https://www.halvorsen.blog

Simple PHP Web API

Hans-Petter Halvorsen

Contents

- A short overview of APIs in general will be given.
  - API is short for Application Programming Interface.
- We will create a simple Web API using PHP.
  - PHP is a server-side framework/programming language for creating web pages and web contents.
  - We will use MySQL as the Database system.
  - We will use the phpMyAdmin tool to administrator and setup the database.
  - We will implement a simple CRUD Web API that Create, Read, Update and Delete data in the Database.
  - We will use Visual Studio Code as the Code editor.
- Finally, we will use Python and Thonny editor to test the REST API.

Introduction

Hans-Petter Halvorsen

API

- Application Programming Interface (API).
- An API is a way for two or more computer programs or components to communicate with each other.
- It is a type of software interface that offers a service to other software.
- APIs come in many shapes, some examples are SOAP API, REST API, GraphQL API, etc.
- Most programming languages today have components/libraries that can be used both to create APIs and to consume APIs (using existing APIs).

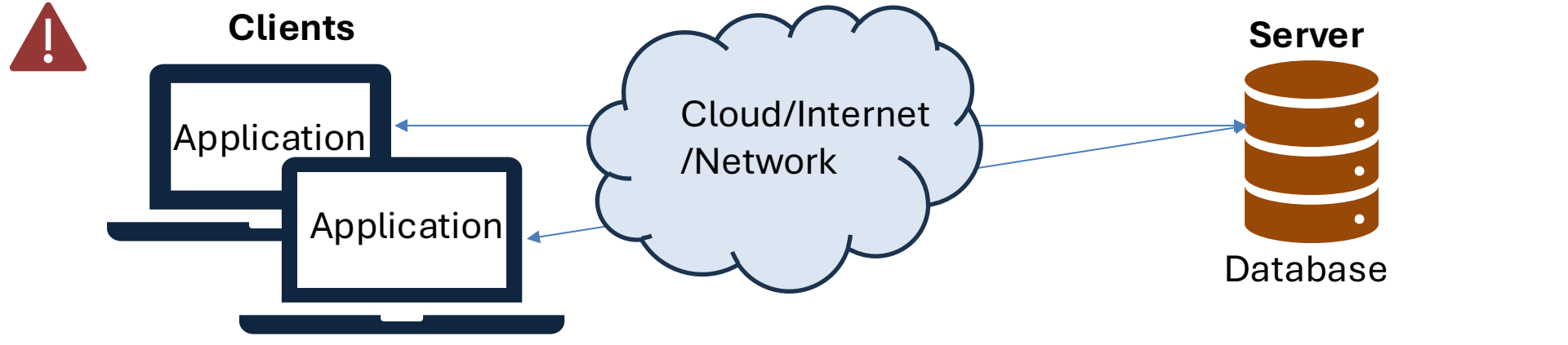
Web API

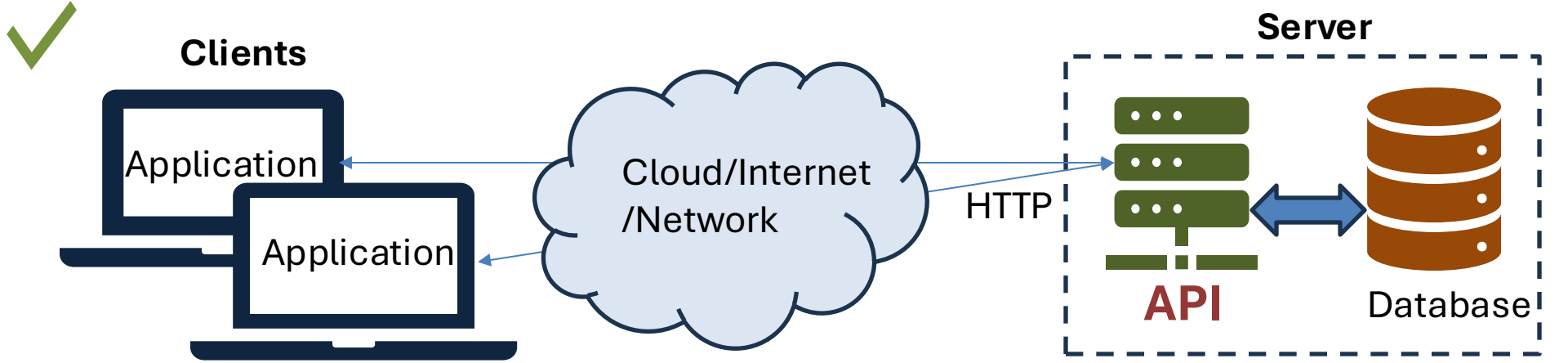
- We can create/use APIs for internal use inside an Application or between 2 or more Applications.
- Basically, an API can be just a Class with Methods that you use several places inside an Application or that you share between multiple Applications.
- A set of Stored Procedures in a Database can also be an API.
- When the Application that consume/use the API is on a local PC and the API itself is located on a Server, we can talk about so-called “Web APIs”.
- Such Web APIs also very often perform CRUD operations against a Database located on the Web.
- Normally it is not allowed to connect directly to a Database located in the Cloud from a local computer unless you configure and give access to the IP addresses for those clients.

CRUD: **C**reate, **R**ead, **U**pdate, **D**elete Data

Web API

Normally it is not allowed to connect directly to a Database located in the Cloud from a local computer unless you configure and give access to the IP addresses for those clients.





JSON

- When it comes to Web APIs and REST APIs JSON is the standard for the data format.
- Example:

```
{
  "Name": "John Wayne",
  "Work": "Actor",
  "Age": 52
  "Children": [
    "Lisa",
    "Thomas",
    "Knut"
  ]
}
```

https://en.wikipedia.org/wiki/JSON

PHP + MySQL

- You need to have a PHP + MySQL Environment on your local computer on get access to it from a server/Internet.
- For local installation you need to download and install Apache, PHP and MySQL.
- You can get server access from many providers (free or paid).
- I will use an internal LAMP server available for employees and students at my University.

LAMP

- LAMP = **L**inux, **A**pache, **M**ySQL, **P**HP
  - PHP is the Programming Language
  - MySQL is the Database System
  - Apache is the Web Server software
  - Linux is the operating system where the Web Server is running

Each part in LAMP is free and open-source, so it is a popular web hosting environment. You find also lots of online documentation and a large community.

LAMP/PHP Web Hosting

- There exists hundreds/thousands of different LAMP/PHP/MySQL Hosting Providers, some free but mostly paid options.
- Hostinger - https://www.hostinger.no
- InfinityFree - https://www.infinityfree.com
- PRO ISP - https://www.proisp.no
- +++ (Just Google)

Why use Web API?

- Normally it is not allowed to connect directly to a Database located in the Cloud from a local computer
  - unless you configure and give access to the IP addresses for those clients.
  - Typically, your IT Department don’t allow that
- You can use the same API for multiple Applications, let say you have a Desktop App, an iPhone App and an Android App
  - All can use the same API
  - You save time and money by developing only once instead of specific code for each application.

API Summary

- Basically, Web APIs, REST APIs or HTTP APIs are basically the same.
- It is just different names for the same.
- They all communicate via Internet and use HTTP as communication protocol.
- And they use JSON (or sometimes XML) as Data Format.

https://www.halvorsen.blog

PHP Web API Example

Hans-Petter Halvorsen

Example

We will create a Web API with CRUD functionality, meaning we will insert, read, update and delete data in a Database.

- We will start by creating a Database and Table using MySQL and the phpMyAdmin tool.
- Then we will create the PHP code for the REST API.
- We will test the API in the URL in the Web Browser.
- Finaly we will test the API creating some basic Python examples.

Tools

The following tool will be used in this example:

- PHP
- MySQL
  - phpMyAdmin
- Visual Studio Code
- WinSCP
- Python
  - Thonny Python Editor

PHP + MySQL

- You need to have a PHP + MySQL Environment on your local computer on get access to it from a server/Internet.
- For local installation you need to download and install Apache, PHP and MySQL.
- You can get server access from many providers (free or paid).
- I will use an internal LAMP server available for employees and students at my University.

Database

We start by creating a simple Database Table, e.g.:

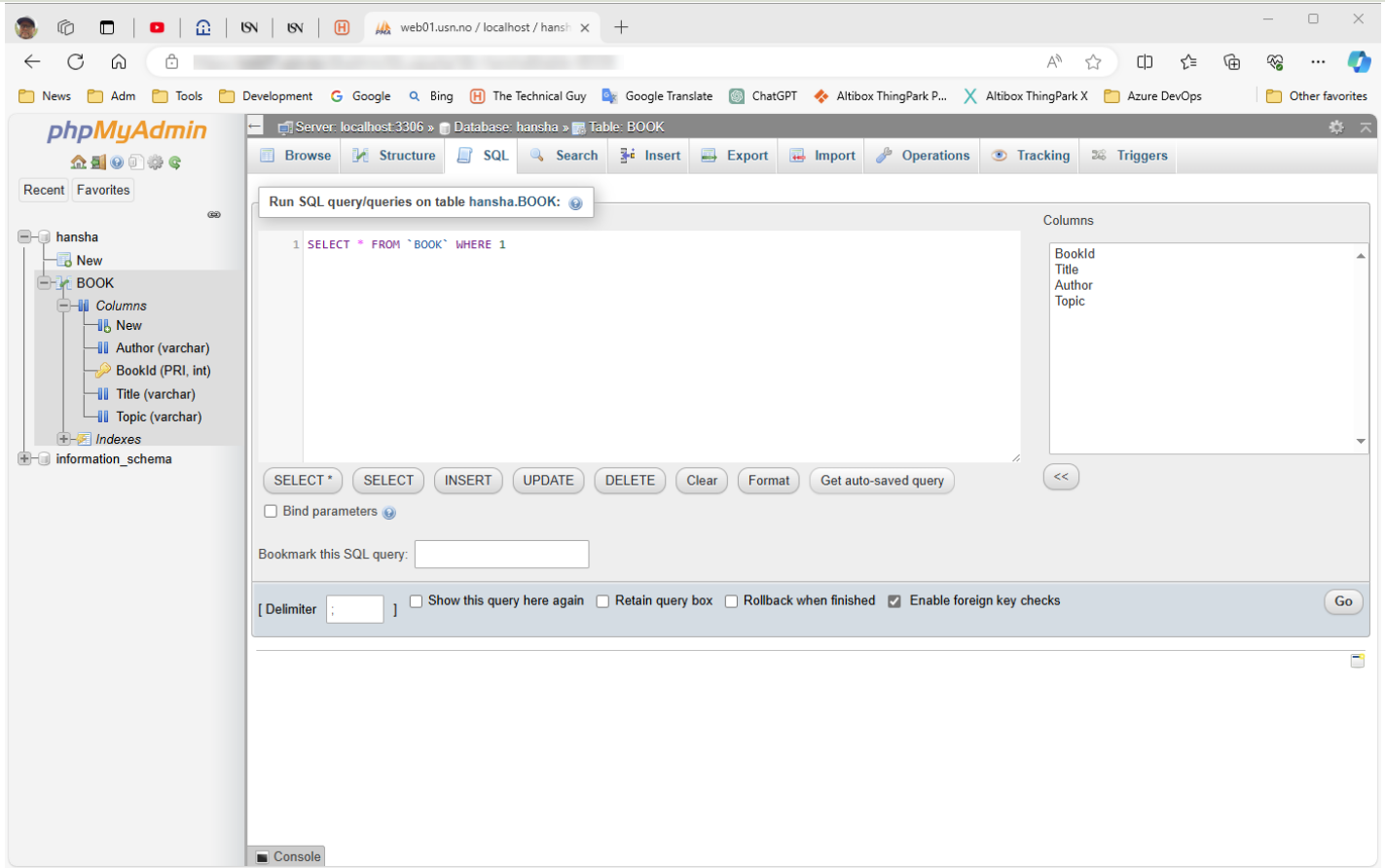
```
CREATE TABLE BOOK
(
    BookId int PRIMARY KEY AUTO_INCREMENT,
    Title varchar(100) NOT NULL,
    Author varchar(100) NOT NULL,
    Topic varchar(100) NOT NULL
);
```

Database

We can also insert some data into the Table, e.g.:

```
insert into BOOK (Title, Author, Topic) values
('Web Apps, 'Elvis Presly', 'Programming');

insert into BOOK (Title, Author, Topic) values
('IoT and Cloud', 'John Wayne', 'IoT');

insert into BOOK (Title, Author, Topic) values
('C#', 'Rune Hansen', 'Programming');
```

phpMyAdmin


web01.usn.no / localhost / hansh
News
Adm
Tools
Development
Google
Bing
The Technical Guy
Google Translate
ChatGPT
Altibox ThingPark P...
Altibox ThingPark X
Azure DevOps
Other favorites
phpMyAdmin
Recent
Favorites
hansha
New
BOOK
Columns
New
Author (varchar)
BookId (PRI, int)
Title (varchar)
Topic (varchar)
Indexes
information_schema
Server: localhost:3306 » Database: hansha » Table: BOOK
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Tracking
Triggers
Run SQL query/queries on table hansha.BOOK:
SELECT * FROM `BOOK` WHERE 1
Columns
BookId
Title
Author
Topic
SELECT *
SELECT
INSERT
UPDATE
DELETE
Clear
Format
Get auto-saved query
<<
Bind parameters
Bookmark this SQL query:
[Delimiter ;]
Show this query here again
Retain query box
Rollback when finished
Enable foreign key checks
Go
Console

PHP API “Methods”

- We can create 2 PHP files, e.g.:
- config.php - contains usernames, passwords, etc. for the MySQL Server database
- In a subfolder, e.g., called “book“ we create 5 PHP files:
  - GetBooks.php
  - GetBookById.php
  - InsertBook.php
  - UpdateBook.php
  - DeleteBook.php

config.php

Connect to your Database:

```
<?php
$host = 'localhost';
$dbname = 'your_database_name';
$username = 'your_username';
$password = 'your_password';
try {
 $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
 $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
 die("Database connection failed: " . $e->getMessage());
}
?>
```

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

GetBook

This method is used to retrieve data from the Database

Hans-Petter Halvorsen

GetBooks.php

```php
<?php
require_once 'config.php';

// Set the content type to JSON
header('Content-Type: application/json');

// Handle HTTP methods
$method = $_SERVER['REQUEST_METHOD'];

$stmt = $pdo->query('SELECT * FROM BOOK');
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);
echo json_encode($result);
?>
```

Visual Studio Code

File Edit Selection View ··· Search

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

config.php GetBooks.php

C: > Users > hansha > OneDrive > web01.usn.no > api > book > 1.0 > GetBooks.php

```php
<?php
require_once 'config.php';

// Set the content type to JSON
header('Content-Type: application/json');

// Handle HTTP methods
$method = $_SERVER['REQUEST_METHOD'];

$stmt = $pdo->query('SELECT * FROM BOOK');
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);
echo json_encode($result);

?>
```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 9, Col 1 Spaces: 4 UTF-8 CRLF PHP

WinSCP (FTP)

1.0 - web01.usn.no - WinSCP

Synchronize | Queue | Transfer Settings Default

web01.usn.no | New Session

C: OS | Upload | Edit | Properties | New

C:\Users\hansha\OneDrive\web01.usn.no\api\book\1.0\

Name	Size	Type	Changed
..		Parent directory	2024-06-10 09:57:16
config.php	1 KB	PHP Source File	2024-02-20 10:50:09
DeleteBook.php	1 KB	PHP Source File	2024-06-10 09:57:50
GetBookById.php	1 KB	PHP Source File	2024-06-10 10:14:29
GetBooks.php	1 KB	PHP Source File	2024-06-10 09:55:06
InsertBook.php	1 KB	PHP Source File	2024-06-10 09:57:50
UpdateBook.php	1 KB	PHP Source File	2024-06-10 09:57:50

0 B of 2,41 KB in 0 of 6

1.0 | Find Files | Download | Edit | Properties | New

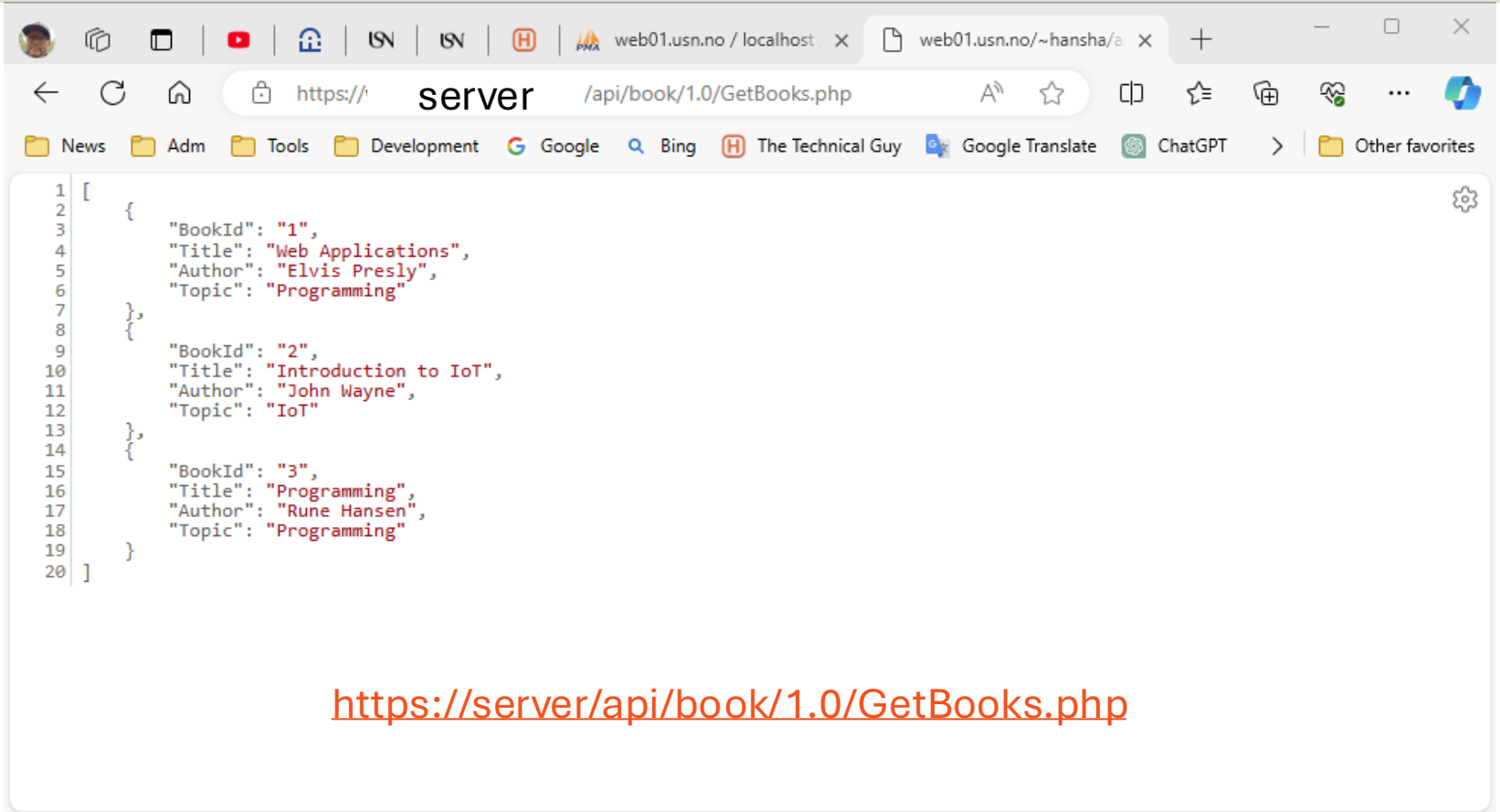
/home/hansha/public_html/api/book/1.0/

Name	Size	Changed	Rights
..		2024-06-10 09:57:58	rwxr-sr-
config.php	1 KB	2024-02-20 10:50:09	rw-r--r-
DeleteBook.php	1 KB	2024-06-10 09:57:50	rw-r--r-
GetBookById.php	1 KB	2024-06-10 10:14:29	rw-r--r-
GetBooks.php	1 KB	2024-06-10 09:55:06	rw-r--r-
InsertBook.php	1 KB	2024-06-10 09:57:50	rw-r--r-
UpdateBook.php	1 KB	2024-06-10 09:57:50	rw-r--r-

0 B of 2,41 KB in 0 of 6

SFTP-3 | 0:43:20

GetBooks - Web Browser




```
[
    {
        "BookId": "1",
        "Title": "Web Applications",
        "Author": "Elvis Presly",
        "Topic": "Programming"
    },
    {
        "BookId": "2",
        "Title": "Introduction to IoT",
        "Author": "John Wayne",
        "Topic": "IoT"
    },
    {
        "BookId": "3",
        "Title": "Programming",
        "Author": "Rune Hansen",
        "Topic": "Programming"
    }
]
```


https://server/api/book/1.0/GetBooks.php

GetBooks - Python

```python
import requests

server = "https://servername/"
api = "api/book/1.0/"
method = "GetBooks"
url = server + api + method + ".php"
print(url)

response = requests.get(url)
print(response)
print(response.json())
```

Thonny – Running Python Script

Thonny - C:\Users\hansha\OneDrive\Courses\Webutvikling\Tutorials\Simple WebAPI\Development\Python\api_getbooks.py @ 8 : 1

File Edit View Run Tools Help

api_getbooks.py

```python
import requests

server = "https://        server        /"
api = "api/book/1.0/"
method = "GetBooks"
url = server + api + method + ".php"
print(url)

response = requests.get(url)
print(response)
print(response.json())
```

Shell

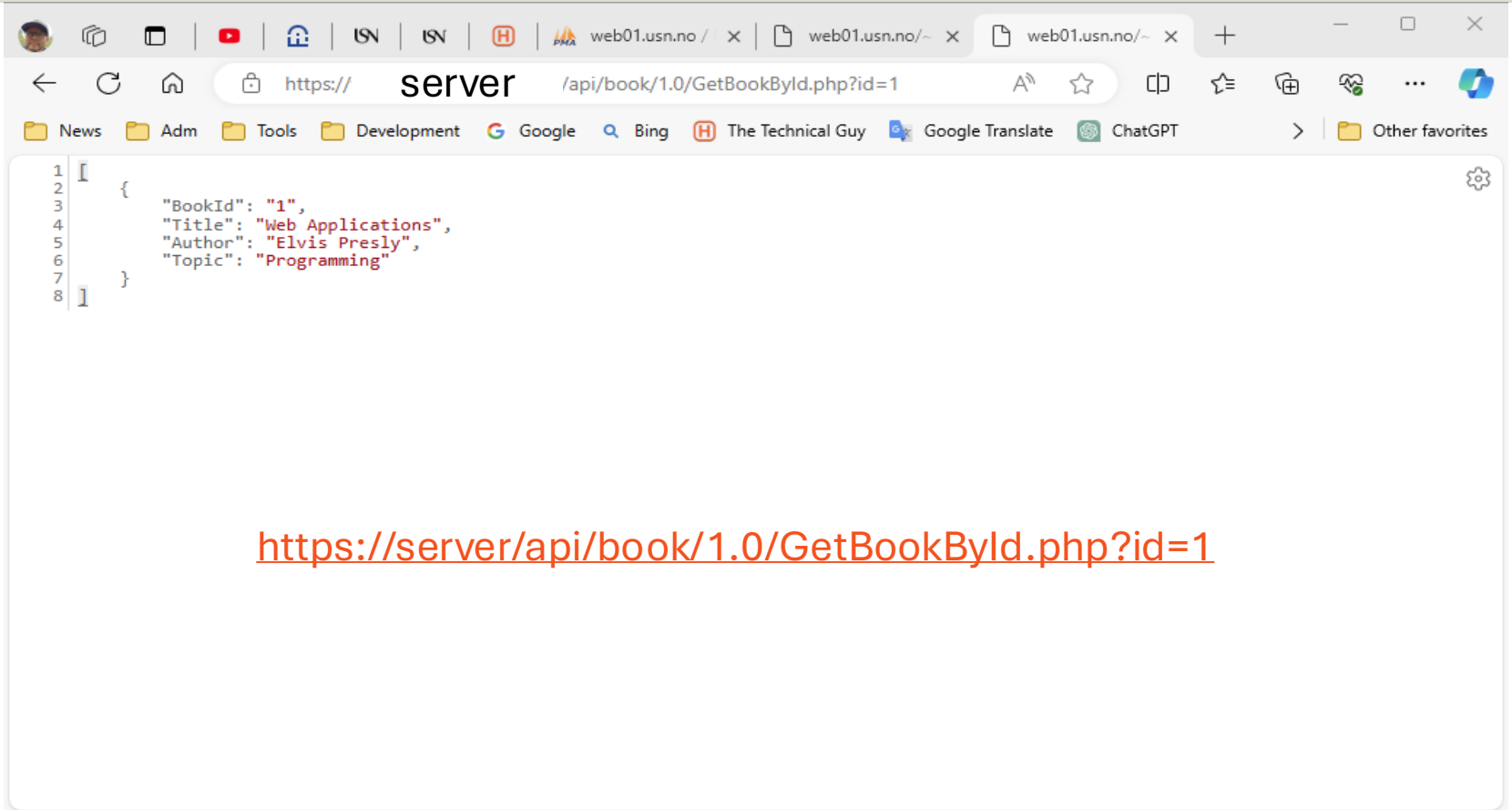
```
>>> %Run api_getbooks.py
                                /api/book/1.0/GetBooks.php
 <Response [200]>
 [{'BookId': '1', 'Title': 'Web Applications', 'Author': 'Elvis Presly', 'Topic': 'Programming'}, {'BookId': '
 2', 'Title': 'Introduction to IoT', 'Author': 'John Wayne', 'Topic': 'IoT'}, {'BookId': '3', 'Title': 'Progra
 mming', 'Author': 'Rune Hansen', 'Topic': 'Programming'}]

>>>
```

Local Python 3 • Thonny's Python

GetBookById.php

```php
<?php
require_once 'config.php';

// Set the content type to JSON
header('Content-Type: application/json');

// Handle HTTP methods
$method = $_SERVER['REQUEST_METHOD'];

$id = htmlspecialchars($_GET['id']);

$stmt = $pdo->prepare('SELECT * FROM BOOK WHERE BookId=?');
$stmt->execute([$id]);

$result = $stmt->fetchAll(PDO::FETCH_ASSOC);
echo json_encode($result);
?>
```

GetBookById - Web Browser



https://server/api/book/1.0/GetBookById.php?id=1

GetBookById - Python

```
import requests

server = "https://servername/"
api = "api/book/1.0/"
method = "GetBookById"

id = "1"
query = "?id=" + id

url = server + api + method + ".php" + query
print(url)

response = requests.get(url)
print(response)
print(response.json())
```

Thonny – Running Python Script

Thonny - C:\Users\hansha\OneDrive\Courses\Webutvikling\Tutorials\Simple WebAPI\Development\Python\api_getbookbyid.py @ 15 : 23

File Edit View Run Tools Help

api_getbooks.py | api_getbookbyid.py

```python
import requests

server = "https://        server        /"
api = "api/book/1.0/"
method = "GetBookById"

id = "1"
query = "?id=" + id

url = server + api + method + ".php" + query
print(url)

response = requests.get(url)
```

Shell

```
>>> %Run api_getbookbyid.py
                                /api/book/1.0/GetBookById.php?id=1
 [{'BookId': '1', 'Title': 'Web Applications', 'Author': 'Elvis Presly', 'Topic': 'Programming'}]
>>>
```

Local Python 3 • Thonny's Python

InsertBook

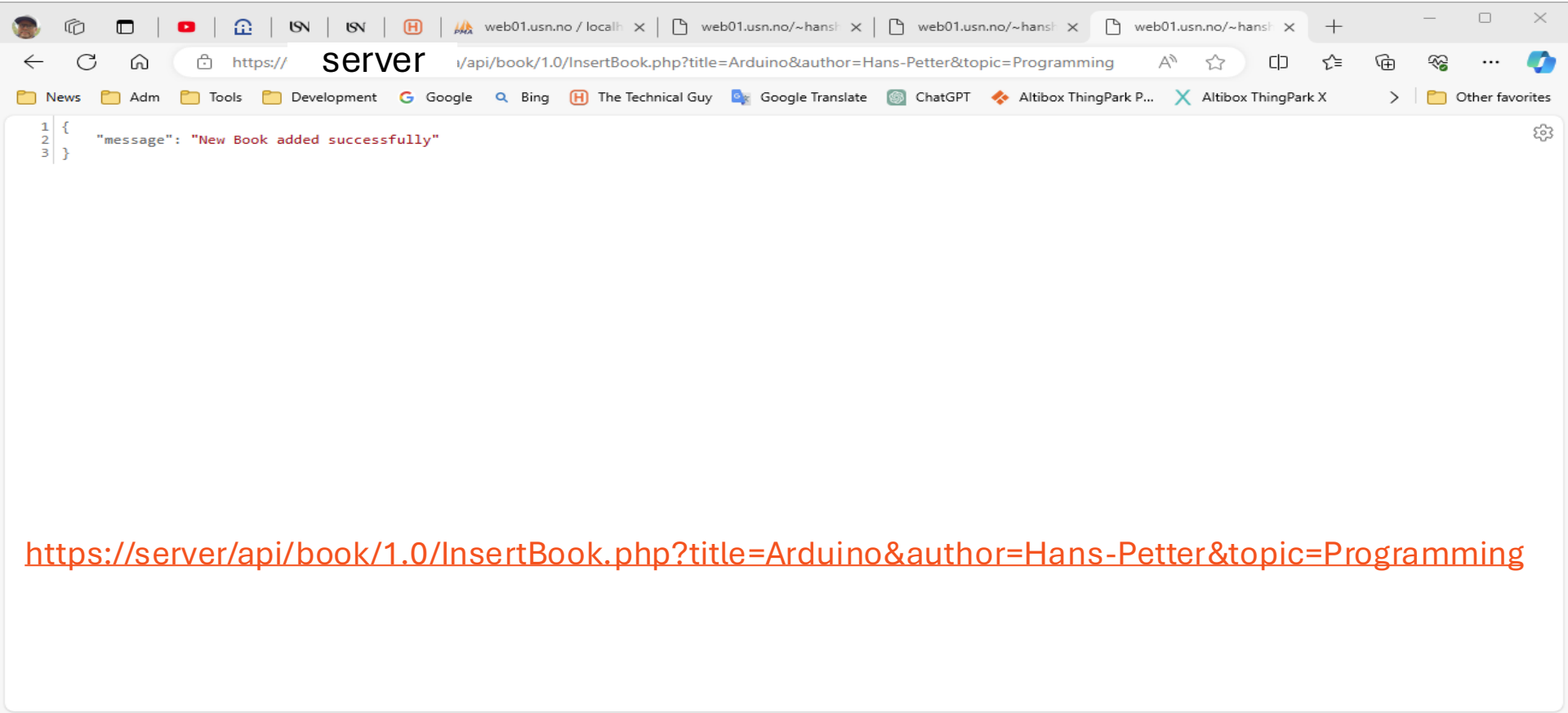
This method is used to insert new data stored in the Database

Hans-Petter Halvorsen

InsertBook.php

```php
<?php
require_once 'config.php';

// Set the content type to JSON
header('Content-Type: application/json');

// Handle HTTP methods
$method = $_SERVER['REQUEST_METHOD'];

$title = htmlspecialchars($_GET['title']);
$author = htmlspecialchars($_GET['author']);
$topic = htmlspecialchars($_GET['topic']);

$stmt = $pdo->prepare('INSERT INTO BOOK (Title, Author, Topic) VALUES (?, ?, ?)');
$stmt->execute([$title, $author, $topic]);

echo json_encode(['message' => 'New Book added successfully']);

?>
```

InsertBook - Web Browser



https://server/api/book/1.0/InsertBook.php?title=Arduino&author=Hans-Petter&topic=Programming

InsertBook - Python

```
import requests

server = "https://servername/"
api = "api/book/1.0/"
method = "InsertBook"

title = "Arduino"
author = "Hans-Petter"
topic = "IoT"
query = "?title=" + title + "&author=" + author + "&topic=" + topic

url = server + api + method + ".php" + query
print(url)

response = requests.get(url)
print(response)
print(response.json())
```

UpdateBook

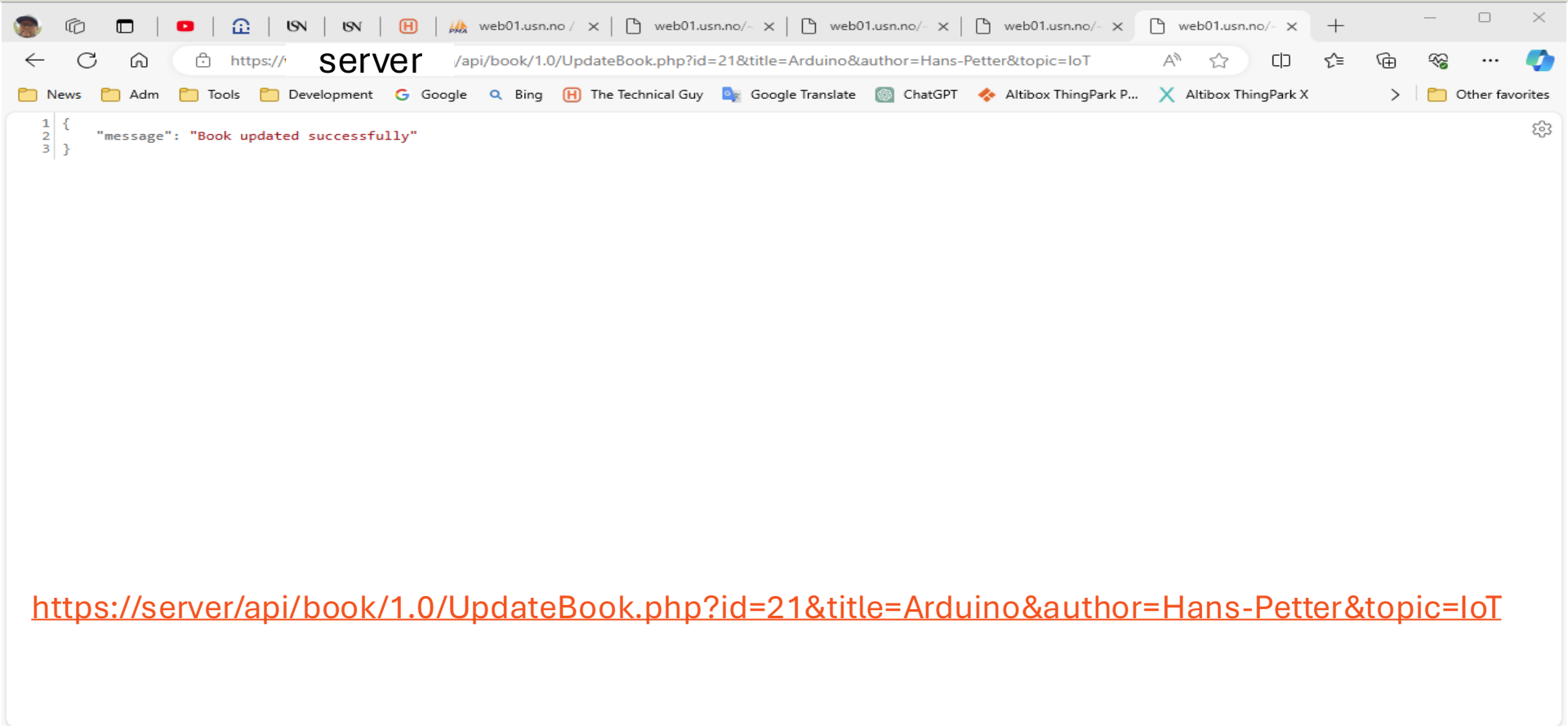
This method is used to update data stored in the Database

Hans-Petter Halvorsen

UpdateBook.php

```php
<?php
require_once 'config.php';

// Set the content type to JSON
header('Content-Type: application/json');

// Handle HTTP methods
$method = $_SERVER['REQUEST_METHOD'];

$id = htmlspecialchars($_GET['id']);
$title = htmlspecialchars($_GET['title']);
$author = htmlspecialchars($_GET['author']);
$topic = htmlspecialchars($_GET['topic']);

$stmt = $pdo->prepare('UPDATE BOOK SET Title=?, Author=?, Topic=? WHERE BookId=?');
$stmt->execute([$title, $author, $topic, $id]);

echo json_encode(['message' => 'Book updated successfully']);

?>
```

UpdateBook - Web Browser



https://server/api/book/1.0/UpdateBook.php?id=21&title=Arduino&author=Hans-Petter&topic=IoT

UpdateBook - Python

```
import requests

server = "https://servername/"
api = "api/book/1.0/"
method = "UpdateBook"

id = "18"
title = "Arduino2"
author = "Hans-Petter2"
topic = "Programming"
query = "?id=" + id + "&title=" + title + "&author=" + author + "&topic=" + topic

url = server + api + method + ".php" + query
print(url)

response = requests.get(url)
print(response)
print(response.json())
```

DeleteBook

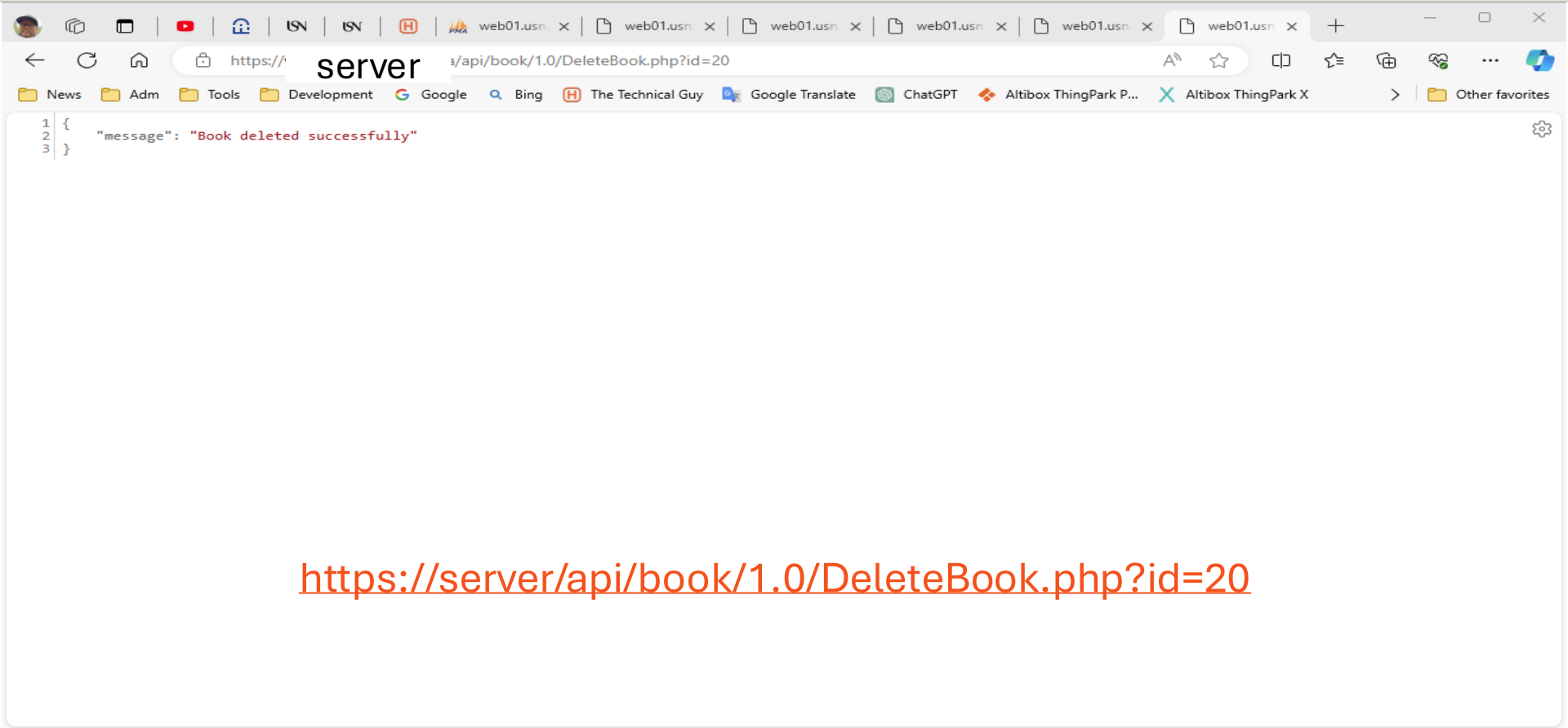
This method is used to delete data stored in the Database

Hans-Petter Halvorsen

DeleteBook.php

```php
<?php
require_once 'config.php';

// Set the content type to JSON
header('Content-Type: application/json');

// Handle HTTP methods
$method = $_SERVER['REQUEST_METHOD'];

$id = htmlspecialchars($_GET['id']);

$stmt = $pdo->prepare('DELETE FROM BOOK WHERE BookId=?');
$stmt->execute([$id]);

echo json_encode(['message' => 'Book deleted successfully']);

?>
```

DeleteBook - Web Browser



https://server/api/book/1.0/DeleteBook.php?id=20

Delete Book - Python

```
import requests

server = "https://servername/"
api = "book/"
method = "DeleteBook"

id = "19"
query = "?id=" + id

url = server + api + method + ".php" + query
print(url)

response = requests.get(url)
print(response)
print(response.json())
```

Summary

Hans-Petter Halvorsen

Web API Structure

Make sure to keep a good folder structure. Sooner or later, you will have multiple APIs and you also probably need to maintain multiple versions of the same API, e.g., like this:

- API
  - Book
    - 1.0
      - » PHP API Files
    - 2.0
      - » PHP API Files
  - Customer
    - 1.0
      - » PHP API Files
    - ..

Basic Security and Access Control

- It is a good idea to add some security and access control.
- In that way only users that has access can get, insert, update or delete data.
- You then need to add a basic check in your PHP files to check if password is correct.

Example:

https://server/api/book/1.0/DeleteBook.php**?password=xxx**&id=20

Updated PHP Files

File Edit Selection View Go Run ··· Search

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

config.php × GetBooks.php GetBookById.php InsertBook.php UpdateBook.php DeleteBook.php

C: > Users > hansha > OneDrive > web01.usn.no > api > book > 1.0 > config.php

```php
<?php
$api_key='[redacted]';
$host = 'localhost';
$dbname = '[redacted]';
$username = '[redacted]';
$password = '[redacted]';
try {
 $pdo = new PDO("mysql:host=$host;dbname=$dbname
 $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERR
} catch (PDOException $e) {
 die("Database connection failed: " . $e->getMes
}
?>
```

Restricted Mode ⊗ 0 ⚠ 0 0

File Edit Selection View Go Run ··· Search

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

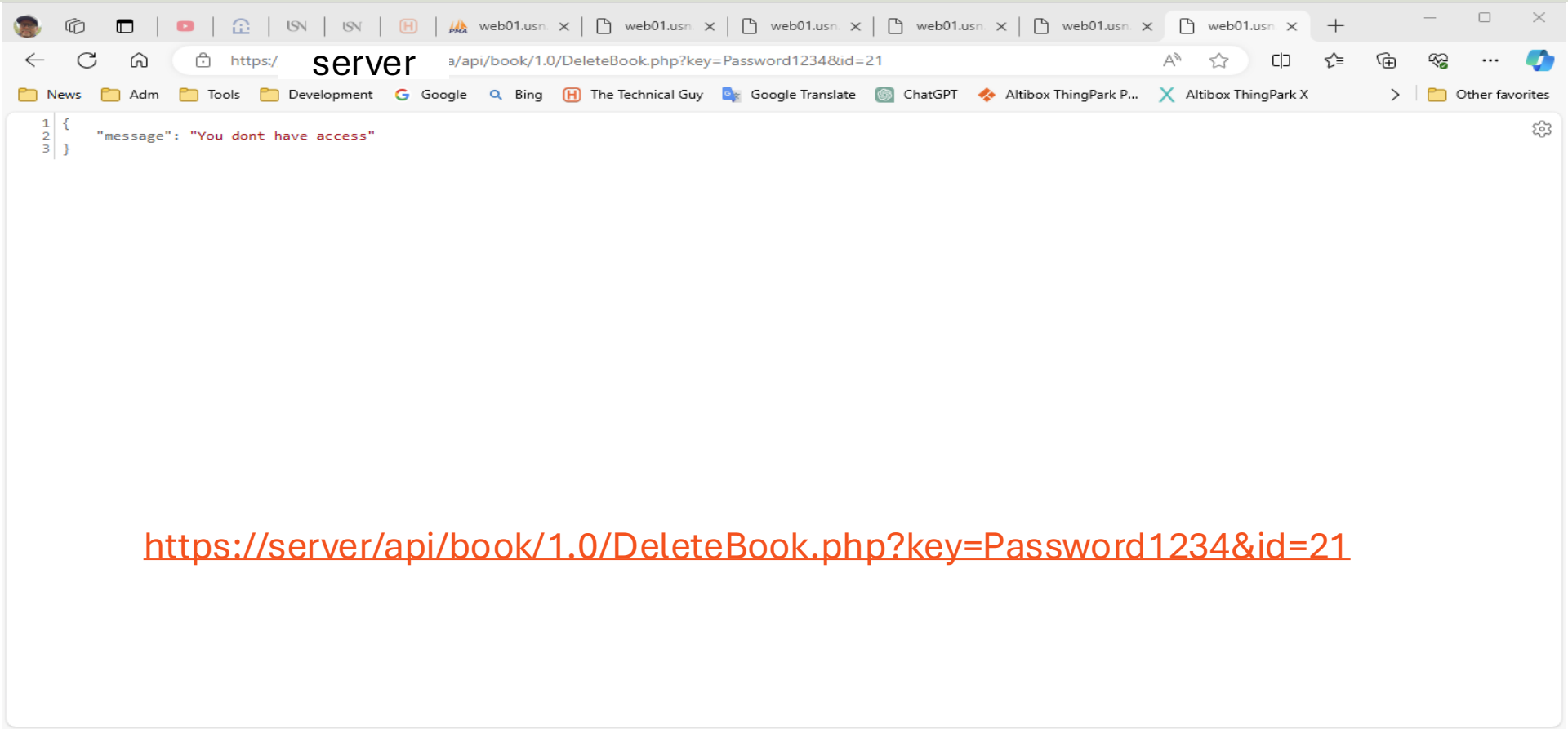
config.php GetBooks.php GetBookById.php InsertBook.php UpdateBook.php DeleteBook.php ×

C: > Users > hansha > OneDrive > web01.usn.no > api > book > 1.0 > DeleteBook.php

```php
<?php
require_once 'config.php';

// Set the content type to JSON
header('Content-Type: application/json');

// Handle HTTP methods
$method = $_SERVER['REQUEST_METHOD'];

$key = htmlspecialchars($_GET['key']);
$id = htmlspecialchars($_GET['id']);

if ($key == $api_key)
{
    $stmt = $pdo->prepare('DELETE FROM BOOK WHERE BookId=?');
    $stmt->execute([$id]);

    echo json_encode(['message' => 'Book deleted successfully']);
}
else
    echo json_encode(['message' => 'You dont have access']);

?>
```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 22, Col 1 Spaces: 4 UTF-8 CRLF PHP

Test in Web Browser



https://server/api/book/1.0/DeleteBook.php?key=Password1234&id=21

Python

Thonny - C:\Users\hansha\OneDrive\Courses\Webutvikling\Tutorials\Simple WebAPI\Development\Python\api_deletebook.py @ 7 : 20

File Edit View Run Tools Help

api_getbooks.py | api_getbookbyid.py | api_insertbook.py | api_updatebook.py | api_deletebook.py

```python
import requests

server = "https://        server        /"
api = "api/book/1.0/"
method = "DeleteBook"

key = "Password1234"
id = "19"
query = "?key=" + key + "&id=" + id

url = server + api + method + ".php" + query
print(url)

response = requests.get(url)
print(response)
print(response.json())
```

Shell

```
>>> %Run api_deletebook.py
                                  api/book/1.0/DeleteBook.php?key=Password1234&id=19
<Response [200]>
{'message': 'You dont have access'}
>>>
```

Local Python 3 • Thonny's Python

Summary

- We have created a simple Web API using PHP.
- We tested the Web API using Python.
- In general, we can use any kind of programming language to interact with this API.
- E.g., we an create a Windows Forms Application in Visual Studio and C#.
- In that way we can insert, read, update or delete data in the remote database from a local application.
- Normally you cannot directly interact with a remote SQL Database from your local computer due to security reasons.
- There are lots of improvements to be made to make a better code structure (create classes, etc.), make it more robust with error handling, improved security, etc. But I leave that to you to improve.
- The code is made simple to illustrate the basic principles using Web APIs.

References

- PHP Tutorial: https://www.w3schools.com/php
- MySQL Tutorial: https://www.w3schools.com/mysql
- https://medium.com/@miladev95/how-to-make-crud-rest-api-in-php-with-mysql-5063ae4cc89
- Python & APIs: https://realpython.com/python-api/

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog