https://www.halvorsen.blog

PHP REST API

Hans-Petter Halvorsen

Contents

- A short overview of APIs in general will be given.
  - API is short for Application Programming Interface.
- Introduction to REST API.
- We will create a simple REST 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 CRUD REST 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 to test the REST API.

Introduction

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).

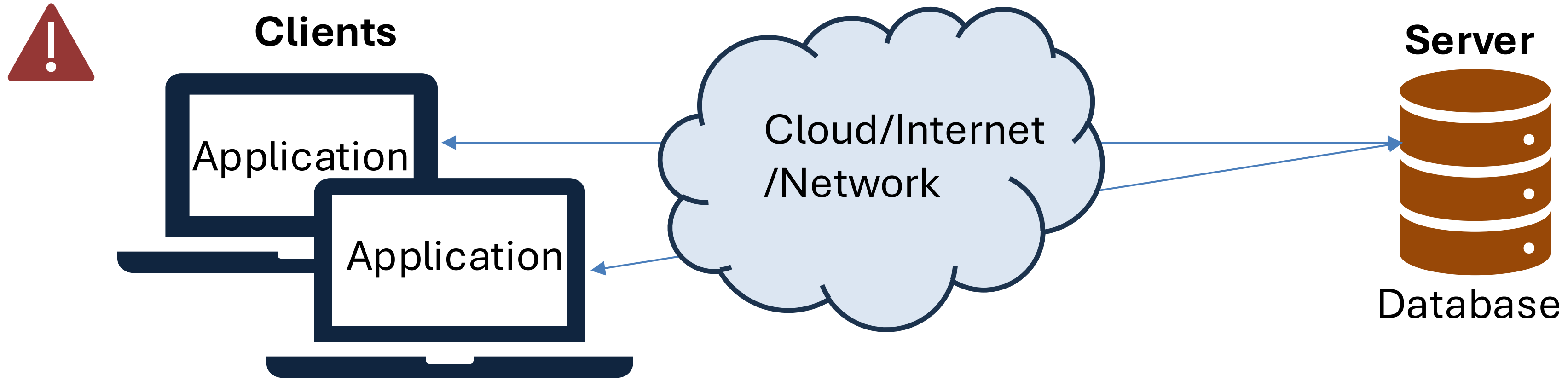
Web API

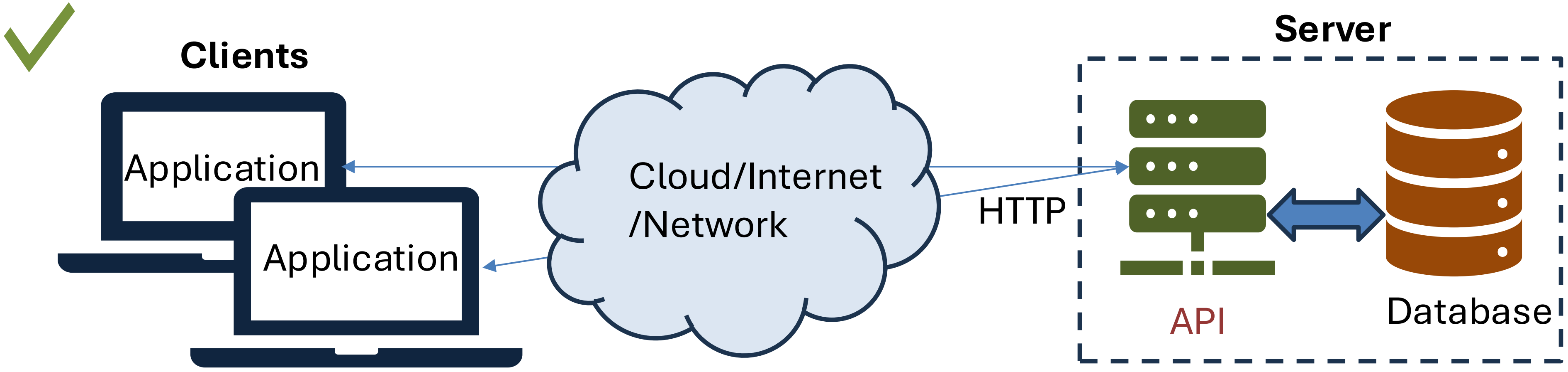
- We can create/use APIs for internal use inside an Application or between 2 or more Applications.
- Basically, an API can be just a Class with Methods that you use several places inside an Application or that you share between multiple Applications.
- A set of Stored Procedures in a Database can also 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.

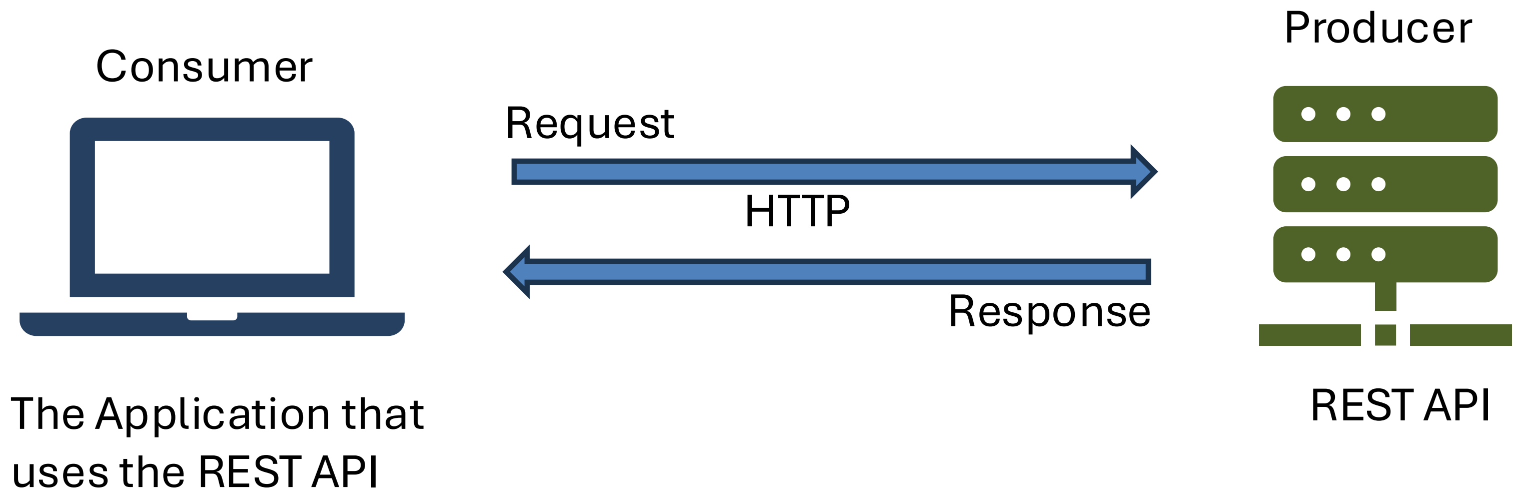




REST API

- REST APIs (also known as RESTful APIs) has been the standard when it comes to Web APIs.
- REST – is short for Representational State Transfer.
- REST APIs are based on the HTTP/HTTPS protocol.
- It is HTTP that controls all communication and traffic between web pages and your local browser.
- REST APIs can be made in all kind of Web Frameworks/Web Programming languages like PHP, ASP.NET, etc.
- You can also consume (use the API) in all types of Programming Languages like Python, C#, etc.

REST API


Consumer
Request
HTTP
Response
Producer
The Application that uses the REST API
REST API

HTTP/HTTPS

- HTTPS is not a separate protocol, but a combination of regular HTTP over an encrypted SSL (Secure Sockets Layer) or TLS (Transport Layer Security) connection.
- HTTP consists of different methods:
  - **GET** – This method is used to retrieve information from the server.
  - **POST** – This is used to send data to the server. Typically used to store data from a web page (an HTTML Form) to ,e.g., a database.
  - **PUT** – This is used to update information on the server.
  - **DELETE** – This is used to delete information on the server.
- You usually refer to these four methods as CRUD operations because they allow you to Create (POST), Read (GET), Update (PUT), and Delete (DELETE) resources, such as information in a database.

GET and **POST** are by far the most used of these HTTP methods

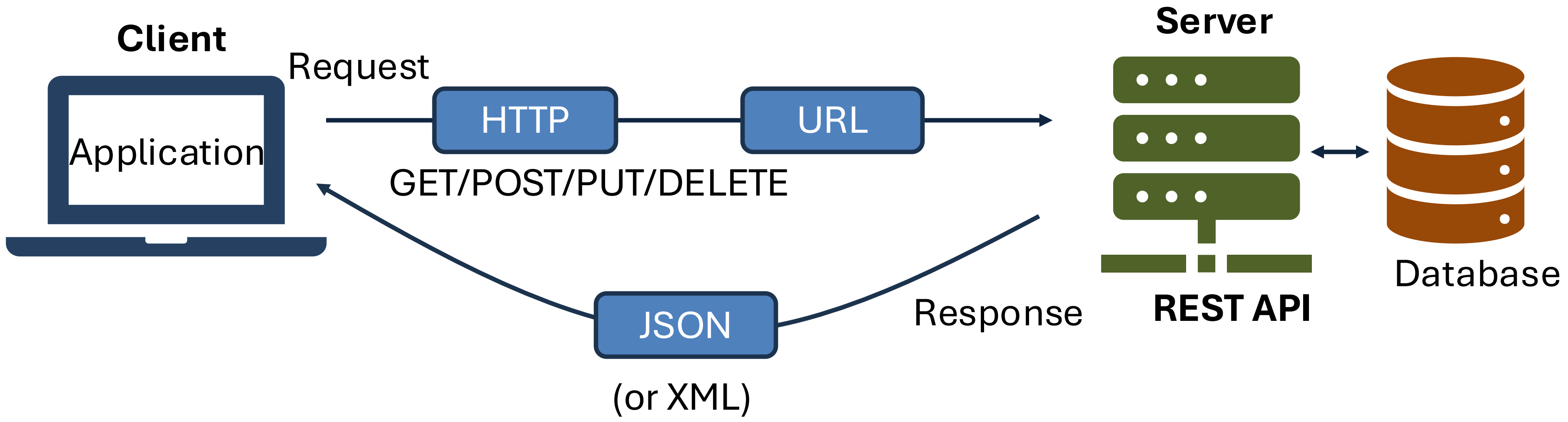
JSON

- 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

REST API


Client
Application
Request
HTTP
URL
GET/POST/PUT/DELETE
Server
Database
REST API
Response
JSON
(or XML)

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)

API Test Tools

- Postman
  Homepage: https://www.postman.com
- Insomnia
  Homepage: https://insomnia.rest

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 REST API Example

Hans-Petter Halvorsen

Example

- We will start by creating a Database and Table using MySQL.
- Then we will create the PHP code for the REST API.
- 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

Database

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

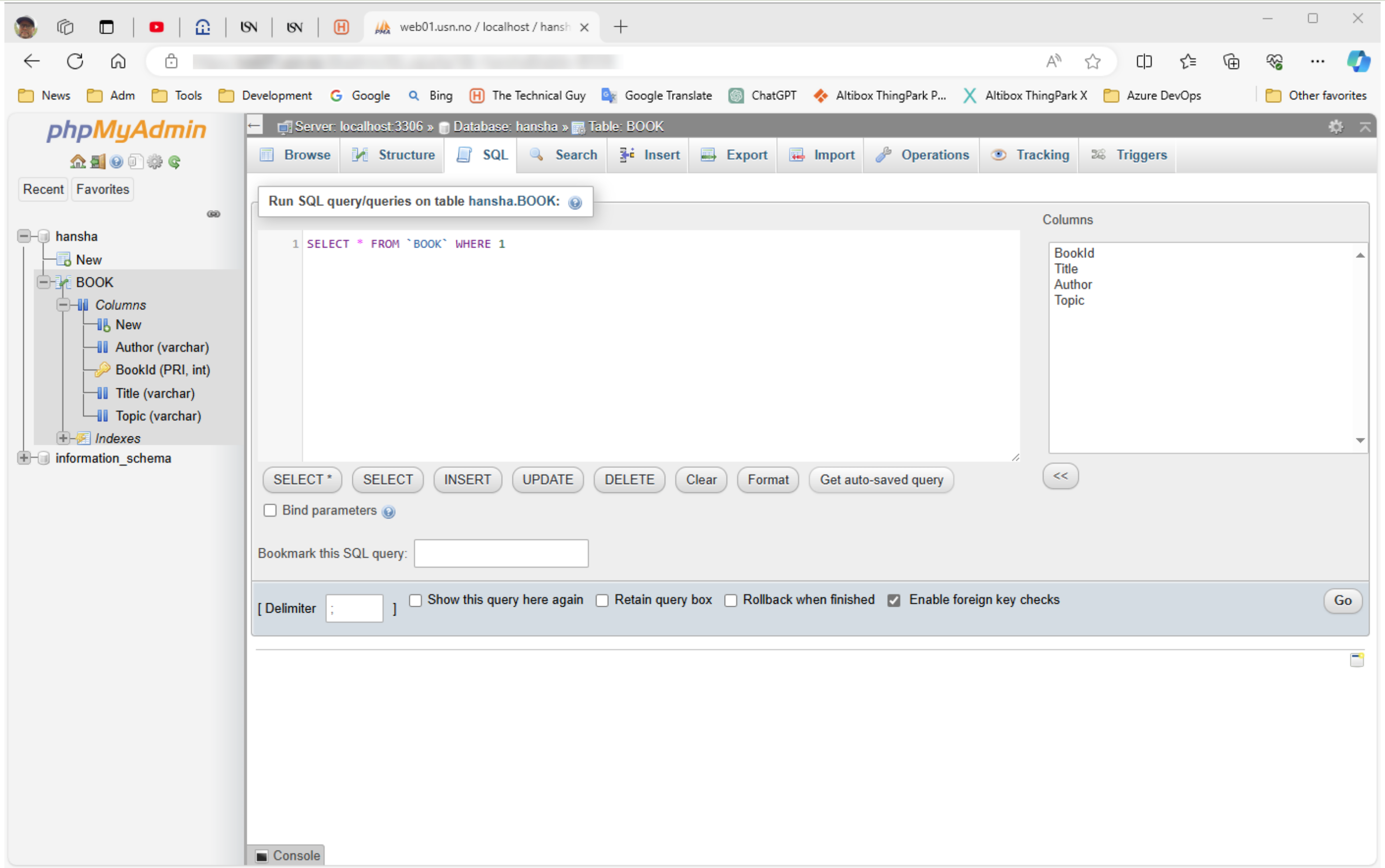
```
CREATE TABLE BOOK
(
    BookId int PRIMARY KEY AUTO_INCREMENT,
    Title varchar(100) NOT NULL,
    Author varchar(100) NOT NULL,
    Topic varchar(100) NOT NULL
);
```

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



Run SQL query/queries on table hansha.BOOK:

```
SELECT * FROM `BOOK` WHERE 1
```

Columns

BookId
Title
Author
Topic

Bind parameters

Bookmark this SQL query:

[Delimiter ;] Show this query here again Retain query box Rollback when finished [✓ Enable foreign key checks]

PHP

We can create 2 PHP files, e.g.:

- config.php
  - This file will contain username, password, etc. for the MySQL Server database.
- index.php
  - This file contains the REST API itself with GET, POST, PUT and DELETE functionality.

config.php

Connect to your Database:

```php
<?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

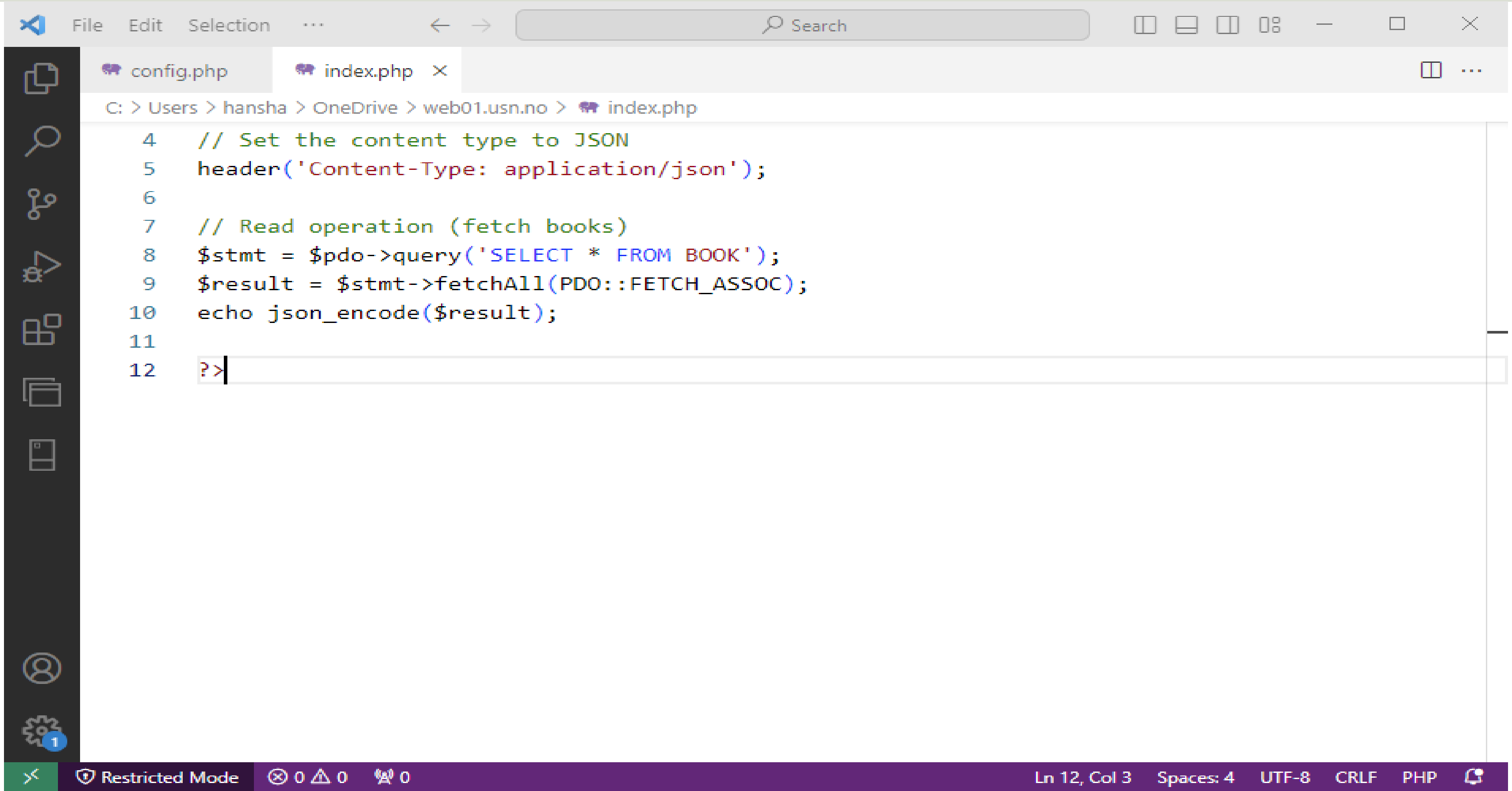
GET

This method is used to retrieve information from the server

index.php - GET

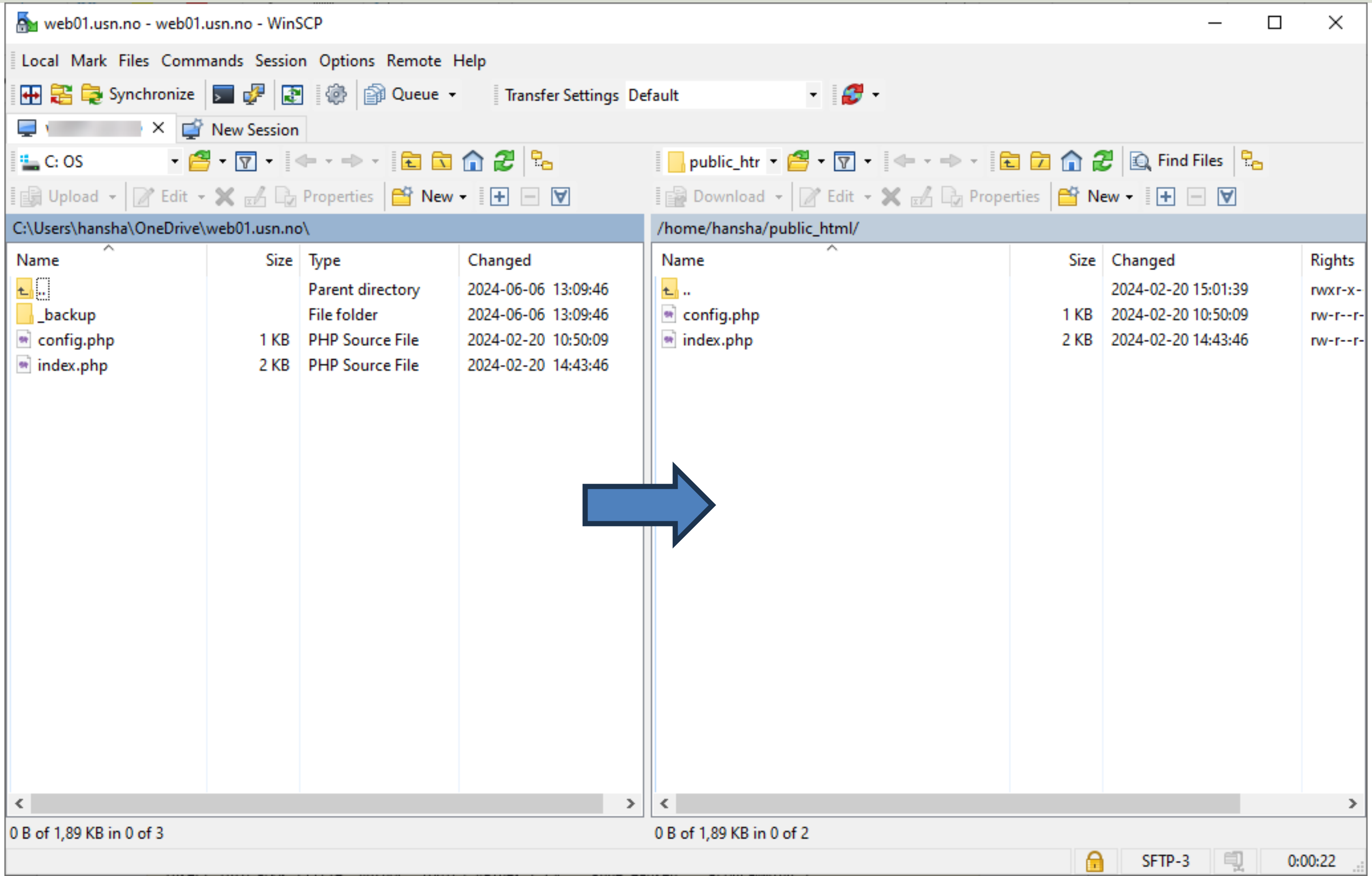
```php
<?php
require_once 'config.php';

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

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

Visual Studio Code



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

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

?>
```

WinSCP (FTP)



web01.usn.no - web01.usn.no - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
New Session
C: OS
public_htr
Find Files
Upload
Edit
Properties
New
Download

C:\Users\hansha\OneDrive\web01.usn.no\

Name	Size	Type	Changed
..		Parent directory	2024-06-06 13:09:46
_backup		File folder	2024-06-06 13:09:46
config.php	1 KB	PHP Source File	2024-02-20 10:50:09
index.php	2 KB	PHP Source File	2024-02-20 14:43:46

0 B of 1,89 KB in 0 of 3

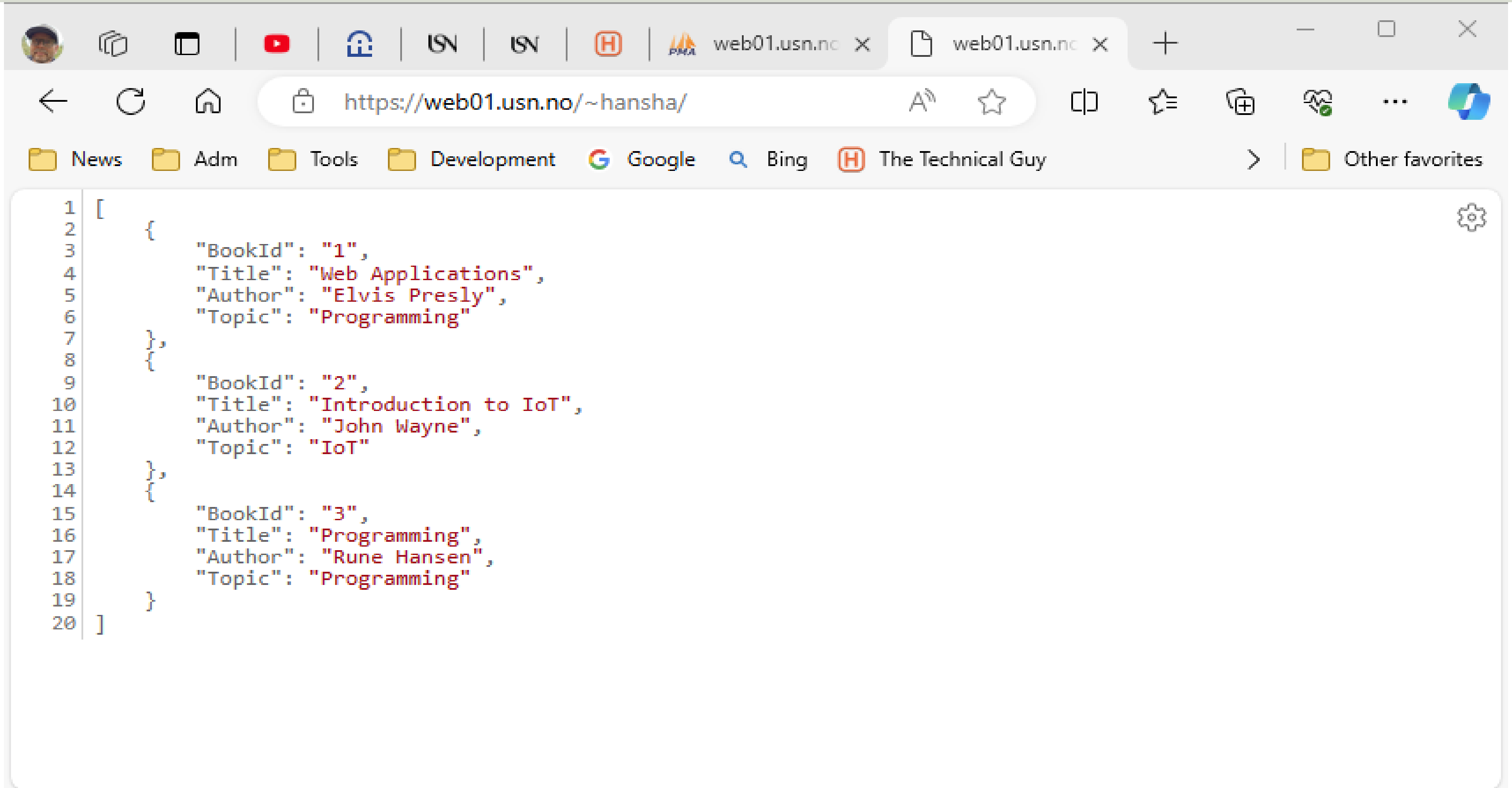
/home/hansha/public_html/

Name	Size	Changed	Rights
..		2024-02-20 15:01:39	rwxr-x-
config.php	1 KB	2024-02-20 10:50:09	rw-r--r-
index.php	2 KB	2024-02-20 14:43:46	rw-r--r-

0 B of 1,89 KB in 0 of 2

SFTP-3 0:00:22

Test in Browser

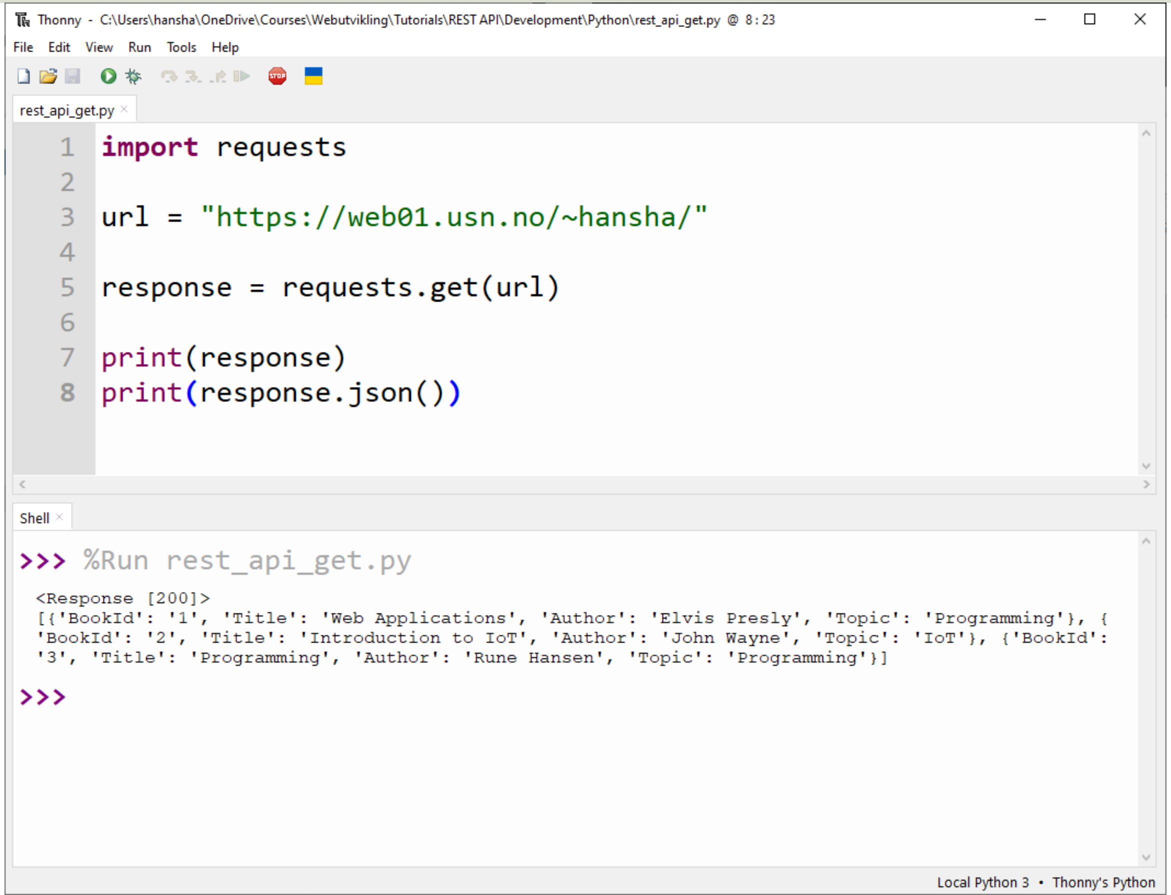


```
[
    {
        "BookId": "1",
        "Title": "Web Applications",
        "Author": "Elvis Presly",
        "Topic": "Programming"
    },
    {
        "BookId": "2",
        "Title": "Introduction to IoT",
        "Author": "John Wayne",
        "Topic": "IoT"
    },
    {
        "BookId": "3",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

Python - GET

```
import requests

url = "https://web01.usn.no/~hansha/"

response = requests.get(url)

print(response)
print(response.json())
```

Thonny – Running GET Script


Thonny - C:\Users\hansha\OneDrive\Courses\Webutvikling\Tutorials\REST API\Development\Python\rest_api_get.py @ 8 : 23
File Edit View Run Tools Help
rest_api_get.py
Shell
Local Python 3 • Thonny's Python

```python
import requests

url = "https://web01.usn.no/~hansha/"

response = requests.get(url)

print(response)
print(response.json())
```

```
>>> %Run rest_api_get.py
  <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': 'Programming', 'Author': 'Rune Hansen', 'Topic': 'Programming'}]

>>>
```

We prepare for POST, etc. by creating a switch statement

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

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

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

switch ($method) {
 case 'GET':
 // Read operation (retrieve books)
 $stmt = $pdo->query('SELECT * FROM BOOK');
 $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
 echo json_encode($result);
 break;

 default:
 // Invalid method
 http_response_code(405);
 echo json_encode(['error' => 'Method not allowed']);
 break;
}
?>
```

POST

This method is used to <u>send</u> data to the server

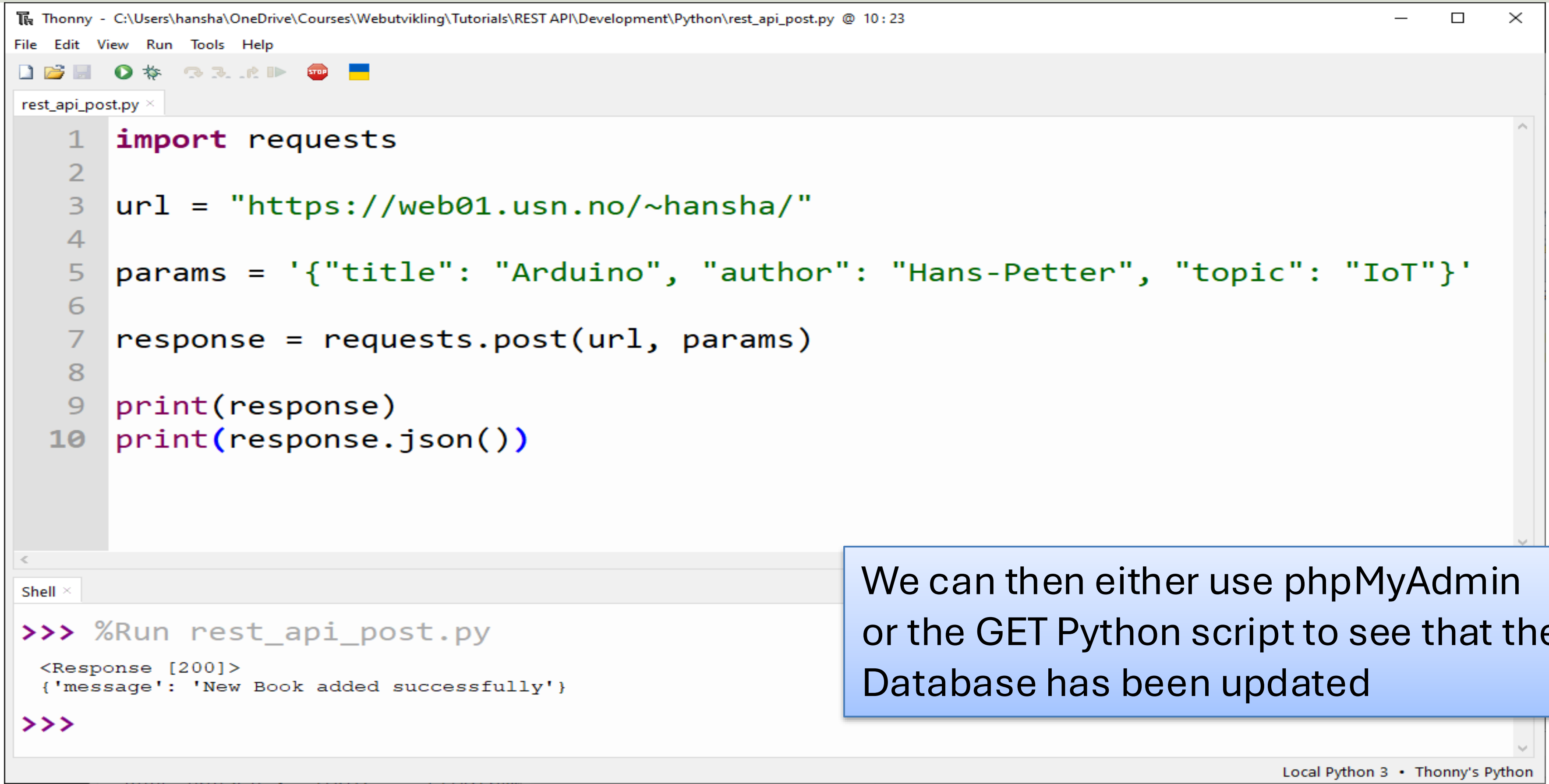
PHP - POST

```
$method = $_SERVER['REQUEST_METHOD'];
..
case 'POST':
 // Create operation (add a new book)
 $json = file_get_contents('php://input');
 $data = json_decode($json,true);
 $title = $data['title'];
 $author = $data['author'];
 $topic = $data['topic'];

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

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

Python - POST

```
import requests

url = "https://web01.usn.no/~hansha/"

params = '{"title": "Arduino", "author": "Hans-Petter",
"topic": "IoT"}'

response = requests.post(url, params)

print(response)
print(response.json())
```

Running Python in Thonny editor


Thonny - C:\Users\hansha\OneDrive\Courses\Webutvikling\Tutorials\REST API\Development\Python\rest_api_post.py @ 10 : 23
File Edit View Run Tools Help
rest_api_post.py
Shell
Local Python 3 • Thonny's Python

```python
import requests

url = "https://web01.usn.no/~hansha/"

params = '{"title": "Arduino", "author": "Hans-Petter", "topic": "IoT"}'

response = requests.post(url, params)

print(response)
print(response.json())
```

```
>>> %Run rest_api_post.py
  <Response [200]>
  {'message': 'New Book added successfully'}
>>>
```

We can then either use phpMyAdmin or the GET Python script to see that the Database has been updated

PUT

This method is used to <u>update</u> information on the server

PHP - PUT

Note! Your Apache/PHP Server may have disabled the PUT method for security reasons.

```
$method = $_SERVER['REQUEST_METHOD’];
..
case 'PUT':
 // Update operation (edit a book)
 $json = file_get_contents('php://input');
 $data = json_decode($json,true);
 $id = $data['id'];
 $title = $data['title'];
 $author = $data['author'];
 $topic = $data['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']);
break;
```

Python - PUT

```
import requests

url = "https://web01.usn.no/~hansha/"

headers = {
  "User-Agent": "",
  "Content-Type": "application/json"
}

data = '{"id": "28", "title": "Arduino3", "author": "Hans-Petter",
"topic": "IoT"}'

response = requests.put(url, headers=headers, data=data)

print(response)
print(response.json())
```

DELETE

This method is used to delete information on the server

PHP - DELETE

Note! Your Apache/PHP Server may have disabled the DELETE method for security reasons.

```
$method = $_SERVER['REQUEST_METHOD'];
..
case 'DELETE':
 // Delete operation (remove a book)
 $json = file_get_contents('php://input');
 $data = json_decode($json,true);
 $id = $data['id'];

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

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

Python - DELETE

```
import requests

url = "https://web01.usn.no/~hansha/"

headers = {
  "User-Agent": "",
  "Content-Type": "application/json"
}

data = '{"id": "5"}'

response = requests.delete(url, headers=headers, data=data)

print(response)
print(response.json())
```

Summary

- We have created a simple REST API using PHP.
- We tested the REST 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, access control, etc. But I leave that to you to improve.
- The code is made simple to illustrate the basic principles creating and using REST 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