

## Webszerver és adatbázis együttműködése Dockerben

Ahhoz, hogy a feladatot teljesíteni tudjuk, mindenképpen szükségünk van egy webes illetve egy adatbázis komponensre. Webes odalon valahogyan ki kell nyernünk az adatokat az adatbázisból, erre legegyszerűbb mód a PHP.

Külön command fájlban megírtam az adatbázis, táblalétrehozás és feltöltés parancsait.

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

CREATE DATABASE IF NOT EXISTS `db` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;
USE `db`;

CREATE TABLE `Auto` (
  `id` int NOT NULL,
  `Rendszam` varchar(255) DEFAULT NULL,
  `Tipus` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COLLATE=utf8_hungarian_ci;;

INSERT INTO Auto VALUES(1, 'AAA-111', 'Opel');
INSERT INTO Auto VALUES(2, 'AAA-112', 'Citroen');
INSERT INTO Auto VALUES(3, 'AAA-113', 'Ford');
INSERT INTO Auto VALUES(4, 'AAA-114', 'Nissan');

COMMIT;
```

Ezt követően megírtam a PHP kódot is, ami a következő képen látható.

```

<html>
<head>
<title>Selecting Records</title>
<link href="./css/bootstrap.css" rel="stylesheet">
</head>
<body>
<h1>Table:</h1>
<?php
$dbhost = $_ENV["MYSQL_HOST"];
$dbuser = $_ENV["MYSQL_USER"];
$dbpass = $_ENV["MYSQL_PASSWORD"];
$dbase = $_ENV["MYSQL_DATABASE"];
$conn = mysqli_connect($dbhost, $dbuser, $dbpass, $dbase);

if(! $conn ) {
    die('Could not connect: ' . mysqli_error($conn));
}

mysqli_select_db( $conn, 'db' );
$sql = "SELECT * FROM Auto";
$retval = mysqli_query( $conn, $sql );
if(! $retval ) {
    die('Could not get data: ' . mysqli_error($conn));
}
echo "<table class='table'><thead class='thead-dark'><tr><th>id</th><th>Rendszam</th><th>Tipus</th></tr></thead>";
while($row = mysqli_fetch_array($retval)) {
    echo "<tr>";
    echo "<td>" . $row['id'] . "</td>";
    echo "<td>" . $row['Rendszam'] . "</td>";
    echo "<td>" . $row['Tipus'] . "</td>";
    echo "</tr>";
}
echo "</table>";

mysqli_close($conn);
?>
<script src="./js/bootstrap.js"></script>
</body>
</html>

```

Itt jogosan merülhet fel a kérdés, hogy a használt adatbázis adatait honnan olvasom ki. Mivel elég sok helyen volt ezekre szükség, ezért ehhez egy külön env fájlt hoztam létre, amiben ezeket környezeti változóként definiálom.

```

MYSQL_USER=web_user
MYSQL_PASSWORD=passwd
MYSQL_ROOT_PASSWORD=passwd
MYSQL_DATABASE=db

MYSQL_HOST=database

```

Ezen kívül szükség volt még egy yaml fájlra is, ami a következő képen látható:

```

version: "3.0"

services:
  webserver:
    build:
      context: ./site
      dockerfile: Dockerfile
    image: beadando:1.0
    environment:
      - MYSQL_USER=${MYSQL_USER}
      - MYSQL_PASSWORD=${MYSQL_PASSWORD}
      - MYSQL_DATABASE=${MYSQL_DATABASE}
      - MYSQL_HOST=${MYSQL_HOST}
    ports:
      - 80:80
    networks:
      - web
    depends_on:
      database:
        condition: service_healthy

  database:
    image: mysql
    command: --default-authentication-plugin=mysql_native_password
    volumes:
      - ./sql:/docker-entrypoint-initdb.d
    environment:
      - MYSQL_USER=${MYSQL_USER}
      - MYSQL_PASSWORD=${MYSQL_PASSWORD}
      - MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD}
      - MYSQL_DATABASE=${MYSQL_DATABASE}
    ports:
      - 3306:3306
    networks:
      - web
    healthcheck:
      test: ["CMD", "mysqladmin" ,"ping", "-h", "localhost"]
      timeout: 3s
      retries: 10

networks:
  web:
    driver: bridge


```

Itt feltűnhet néhány dolog. Az egyik a network. Ahhoz, hogy tudjon az adatbázis és a weblap kommunikálni ahhoz ismereteim alapján ezeknek egy networkbe kell tartozni. A másik a healthcheck. Ez az adatbázisnál van definiálva, ami azt csinálja, hogy megpingeli az adatbázist, ha az nem válaszol akkor 10- szer próbálkozik maximum, ezután unhealthynek fogja jelölni azt. Ez azért fontos, mivel a webserver résznél van egy depends\_on tag, ami az adatbázisra hivatkozik. Ez azt fogja csinálni, hogy csak akkor indul el a webserver, ha az adatbázis healthy és fut.

Amikor ez mind megvolt, akkor CMD segítségével elnavigáltam addig a mappáig ahová a fájlokat pakoltam. Itt kiadtam a docker-compose up parancsot. Ekkor elindult a konténer, majd ha a localhost 80-as portját böngészőben megnyitottam, akkor a következő látvány tárult elém.

Table:

id	Rendszam	Típus
1	AAA-111	Opel
2	AAA-112	Citroen
3	AAA-113	Ford
4	AAA-114	Nissan

**Images** [Give Feedback](#) 

**LOCAL** REMOTE REPOSITORIES

☐ In use only

NAME ↑		TAG	IMAGE ID	CREATED	SIZE
adminer	IN USE	latest	250ff692aeba	22 days ago	90.69 MB
beadando	IN USE	1.0	9b03f7b1f085	less than a minute ago	428.31 MB
mysql	IN USE	latest	c2c2eba5ae85	1 day ago	535.46 MB