Display all available fruits (resources).
  - Create links.

# Modifying A Resource with the POST Method

# POST Method

- POST method is aimed to be used for creating a new resource.
  - But, in most situation it is also used to modify or even removing an existing resource.
  - Since resource modification and removal affect the resource state, hence it is **not** qualified to be shareable.
  - Since it is **not** qualified to be shareable, browsers tend to hides POST request.
- Parameters attached to an HTTP request with POST method are unseen by the user.

# Remember!

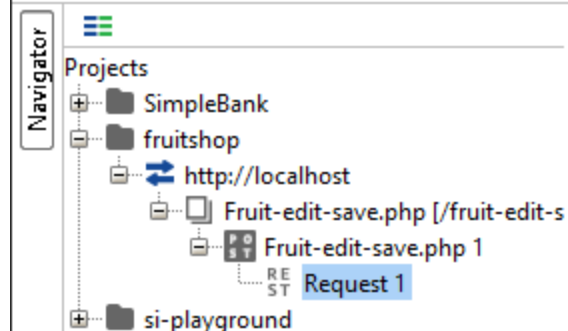
**Creation is only for inexistent resource!**

**Retrieval, modification, and removal actions are only for existing resource!**

# POST Method: Getting Dirty

# Case Study: Simple Fruit Shop

- What we are going to see:
  - Create a form to modify resource data (price).
  - How to retrieve a fruit (resource) by its identifier field and modify it.
  - Display the modified fruits (resources).



## Request 1

Method POST Endpoint http://localhost Resource /fruit-edit-save.php Parameters

Name	Value	Style	Level
id	0e21b9e56243d2915b...	QUERY	RESOURCE
price	4.40	QUERY	RESOURCE

Required: ☐ Sets if parameter is required

Type:

Media Type multipart/form-data ☒ Post QueryString

Auth Headers (0) Attachments (0) Representations (2) JMS Headers JMS Property (0)

response time: 2018ms (448 bytes)

XML

```
<html>
<head>
  <meta content="HTML Tidy for Java (vers. 27
  <title/>
</head>
<body>
  <pre>Array
(
  [0] => Array
(
    [id] => 0e21b9e56243d2915bd56d09034cec
    [name] => Grape
    [latin] => Vitis vinifera
    [color] => Green, Purple, Red, &amp; W
    [description] => Very delicious.
    [image] => images/grape.jpg
    [price] => 4.40
    [stock] => 20
    [added_at] => 2018-09-09 20:24:13
  )
)
</pre>
</body>
</html>
```

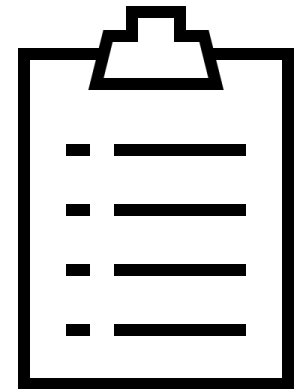
Heade... Attachme... SSL ... Representati... Schema (co... JM...

11 : 8



# To-dos

1. Can we create a link with POST method?
2. Why do we use POST for modification?  
Shouldn't it only be used for resource creation?
3. Is it possible to create a use GET, PUT or PATCH in the form's method attribute?



# References

Srinivasan, M. (2012). Web Technology: Theory and Practice. Pearson.

Mozilla. HTML Element Reference.

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

Thank  
you

