

# **XML API – Calling Web Service**

Inlämning 3 Calling Web Service

Namn: Anton Karlsson

E-mail: c2oantka@student.his.se

Pnr: 940916-7534

# 1 Lösningsdesign

## 1.1 Beskrivning

Uppgiften var att kalla på en webservice applikation och sedan skriva ut tillverkarna i en lista för att kunna söka på dem och få resultaten i en tabell. Denna gången skulle en self-referencing php applikation göras, alltså ingen responsepage gjordes. Tabellen gjordes återigen med column layout enligt beskrivning. Uppgiften är lik de föregående uppgifterna men här ska vi hämta datan till arrayen från en URL (web service).

```
echo "<h1>Database Trucks</h1>";
    echo "<form method='post' action='Assignment3.php'>";
    echo "<select name='country'>";

    $url="https://wwwlab.iit.his.se/gush/XMLAPI/vehiclesservice/manu-
cturer/".$manufacturer;
    $json_text = file_get_contents($url);
    $arr = json_decode($json_text);
```

**Figur 1** Kodblock för Assignment 3

Koden är ganska straight forward, startar med en header för att skriva ut Database Trucks och sedan skrivs form method post ut samt action fast eftersom det är self-referencing så refererar den till sig själv, i detta fallet Assignment3.php istället för responsepage.php.

Koden ovanför följer exemplet givet utav *Högskolan i Skövde (2021)* där vi med hjälp utav \$URL kan peka på vart vi vill get\_content (hämta data) och sedan med hjälp av json decode fylla upp vår array (\$arr). Det är väldigt likt de föregående uppgifterna men istället för att skriva ut datan i uppgiften hämtar vi den istället. Med hjälp utav print\_r funktionen så kunde man hämta varje länk som skulle användas och se dess data utskrivet i webbläsaren när applikationen kördes, tipset gavs utav *Högskolan i Skövde (2021)*.

```
echo "<option>Select";

    foreach($arr as $manufacturer){
        echo "<option value='".$manufacturer[1]."'>".$manufacturer[0]."</o-
ption>";
    }
```

**Figur 2** Kodblock för Assignment 3

Nästa steg blev att skriva ut datan i arrayen i option dropdownen med hjälp utav ovanstående kod i figur 2, arr som var fylld med den hämtade datan blir nu manufacturer arrayen istället som sedan ska finnas med i option value.

## 1.2 Resultat

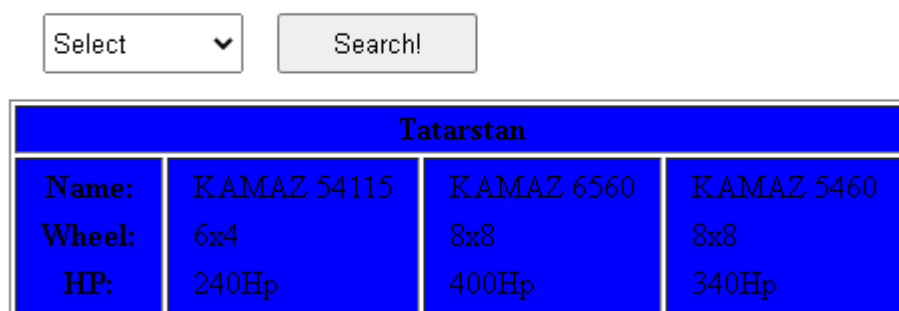
### Database Trucks



**Figur 3** Skärmdump på startsida Assignment 3

Ovanför syns det en skärmdump taget på förstasidan, där headern står klart och tydligt med en dropdown meny och en söknapp enligt instruktionerna.

### Database Trucks



Tatarstan			
Name:	KAMAZ 54115	KAMAZ 6560	KAMAZ 5460
Wheel:	6x4	8x8	8x8
HP:	240Hp	400Hp	340Hp

**Figur 4** Skärmdump på Assignment3 där sökning har gjorts på KaMAZ

Ovanför är ett exempel där en sökning på KaMAZ har gjorts i menyn, då kommer arrayens information upp som har blivit utskriven i tabellen. Tabellen är nästan identiskt som föregående uppgift fast den har blivit stylad lite grann och där Name, Wheel och HP står utskrivet för att tydliggöra information bättre.

## 2 Diskussion

I början så kändes det svårt och förstod inte riktigt tills materialet som tillhandahölls utav *Högskolan i Skövde (2021)* lästes igenom där olika tips och exempel gavs, så när man tog ett steg i taget med att kalla på webservices och se datan som fanns i den fyllda arrayen så kände man igen sig från föregående uppgifter och det löstes ganska smidigt efter det.

### 2.1 Fördel och nackdelar med webservice

Fördelen med webservice är att man enkelt kan nå koden online och kan helt enkelt återanvända den smidigt. Istället för att skicka ett stort dokument eller massvis med text till någon kan man helt enkelt tillhandahålla en URL som man sedan kan använda för att hämta data till sin applikation.

Nackdelen är att man inte helt enkelt ser datan i sin array i php filen utan man måste först skriva `print_r` för att få se datan utskriven och sen kopiera eller ha väldigt bra minne för att komma ihåg vilken data som är vart. Så vid mindre datamängd kan det vara smidigare att köra utan webservice, men vid stora data mängde från olika håll är det helt klart mycket smidigare med webservice istället.

Sist men inte minst är en stor nackdel att om man förlitar sig på webservice så funkar inte applikationen utan internetanslutning.

## Referenser

Högskolan i Skövde (2021) *Lecture 3 – XML API – IT385G* [Online]. Available at [https://www.youtube.com/watch?v=ZY\\_Mqqro3Ug&list=PLAxZA8hcpPLLZMWk4SvetyJ563DrR\\_aV2&index=5](https://www.youtube.com/watch?v=ZY_Mqqro3Ug&list=PLAxZA8hcpPLLZMWk4SvetyJ563DrR_aV2&index=5) (Accessed: 09 April 2021)