

Backend Selektion=7;

Utbildare: Mahmud Al Hakim

NACKADEMIN

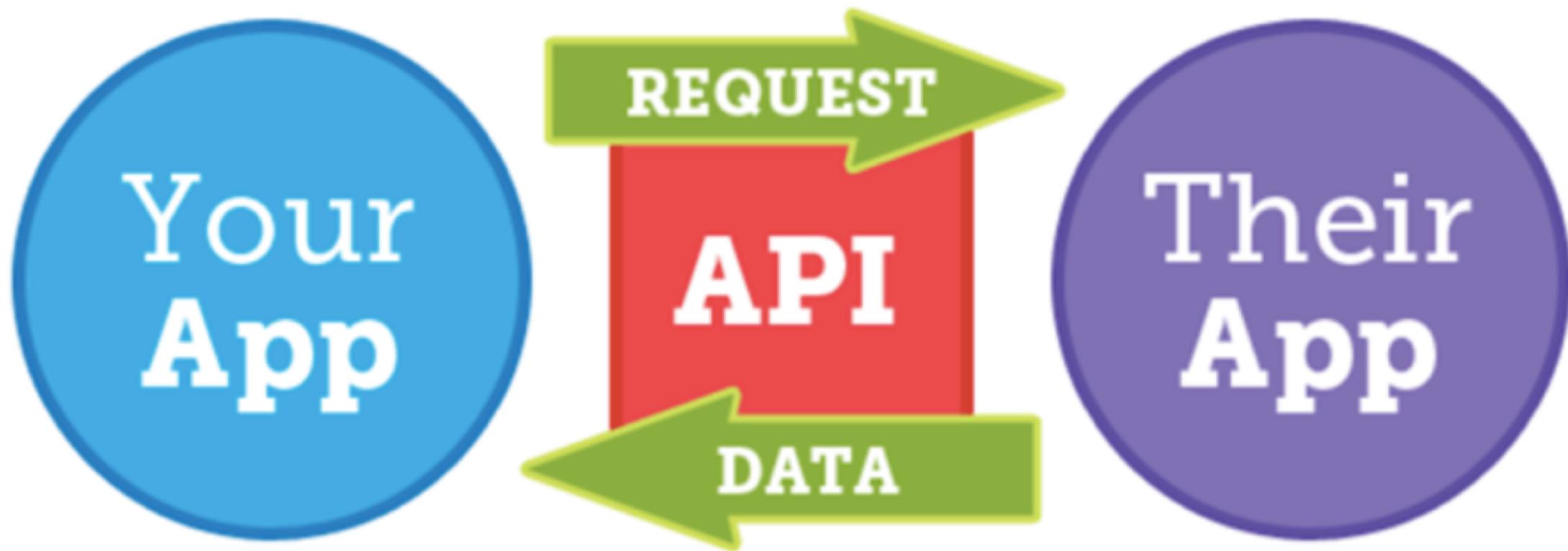
Lektionstillfällets mål

- **API:er del 1 av 2**
- Introduktion till API
- RESTful API:er
- JSON
- Arbeta mot befintliga servrar

Kort summering av föregående lektion

- Vi har gått igenom applikationsutveckling med PHP och MySQL

Vad är API?



API – Application Programming Interface

- De flesta programvaror i dagens läge är applikationer som knyter samman annan mjukvarufunktion i olika former och skapar en meningsfull helhet, och denna sammanknytning sker med hjälp av API:er.
- API:er ger alltså möjlighet att på ett strukturerat sätt återanvända redan utvecklad och kvalitetssäkrad mjukvara som har kapslats in i någon form av kodbibliotek.
- Källa:
https://sv.wikipedia.org/wiki/Application_Programming_Interface

Webbtjänster

- Webbtjänster (web services) betecknar webbaserade datorprogram som kommunicerar och samarbetar dynamiskt med andra webbtjänster på samma vis som en människa kan surfa till olika webbsidor.
- För att skicka data mellan applikationerna används protokollet HTTP.
- För att beskriva applikationernas gränssnitt används XML.
- Källa: https://sv.wikipedia.org/wiki/Web_service

REST – Representational State Transfer

- REST eller RESTful webbtjänst är ett IT-arkitekturbegrepp som beskriver hur tjänster för maskin-till-maskin-kommunikation kan tillhandahållas via webbteknologi.
- Tillämpningsexempel kan vara att en mobil applikation, en bot eller en webbserver som fungerar som agent kan accessa innehållet i en databas via en webbplats som är designad för maskiner snarare än för mänskor.
- Vanligen överförs data på JSON-format mellan maskinerna.
- Källa: https://sv.wikipedia.org/wiki/Representational_State_Transfer

XML – EXTENSIBLE MARKUP LANGUAGE

- XML, är ett universellt och utbyggbart märkspråk och en förenklad efterträdare till SGML "Standard Generalized Markup Language".
- Syftet med XML är att kunna utväxla data mellan olika informationssystem.
- För att kunna hantera ett XML-dokument, krävs att sändare och mottagare av informationen har kommit överens om vilka element och attribut som ska kunna användas. Därför måste man ha en gemensam dokumentmall.
- Källa: <https://sv.wikipedia.org/wiki/XML>

Exportera data som XML via phpmyadmin

The screenshot shows the phpMyAdmin interface with the following elements:

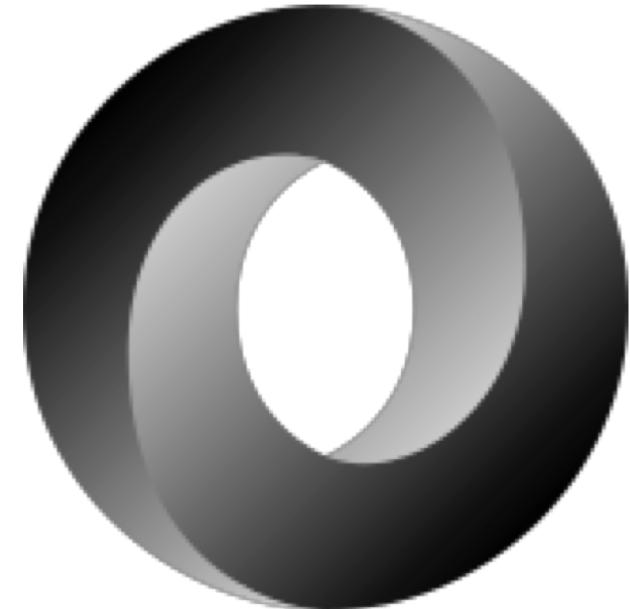
- Top navigation bar:** Bläddra, Struktur, SQL, Sök, Lägg till, Exportera.
- Section: Exportmetod:**
 - Snabb - visa endast minimala alternativ
 - Anpassad - visa alla möjliga alternativ
- Section: Format:**
 - XML (selected)
- Bottom button:** Kör

XML – Exempel

```
<database name="contactdb">
    <!-- Tabell contacts -->
    <table name="contacts">
        <column name="id">28</column>
        <column name="name">Mahmud Al Hakim</column>
        <column name="tel">08-1234567</column>
    </table>
</database>
```

JSON

- JSON står för JavaScript Object Notation.
- JSON är ett kompakt, textbaserat format som används för att utbyta data.
- JSON är ett mindre utrymmeskrävande alternativ till XML.
- OBS! JSON är enbart ett format för data. JSON säger inget om hur data ska presenteras för användaren.
- <https://www.json.org/>



Exportera data som JSON via phpmyadmin

The screenshot shows the phpMyAdmin interface for exporting data from the 'kontakt' table in the 'telefonlista' database on 'Server: 127.0.0.1'. The 'Format:' dropdown is set to 'JSON'. A 'Kör' (Run) button is visible at the bottom left.

→ Server: 127.0.0.1 » Databas: telefonlista » Tabell: kontakt

Bläddra Struktur SQL Sök Lägg till Exportera

Format:

JSON ▾

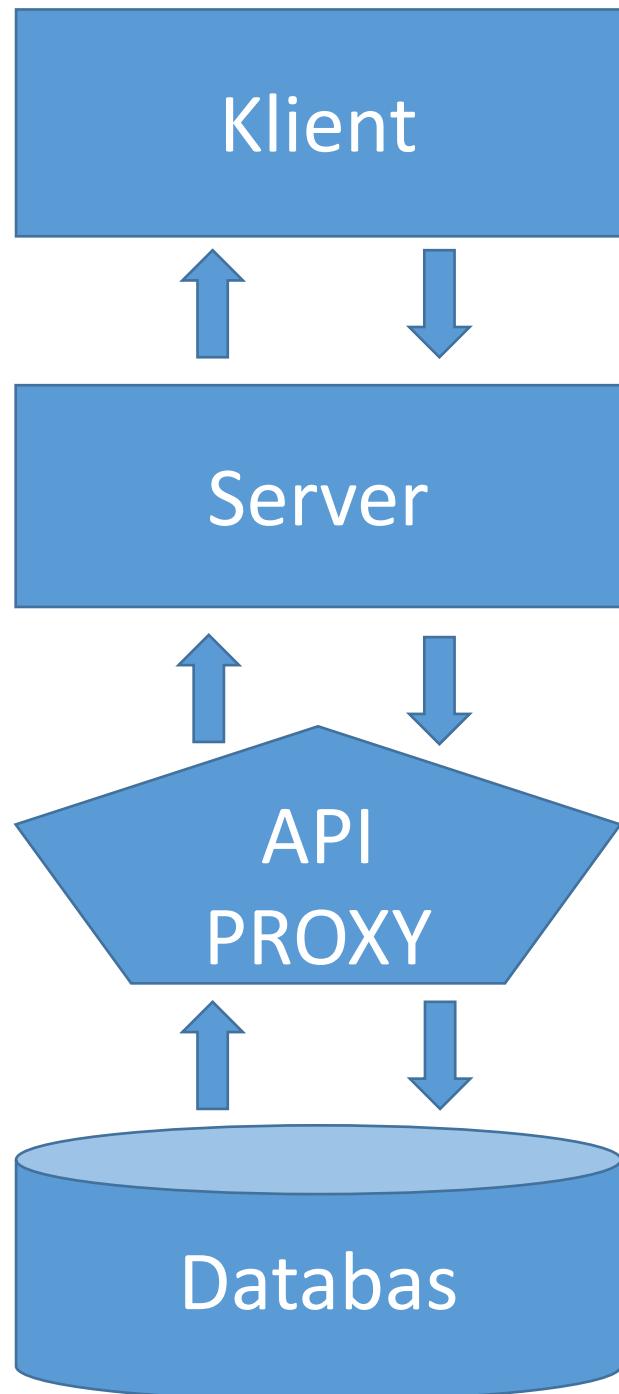
Kör

JSON – Exempel

```
{  
    "type": "table",  
    "name": "contacts",  
    "database": "contactdb",  
    "data":  
        [  
            {  
                "id": "28",  
                "name": "Mahmud Al Hakim",  
                "tel": "08-1234567"  
            }  
        ]  
}
```

API Proxy

- API Proxy är ett mellanliggande system som ligger mellan vår server/applikation och databas.
- Med hjälp av en API Proxy kan utvecklare arbeta med data utan att behöva förstå hur bakomliggande logiken fungerar.
- Vi behöver heller inte veta hur och var data lagras.



Arbeta med REST API

<https://jsonplaceholder.typicode.com>

JSONPlaceholder

Fake Online REST API for Testing and Prototyping

Serving ~200M requests per month

Resources (endpoints)

- <https://jsonplaceholder.typicode.com/posts>
- <https://jsonplaceholder.typicode.com/comments>
- <https://jsonplaceholder.typicode.com/albums>
- <https://jsonplaceholder.typicode.com/photos>
- <https://jsonplaceholder.typicode.com/todos>
- <https://jsonplaceholder.typicode.com/users>

Arbata med en befintlig tjänsteserver – Exempel 1

Hämta alla användare från JSONPlaceholder

```
<?php  
// Ange en endpoint  
$url = "https://jsonplaceholder.typicode.com/users";  
  
// Hämta data  
$json = file_get_contents($url);  
  
// Skriv ut data  
echo "<pre>$json</pre>";
```

Konvertera JSON till en PHP-Array

```
// json_decode()  
// Takes a JSON encoded string and converts it into a PHP variable.  
// http://php.net/manual/en/function.json-decode.php  
  
$array = json_decode($json, true);  
// true skapar en associativ array  
  
  
echo "<pre>";  
print_r($array);  
echo "</pre>";
```

Välj data att skriva ut från arrayen

```
foreach ($array as $key => $value) {  
    echo $value['name'] . '<br>';  
}
```

Skriv ut en tabell över namn och e-post

```
$table = "<table border=1>";  
$table .= "<tr><th>Name</th><th>Email</th></tr>";  
foreach ($array as $key => $value) {  
    $table .= "<tr>  
        <td> $value[name] </td>  
        <td> $value[email] </td>  
    </tr>";  
}  
$table .= "</table>";  
  
echo $table;
```

Övning – Skriv ut adress-Etiketter

Leanne Graham

Kulas Light
Apt. 556
92998-3874
Gwenborough

Ervin Howell

Victor Plains
Suite 879
90566-7771
Wisokyburgh

Clementine Bauch

Douglas Extension
Suite 847
59590-4157
McKenziehaven

Patricia Lebsack

Hoeger Mall
Apt. 692
53919-4257
South Elvis

Chelsey Dietrich

Skiles Walks
Suite 351
33263
Roscoeview

Mrs. Dennis Schulist

Norberto Crossing
Apt. 950
23505-1337
South Christy

Lösningsförslag

```
<style>
div{ float:left;padding:10px; margin:5px; width:200px; border:1px double grey; }
</style>

foreach ($array as $key => $value) {
    echo "<div>";
    echo "<h4>" . $value['name'] . "</h4>";
    echo $value['address']['street'] . "<br>";
    echo $value['address']['suite'] . "<br>";
    echo $value['address']['zipcode'] . "<br>";
    echo $value['address']['city'] . "<br>";
    echo "</div>";
}
```

Arbeta med en befintlig tjänsteserver – Exempel 2

Hämta bilder från JSONPlaceholder

```
<?php  
// Ange en endpoint  
$url = "https://jsonplaceholder.typicode.com/albums/1/photos";  
  
// Hämta data  
$json = file_get_contents($url);  
  
// Skriv ut data  
echo "<pre>$json </pre>";
```

Visa de första 10 bilderna

```
$array = json_decode($data, true);

foreach ($array as $key => $value) {
    if ($key == 9) break;
    echo "<a href='".$value[url]' target='_blank'>";
    echo "<img src='".$value[thumbnailUrl].'">";
    echo "</a>";
}
```

NamnAPI.se

[Varför](#)[Hur man använder NamnAPI](#)[Exempel](#)[Om](#)[Kommentarer](#)

v1 >

Like 15

Share

Tweeta

Varför

Olika namnexempel är ofta förekommande i exempelvis programmeringslitteratur, men de stannar oftast som ett exempel fast än man så ofta behöver dem i olika sorters applikationer eller också annat som är "offline".

Inte allt för sällan när man exempelvis ska göra en databas så behöver man fylla den med exempelkunder för att testa, extremt tråkigt är det att sitta och komma på ett massor med namn som man behöver föra in manuellt.

NamnAPI råder bot på de här problemen.

I NamnAPIs databas så finns de 100 vanligaste tilltalsnamnen för både kvinnor och män, och dessutom de 100 vanligaste efternamnen. Totalt så finns det alltså 20 000 olika kombinationer.

Arbete med XML

- Några endpoints i XML-format från [NamnAPI.se](#)
- <http://api.namnapi.se/v2/names.xml>
- <http://api.namnapi.se/v2/names.xml?limit=3>
- <http://api.namnapi.se/v2/names.xml?limit=3&type=firstname>
- <http://api.namnapi.se/v2/names.xml?limit=3&type=firstname&gender=male>

Hämta XML-data

```
<?php  
$url = "http://api.namnapi.se/v2/names.xml";  
  
$xml = simplexml_load_file($url);  
// Interprets an XML file into an object  
// http://php.net/simplexml-load-file  
  
print_r($xml);
```

Skriv ut data från XML

```
foreach($xml->name as $name) {  
    echo "$name->firstname $name->surname<br>";  
}
```

Tips:

https://www.w3schools.com/php/php_xml_simplexml_get.asp

NamnAPI.se Några endpoints i JSON-format

- <http://api.namnapi.se/v2/names.json>
- <http://api.namnapi.se/v2/names.json?limit=5>
- <http://api.namnapi.se/v2/names.json?gender=male>
- <http://api.namnapi.se/v2/names.json?gender=male&type=firstname>
- <http://api.namnapi.se/v2/names.json?gender=male&type=surname>
- <http://api.namnapi.se/v2/names.json?gender=female&type=firstname>

Övning

- Visa 10 slumpmässiga namn från www.NamnAPI.se

Birgit Lindgren
Moa Sundberg
Thomas Löfgren
Rune Pettersson
Joel Holm
Kenneth Sundqvist
Bertil Hansson
Hanna Persson
Håkan Engström
Claes Ek

Lösningsförslag

```
$url    = "http://api.namnapi.se/v2/names.json";
$json   = file_get_contents($url);
$array  = json_decode($json, true);

foreach($array['names'] as $key => $value){
    echo "$value[firstname] $value[surname] <br>";
}
```

Summering av dagens lektion

- Vi har jobbat mer med grunderna i API:er.
- Reflektioner kring dagens lektion?

Framåtblick inför nästa lektion

Under nästa lektion kommer vi att jobba mer med APIer.

NACKADEMIN