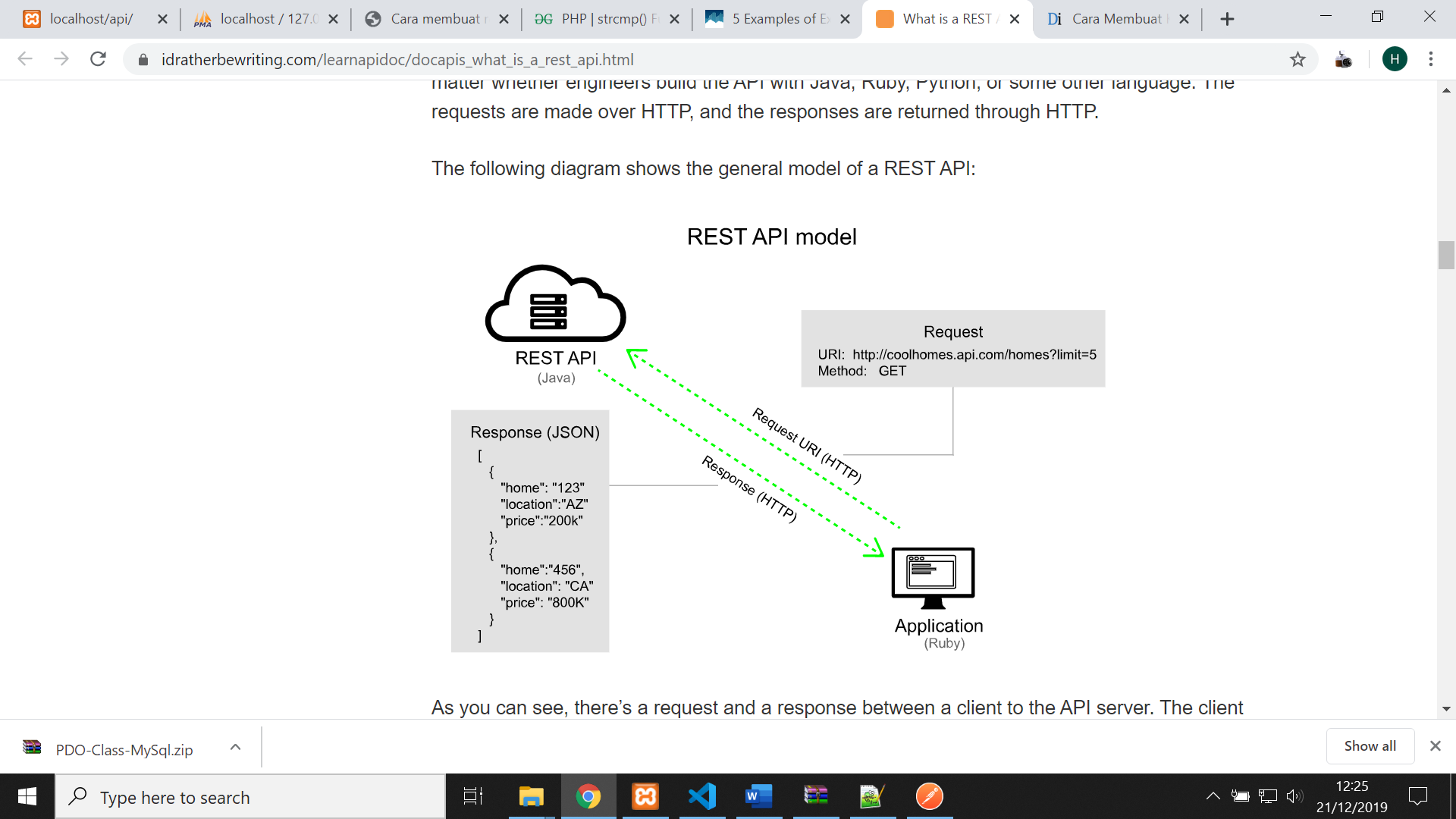
API

* Function
* Protocols
* Tools

konsep:



Api merupakan konsep, dimana kita bisa membuatnya dengan Java (seperti yang di atas), php, dll. Dalam contoh di atas kita membuat request ke applikasi Api(menggunakan Java) dan terakhir responsenya adalah json.

Open/Public API

Open API adalah sebuah API yang terbuka, maksudnya bisa di akses dari tempat lain.

Namun tetap ada proses otentikasi:

1. Tanpa otentikasi
2. Key/ token
3. OAuth (login ke aplikasi lain menggunakan fb/goole)
4. CORS (apakah aplikasi API nya terbuka)

JSON

Untuk membuat/menyediakan data dengan type json, maka extension dari filenya adalah .json

contoh format JSON

{

“nama” : “Henro”,

“alamat” : “Jakarta”

}

Itu adalah contoh simple dari JSON.

Format penulisan JSON adalah seperti berikut ini:

1. Tanda parentesis tutup dan buka : {}
2. Key : yang di sebelah kiri block dengan tanda petik 2 , tidak boleh petik 1
3. Value di sebelah kanan, kalau string di beri petik kalau number tidak usah.

Ada beberapa 6 type data yang bisa ditampung sebagai value dari JSON.

Value itu adalah:

1. Strings
2. Number
3. Object
4. Arrays
5. Booleans (true atau false)
6. Null

Contoh di atas hanya merupakan contoh json simple. Berikut contoh object bersarang type json.

{

"sammy" : {

"username" : "SammyShark",

"location" : "Indian Ocean",

"online" : true,

"followers" : 987

},

"jesse" : {

"username" : "JesseOctopus",

"location" : "Pacific Ocean",

"online" : false,

"followers" : 432

},

}

Kalau kita perhatikan paling luar ada tanda parentesis pembuka dan penutup { dan }.

Objectnya adalah 1 level di bawahnya, yaitu sammy dan jesse. Setelah sammy dan jesse ada lagi parentesis pembuka dan penutup dan isinya yaitu :

{

"username" : "SammyShark",

"location" : "Indian Ocean",

"online" : true,

"followers" : 987

},

Didalam object itu juga kita bisa buat object lagi. Jadi object dalam object sebanyak yang kita mau.

Contoh:

{

"sammy" : {

"username" : "SammyShark",

"location" : "Indian Ocean",

"online" : true,

"followers" : 987

"website" : {

"description": "belajar",

"url": "http://belajar.com"

}

{

"description": "komedi",

"url": "http://komedi.com"

}

},

}

Terus bagaimana cara membuat arrays bersarang? Caranya kita cukup hanya menambahkan tanda kotak pembuka [ dan penutup ] sebelum tanda parentesisnya.

Contoh:

{

"sammy" : {

"username" : "SammyShark",

"location" : "Indian Ocean",

"online" : true,

"followers" : 987

"website" : [{

"description": "belajar",

"url": "http://belajar.com"

}

{

"description": "komedi",

"url": "http://komedi.com"

}]

},

}

Cara memanggil/membaca file ber ekstensi .json dari php.

Untuk bisa membaca file dengan extensi json kedalam php adalah dengan menggunakan :

File\_get\_contents.

Contoh:

$json\_text = file\_get\_contents('user\_profile.json');

Jika berhasil memanggil file .json nya sekarang semua datanya sudah ada di variable $json\_text dalam bentuk text. Sekarang kita perlu mengubah/convert typenya ke dalam object maupun array.

Caranya adalah dengan menggunakan key: json\_decode ini merupakan kebalikan dari json encode.

Contoh:

$object = json\_decode($json\_text);

  print\_r ($object->first\_name);

Kenapa kita perlu merubahnya, karena kalau type text untuk memaggil yang kita perlukan akan sangat susah. Seperti yang kita lihat di atas kita bisa memanggil first\_name dengan cara :

$object->first\_name

Tapi kalau dalam bentuk text itu tidak bisa dilakukan.

Untuk mengkonvert json ke dalam array adalah :

$object = json\_decode($json\_text, true);

Seperti yang kita lihat, kita hanya menambah argument true setelah variable jsonnya.

Note :file\_get\_contents() kita juga bisa gunakan untuk memanggil function di dalam object.

Contoh di codeighniter:

file\_get\_contents("http://10.10.10.29:8007/intsys/copy\_file\_tmli");

Data Json dari file luar kita ambil dengan php dengan tujuan akan kita simpan atau olah ke database. Sekarang kita akan mengerjakan kebalikannya. Dimana kita akan mengambil dari dari database dan ditampilkan ke luar/user/client. Dalam hal ini , kita akan menampilkannya dalam bentuk json juga. Untuk menampilkan ke dalam json kita bisa menggunakan bawaan php **json\_encode.**

Contoh:

Untuk sementara datanya di buat langsung hardcode di dalam php. Berikut adalah data array assosiative yang di buat di dalam php.

// Data 1

$data [] = array(

'no' => 1,

'nama' => 'Rudi',

'alamat' => 'Nganjuk'

);

// Data 2

$data [] = array(

'no' => 2,

'nama' => 'Indah',

'alamat' => 'Madiun'

);

Kemudian kita encode data array di atas kedalam type json.

// Mengencode data menjadi json

$jsonfile = json\_encode($data, JSON\_PRETTY\_PRINT);

Untuk menggenarte 1 file .json secara otomatis kita bisa menggunakan : fungsi php yaitu:

file\_put\_contents. Fungsin akan otomati menambah file jiga file dengan nama tersebut belum ada.

Contoh: **file\_put\_contents**('anggota.json', $jsonfile);

Cara mengedit/ menambah data dari file anggota.json yang kita tercreate otomatis itu:

Flownya:

1. Panggil data file dengan file\_get\_contents()
2. Ubah ke dalam bentuk array dengan json\_decode.
3. Tambah data ke dalam array datanya.

Perhatikan code di bawah ini

**<?php**

// File json yang akan dibaca

$file = "anggota.json";

// Mendapatkan file json

$anggota = file\_get\_contents($file);

// Mendecode anggota.json

$data = json\_decode($anggota, true);

// Data array baru

$data [] = array(

'no' => 3,

'nama' => 'Elika',

'alamat' => 'Kediri'

);

// Mengencode data menjadi json

$jsonfile = json\_encode($data, JSON\_PRETTY\_PRINT);

// Menyimpan data ke dalam anggota.json

$anggota = file\_put\_contents($file, $jsonfile);

Mengedit data yang ada di array:

**<?php**

// File json yang akan dibaca

$file = "anggota.json";

// Mendapatkan file json

$anggota = file\_get\_contents($file);

// Mendecode anggota.json

$data = json\_decode($anggota, true);

// Membaca data array menggunakan foreach

foreach ($data as $key => $d) {

// Perbarui data kedua

if ($d['no'] === 2) {

$data[$key]['alamat'] = 'Surabaya';

}

}

// Mengencode data menjadi json

$jsonfile = json\_encode($data, JSON\_PRETTY\_PRINT);

// Menyimpan data ke dalam anggota.json

$anggota = file\_put\_contents($file, $jsonfile);

Mengkonvert JSON string ke object javascript.

Javascript menyediakan fungsi berikut:

JSON.parse()

Contoh:

<script>

// Convert JSON String to JavaScript Object

var JSONString = '[{"name":"Jonathan Suh","gender":"male"},{"name":"William Philbin","gender":"male"},{"name":"Allison McKinnery","gender":"female"}]';

var JSONObject = JSON.parse(JSONString);

console.log(JSONObject); // Dump all data of the Object in the console

alert(JSONObject[0]["name"]); // Access Object data

</script>

Penjelasan:

Kita membuat data json hardcode:

var JSONString = '[{"name":"Jonathan Suh","gender":"male"},{"name":"William Philbin","gender":"male"},{"name":"Allison McKinnery","gender":"female"}]';

kemudian kita convert ke dalam object javascript

var JSONObject = JSON.parse(JSONString);

di browser internet expolorer 7 , JSON.parse tidak support. Tapi kita bisa atasi dengan fungsi jquery yaitu: $.parseJSON()

contoh:

var JSONObject = $.parseJSON(JSONString);

untuk mengolah data object javascriptnya, kitab bisa mengolah seperti loop berikut ini:

<script>

// Loop through Object

var JSONObject = ...; // Replace ... with your JavaScript Object

for (var key in JSONObject) {

if (JSONObject.hasOwnProperty(key)) {

console.log(JSONObject[key]["name"] + ", " + JSONObject[key]["gender"]);

}

}

</script>

Keblikan dari json.parse adalah JSON.stringify

Tujuannya adalah mengembalikan/mengkonvert javascript object ke dalam JSON String.