

Introduction to WebAPI

LUKÁŠ VAVREK

Outline

HTTP Protocol

REST

.Net Core WebAPI

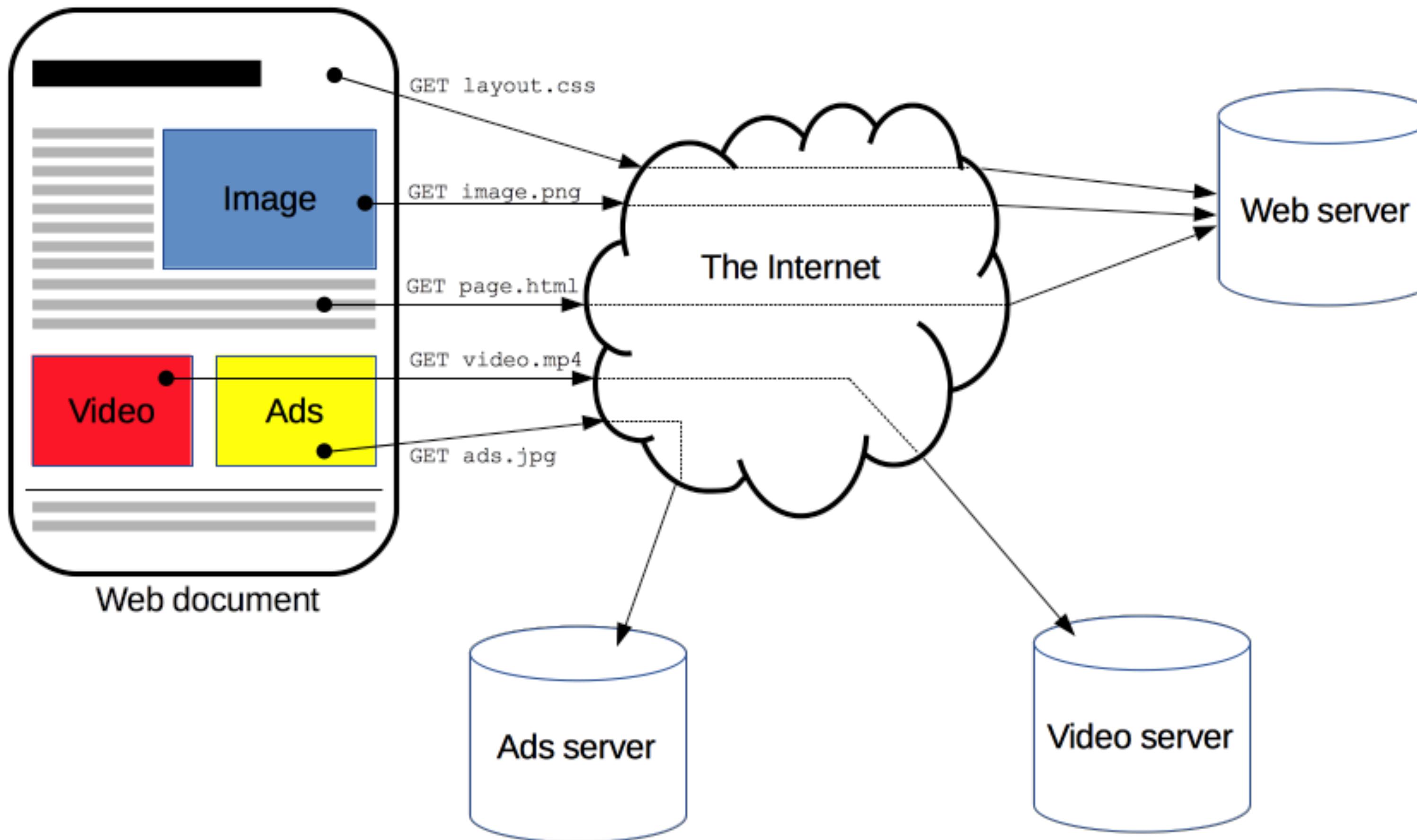
HTTP Protocol



HTTP Protocol

HTTP is a protocol which allows the **fetching of resources**, such as HTML documents. It is the foundation of any data exchange on the Web and a **client-server** protocol, which means **requests are initiated by the recipient**, usually the Web browser.

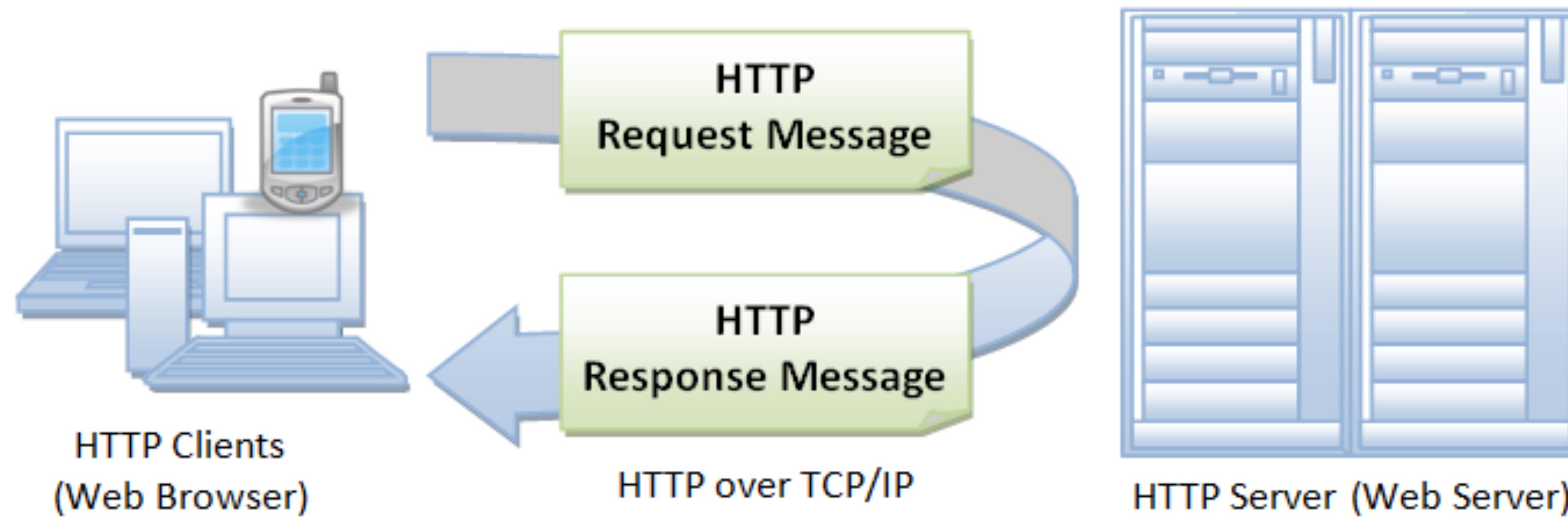
Querying resources



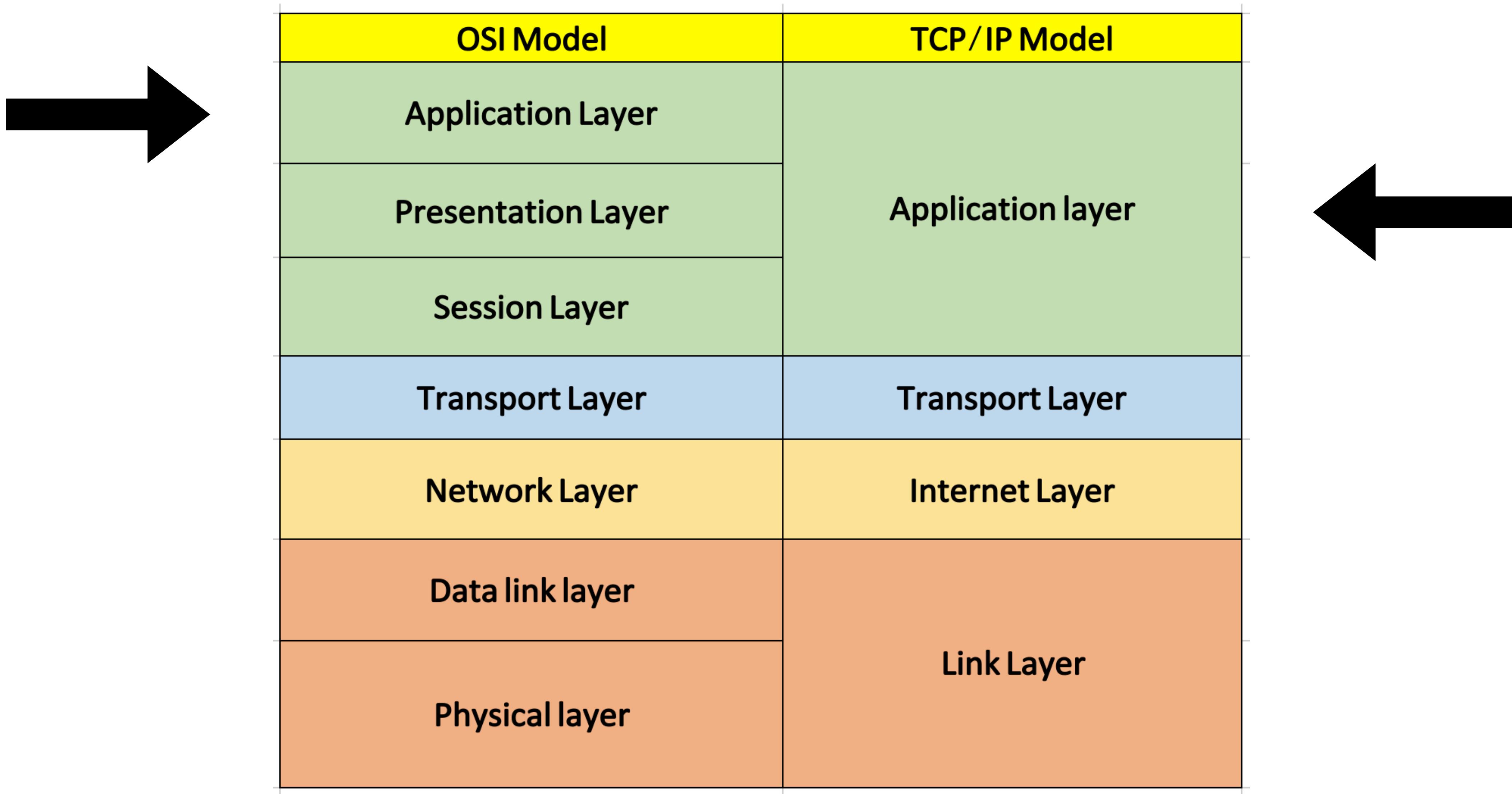
Main Characteristics

- ⟩ Request - response
- ⟩ Client - server
- ⟩ Application layer
- ⟩ Uses TCP at transport layer (mostly)

Client / Server Request / Response



Layered architecture



URI

host

`https://www.wikipedia.org/index.html`

scheme

path

Communication



Status Codes

- ⟩ Issued by a server in response to a client's request
 - ⟩ First digit specifies standard response class
-
- ⟩ 1xx (Informational):
The request was received, continuing process
 - ⟩ 2xx (Successful):
The request was successfully received, understood, and accepted
 - ⟩ 3xx (Redirection):
Further action needs to be taken in order to complete the request
 - ⟩ 4xx (Client Error):
The request contains bad syntax or cannot be fulfilled
 - ⟩ 5xx (Server Error):
The server failed to fulfill an apparently valid request

CRUD Operations

➤ GET

➤ POST

➤ PUT

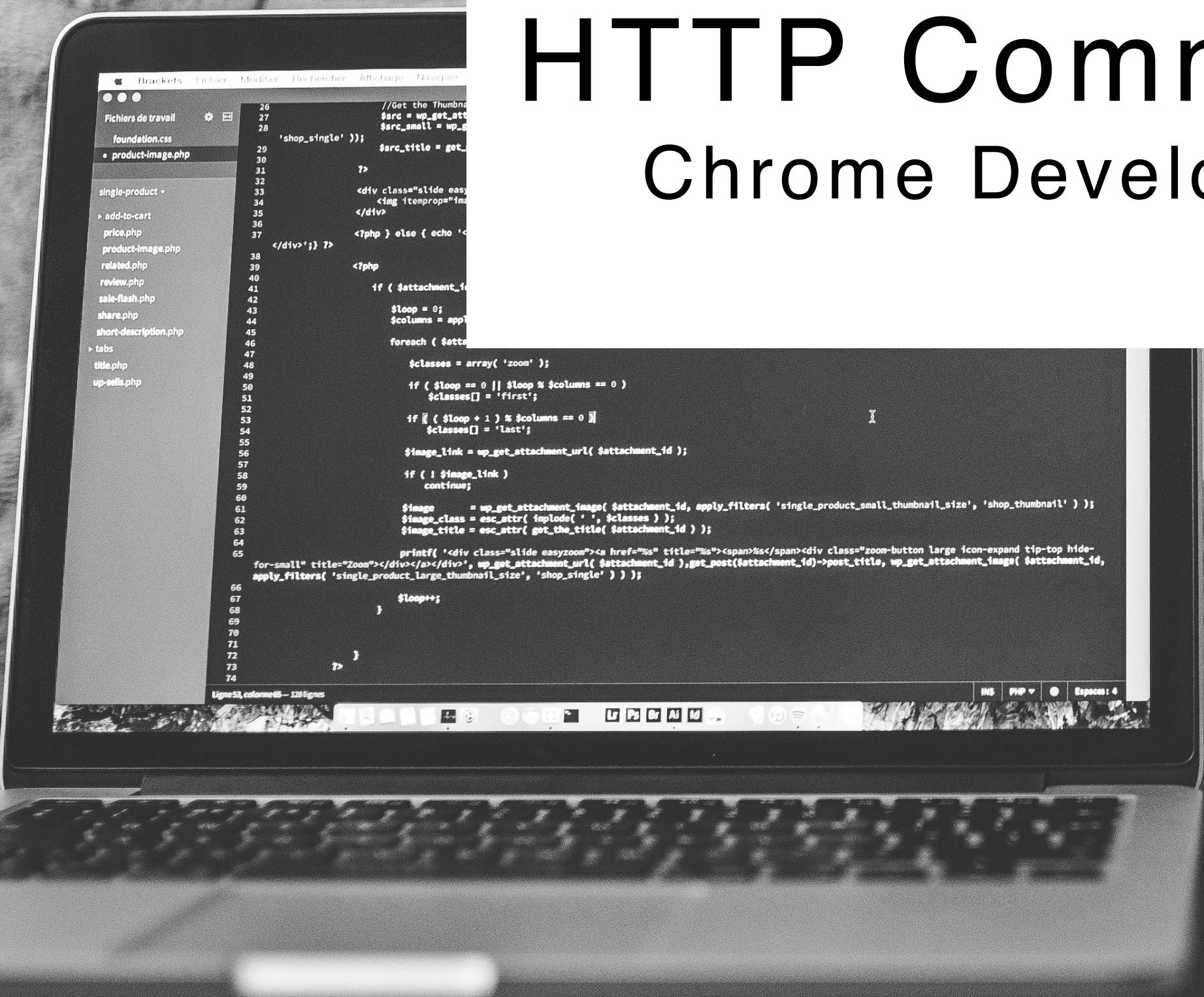
➤ PATCH

➤ DELETE

DEMO

HTTP Communication

Chrome Developer Console



REST



Definition

REST (Representational State Transfer) is a **software architectural style** that defines a **set of constraints** to be used for creating Web services. REST is usually used over **HTTP**.

APIs should be **client-server** architecture oriented, **stateless**, **support caching**, have **uniform interface**, be layered and support code on demand.

API Example

GET /todo-items

GET /todo-items/{item id}

POST /todo-items

PUT /todo-items/{item id}

DELETE /todo-items/{item id}

REST Client

- ⟩ API testing / debugging
- ⟩ Easy and simple
- ⟩ Build requests quickly

Insomnia

The screenshot shows the Insomnia REST client interface. The top bar displays the title "Insomnia – Auth - Login" and the status "200 OK". The URL is "https://glnetworkingserver.azurewebsites.net/api/authentication/login". The response time is 989 ms and the size is 259 B.

The left sidebar shows a tree view of API endpoints under the "GLNetworkingServer" folder, including "Authentication", "Auth - Login", "Auth - Get Time - Authori...", "StatusCode-AsYouWish", and "StatusCode-Probability".

The main panel shows the request configuration: Method "POST", URL "https://glnetworkingserver.azurewebsites.net/api/authentication/login", Headers (Content-Type: application/json), and Body (JSON):

```
1 {  
2   "username": "asdf",  
3   "password": "qwer"  
4 }
```

The response pane shows the JSON response:

```
1 {  
2   "token":  
3     "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcyc54bWxzb2FwLm9yZy93cy8yM  
DA1LzA1L2lkZW50aXR5L2NsYWltcy9uYnliIjoiIwiZXhwIjoxNTIzMjE4LCJpc3MiOiJnbG5ldHdv  
cmtpbmcilCJhdWQiOjnbG5ldHdvcmtpbmcifQ.s1WF7t5QJBnmkRAJee9wgz6g3y5cTYVaoRVoQBde64"  
3 }
```

Postman

The screenshot shows the Postman application interface. The top navigation bar includes 'NEW', 'Runner', 'Import', 'Builder' (which is selected), 'Team Library', and various status indicators. The left sidebar displays a 'History' section for today, listing multiple failed requests to 'https://colony7.blob.core.windows.net/folder-customer-profilepic'. The main workspace shows a 'PUT' request to the same URL. The 'Headers' tab is selected, showing the following configuration:

Key	Value	Description
x-ms-version	2015-12-11	
x-ms-date	Thu, 19 Oct 2017 15:27:05 GMT	
Content-Type	image/jpeg	
x-ms-blob-content-disposition	attachment	
x-ms-blob-type	BlockBlob	
Authorization	SharedKey colony7:rlJ8rhUmvlqfgdWgk2BCMvyO...	
x-ms-meta-m1	v1	
x-ms-meta-m2	v2	

The 'Body' tab shows the XML response from the server, indicating an authentication failure:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Error>
3   <Code>AuthenticationFailed</Code>
4   <Message>Server failed to authenticate the request. Make sure the value of Authorization header is formed correctly including the signature.
5 RequestId:171221ef-001e-004f-08ef-487884000000
6 Time:2017-10-19T15:30:33.6850313Z</Message>
7   <AuthenticationErrorDetail>The MAC signature found in the HTTP request 'rlJ8rhUmvlqfgdWgk2BCMvyO1iyh5wxXfiWkyabrtcs='
8     is not the same as any computed signature. Server used following string to sign: 'PUT
9
10 355122
11
```

iTunes API

Working with iTunes

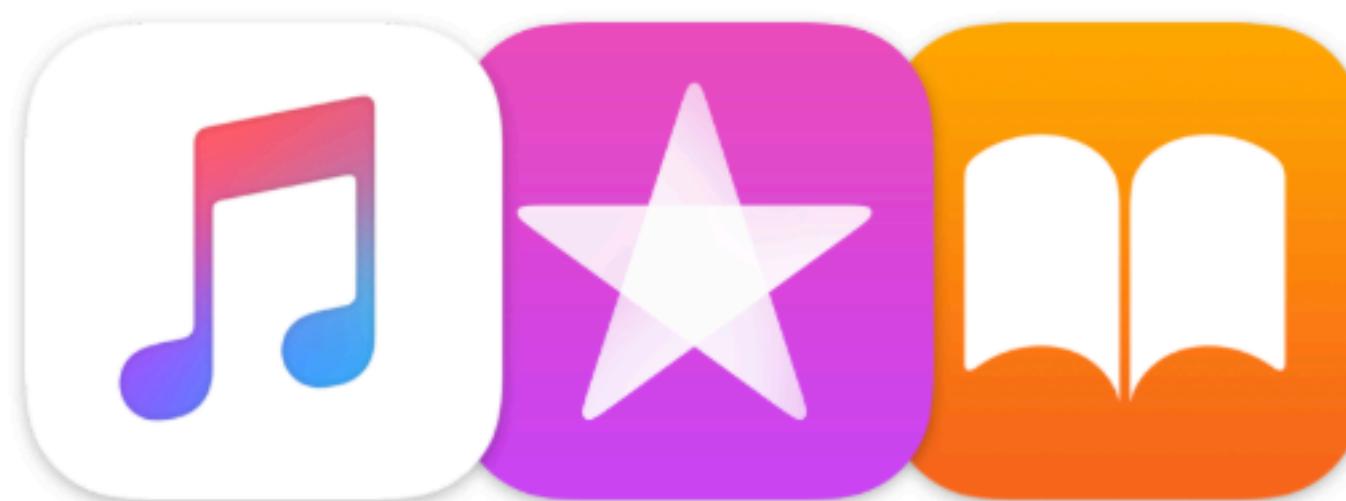
Sell Your Content

Enhance Your Content

Market with iTunes

Link to iTunes

iTunes Affiliate Resources



iTunes Search API

Table of Contents

- [Overview](#)
- [Searching the iTunes Store](#)
- [Notes](#)
- [Search Examples](#)
- [Lookup Examples](#)
- [Understanding Search Results](#)
- [Caching Architecture](#)
- [Legal](#)

DEMO

Fetch music data iTunes API



```
function fetchMusicData($query) {
    $url = "https://itunes.apple.com/search?term=" . urlencode($query);
    $response = file_get_contents($url);
    $data = json_decode($response, true);
    return $data['results'];
}

// Example usage
$musicData = fetchMusicData("jazz");
print_r($musicData);
```



.NET Core WebAPI



The host

WebHost encapsulates all of the app's resources, such as:

- * An HTTP server implementation
- * Middleware components
- * Logging
- * Dependency injection
- * Configuration

Web Server

ASP.NET Core app uses an **HTTP server** implementation to listen for HTTP requests. The server surfaces requests to the app as a set of request features composed into an **HttpContext**.

By default it uses Kestrel.

Configuration

ASP.NET Core provides a configuration framework that gets settings as name-value pairs from an ordered set of configuration providers.

Built-in support for json, xml, environment variables and command-line arguments.

Startup

First entry of our web application code. It is a great place to register dependences, configure application, setup middleware pipeline and prepare application for request processing.

DEMO

Blank WebAPI project

ASP.NET WebAPI

DIAKK WEB

ASP.NE

Brackets Editor

Fichiers de travail

- foundation.css
- product-image.php

single-product

> add-to-cart

price.php

product-image.php

related.php

review.php

sale-flash.php

share.php

short-description.php

> tabs

title.php

up-sells.php

```
26             //Get the Thumbnail
27             $src = wp_get_attachment_image_src( $attachment_id, 'shop_thumbnail' );
28             $src_small = wp_get_attachment_image_src( $attachment_id, 'single_product_small_thumbnail_size' );
29             $src_title = get_the_title( $attachment_id );
30         ?>
31     </div>
32     <div class="slide easyzoom-for-small" title="Zoom"><a href="#" title="Zoom"><span>%s</span></a></div>
33     <div class="zoom-button large icon-expand tip-top hide-for-small" title="Zoom"></div></div>;
34     wp_get_attachment_url( $attachment_id ), get_post( $attachment_id )->post_title, wp_get_attachment_image( $attachment_id, apply_filters( 'single_product_large_thumbnail_size', 'shop_single' ) ) );
35
36     if ( ! $image_link )
37     echo '<?php } else { echo ''';
38
39     if ( $attachment_id )
40     {
41         $loop = 0;
42         $columns = apply_filters( 'single_product_columns', 3 );
43
44         foreach ( $attachments as $attachment )
45         {
46             $classes = array( 'zoom' );
47
48             if ( $loop == 0 || $loop % $columns == 0 )
49                 $classes[] = 'first';
50
51             if ( ( $loop + 1 ) % $columns == 0 )
52                 $classes[] = 'last';
53
54             $image_link = wp_get_attachment_url( $attachment_id );
55
56             if ( ! $image_link )
57                 continue;
58
59             $image      = wp_get_attachment_image( $attachment_id, apply_filters( 'single_product_small_thumbnail_size', 'shop_thumbnail' ) );
60             $image_class = esc_attr( implode( ' ', $classes ) );
61             $image_title = esc_attr( get_the_title( $attachment_id ) );
62
63             printf( '<div class="slide easyzoom"><a href="#" title="%s"><span>%s</span></a><div class="zoom-button large icon-expand tip-top hide-for-small" title="Zoom"></div></a></div>', wp_get_attachment_url( $attachment_id ), get_post( $attachment_id )->post_title, wp_get_attachment_image( $attachment_id, apply_filters( 'single_product_large_thumbnail_size', 'shop_single' ) ) );
64
65             $loop++;
66
67         }
68
69     }
70
71     ?>
72
73
74
```

Ligne 53, colonne 65 - 128 lignes

INS PHP Espaces 4



Q & A

