Web APIs – Introduction

API adalah singkatan dari Application Programming Interface.

API Web adalah antarmuka pemrograman aplikasi untuk Web.

API Browser dapat memperluas fungsionalitas browser web.

API Server dapat memperluas fungsionalitas server web.

Semua browser memiliki sekumpulan API Web bawaan untuk mendukung operasi yang kompleks, dan untuk membantu mengakses data. Misalnya, Geolocation API bisa mengembalikan koordinat tempat browser berada.

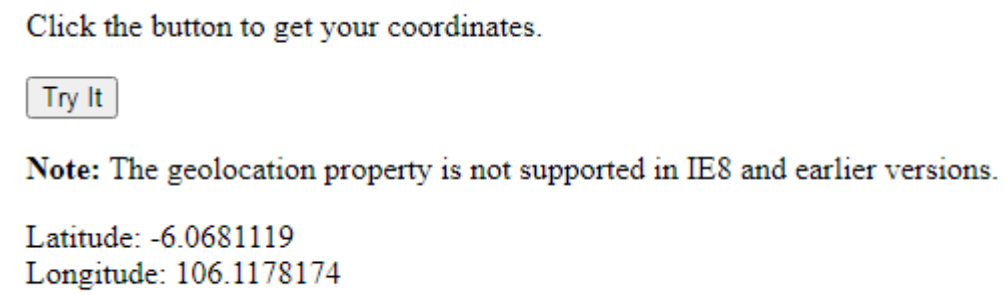
Contoh

Dapatkan lintang dan bujur dari posisi pengguna:

```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to get your coordinates.</p>

<button onclick="getLocation()">Try It</button>

<p><strong>Note:</strong> The geolocation property is not supported in IE8 and earlier versions.</p>
```

```
<p id="demo"></p>

<script>
var x = document.getElementById("demo");

function getLocation() {
  if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this
browser.";
  }
}

function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude
+
  "<br>Longitude: " + position.coords.longitude;
}
</script>
```

```
</body>

</html>
```



Running Program 1

PHP and JSON

What is JSON?

JSON adalah singkatan dari JavaScript Object Notation, dan merupakan sintaks untuk menyimpan dan bertukar data.

Karena format JSON adalah format berbasis teks, ini dapat dengan mudah dikirim ke dan dari server, dan digunakan sebagai format data oleh bahasa pemrograman apa pun.

PHP memiliki beberapa fungsi built-in untuk menangani JSON.

Pertama, kita akan melihat dua fungsi berikut:

- json_encode()
- json_decode()

PHP - json_encode()

Fungsi json_encode () digunakan untuk menyandikan nilai ke format JSON.

Contoh

```
<!DOCTYPE html>
<html>
<body>

<?php
$age = array("Peter"=>35, "Ben"=>37, "Joe"=>43);

echo json_encode($age);
?>

</body>
</html>
```

Hasil Runing

```
{"Peter":35,"Ben":37,"Joe":43}
```

PHP - json_decode()

Fungsi json_decode () digunakan untuk mendekode objek JSON menjadi objek PHP atau array asosiatif.

```
<!DOCTYPE html>
<html>
<body>

<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

var_dump(json_decode($jsonobj));
?>

</body>
</html>
```

Hasil Runing

```
object(stdClass)#1 (3) { ["Peter"]=> int(35) ["Ben"]=> int(37) ["Joe"]=> int(43) }
```

Fungsi json_decode () mengembalikan objek secara default. Fungsi json_decode () memiliki parameter kedua, dan jika disetel ke true, objek JSON didekodekan menjadi array asosiatif.

```
<!DOCTYPE html>
<html>
<body>

<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

var_dump(json_decode($jsonobj, true));
```

```
?>


</body>
</html>
```

Hasil Running

```
array(3) { ["Peter"]=> int(35) ["Ben"]=> int(37) ["Joe"]=> int(43) }
```

PHP - Looping Through the Values

Anda juga bisa mengulang nilai-nilai dengan foreach () loop:

```
<!DOCTYPE html>
<html>
<body>

<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

$arr = json_decode($jsonobj, true);

foreach($arr as $key => $value) {
  echo $key . " => " . $value . "<br>";
}
?>

</body>
</html>
```

Hasil Running

```
Peter => 35
Ben => 37
Joe => 43
```

PHP MySQL Prepared Statements

Prepared Statements and Bound Parameters

Pernyataan yang disiapkan adalah fitur yang digunakan untuk menjalankan pernyataan SQL yang sama (atau serupa) berulang kali dengan efisiensi tinggi.

Pernyataan yang disiapkan pada dasarnya bekerja seperti ini:

1. Mempersiapkan: Templat pernyataan SQL dibuat dan dikirim ke database. Nilai tertentu dibiarkan tidak ditentukan, disebut parameter (berlabel "?"). Contoh: INSERT INTO MyGuests VALUES (?,?,?)
2. Database mem-parsing, mengompilasi, dan melakukan pengoptimalan kueri pada template pernyataan SQL, dan menyimpan hasilnya tanpa menjalankannya
3. Jalankan: Di lain waktu, aplikasi mengikat nilai ke parameter, dan database menjalankan pernyataan tersebut. Aplikasi dapat mengeksekusi pernyataan tersebut sebanyak yang diinginkannya dengan nilai yang berbeda

Dibandingkan dengan menjalankan pernyataan SQL secara langsung, pernyataan yang disiapkan memiliki tiga keunggulan utama:

1. Pernyataan yang disiapkan mengurangi waktu penguraian karena persiapan pada kueri dilakukan hanya sekali (meskipun pernyataan tersebut dijalankan beberapa kali)
2. Parameter terikat meminimalkan bandwidth ke server karena Anda hanya perlu mengirim parameter setiap kali, dan bukan seluruh kueri
3. Pernyataan yang disiapkan sangat berguna terhadap injeksi SQL, karena nilai parameter, yang dikirim nanti menggunakan protokol berbeda, tidak perlu di-escape dengan benar. Jika templat pernyataan asli tidak berasal dari input eksternal, injeksi SQL tidak dapat terjadi.

Prepared Statements in MySQLi

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

// prepare and bind
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email) VALUES (?, ?, ?)");
$stmt->bind_param("sss", $firstname, $lastname, $email);

// set parameters and execute
$firstname = "John";
$lastname = "Doe";
```

```
$email = "john@example.com";

$stmt->execute();



$firstname = "Mary";

$lastname = "Moe";

$email = "mary@example.com";

$stmt->execute();



$firstname = "Julie";

$lastname = "Dooley";

$email = "julie@example.com";

$stmt->execute();



echo "New records created successfully";



$stmt->close();

$conn->close();

?>
```

Penjelasan

```
"INSERT INTO MyGuests (firstname, lastname, email) VALUES
(?, ?, ?)"
```

Dalam SQL kami, kami memasukkan tanda tanya (?) Di mana kami ingin mengganti nilai integer, string, double atau blob.

Kemudian, lihat fungsi bind_param ():

```
$stmt->bind_param("sss", $firstname, $lastname, $email);
```

Fungsi ini mengikat parameter ke kueri SQL dan memberi tahu database apa saja parameternya. Argumen "sss" mencantumkan jenis data yang menjadi parameternya. Karakter s memberitahu mysql bahwa parameternya adalah string.

Argumennya mungkin salah satu dari empat jenis:

i - integer

d - ganda

s - string

b - BLOB

Kita harus memiliki salah satunya untuk setiap parameter.

Dengan memberi tahu mysql jenis data apa yang diharapkan, kami meminimalkan risiko injeksi SQL.