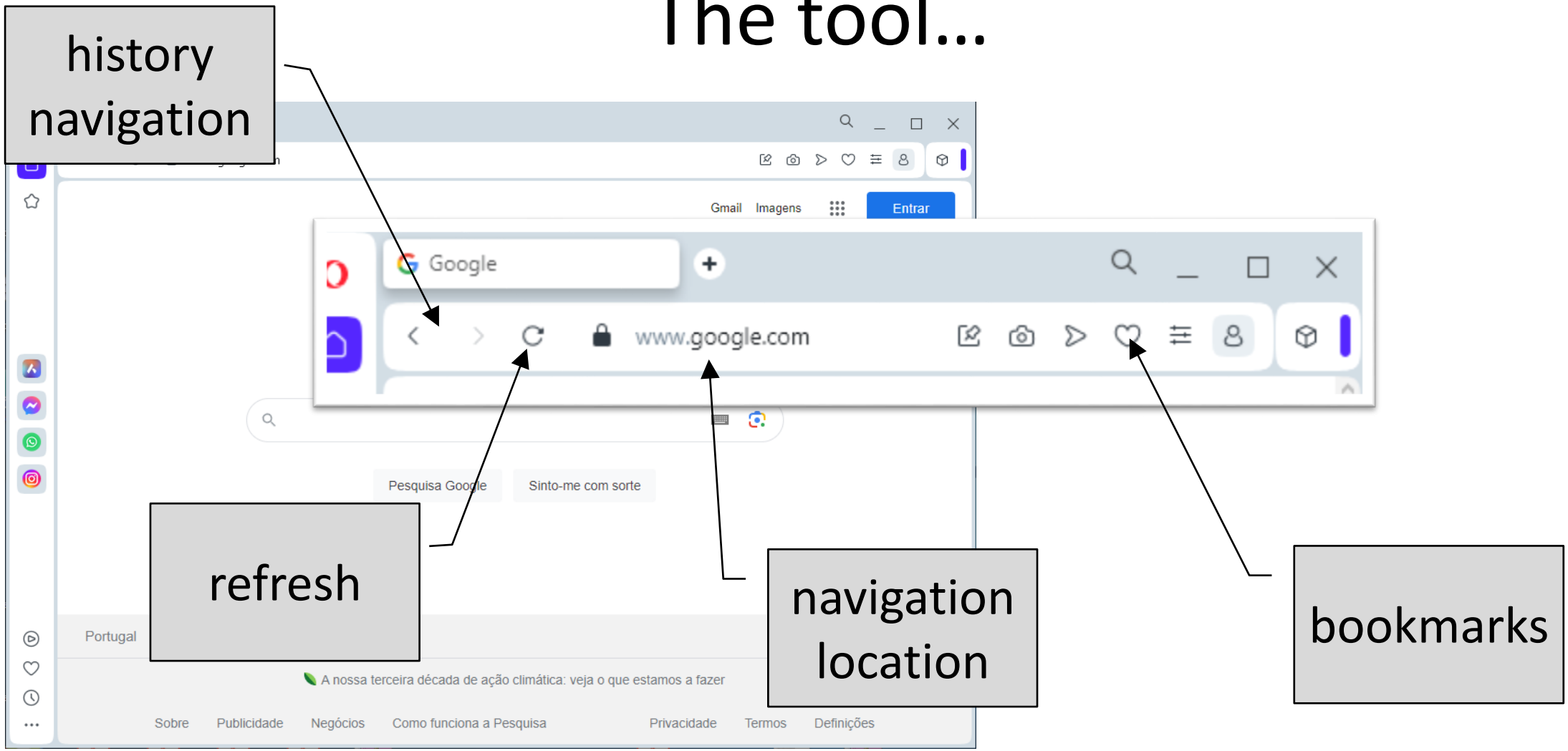
Request-response sequence of a web page

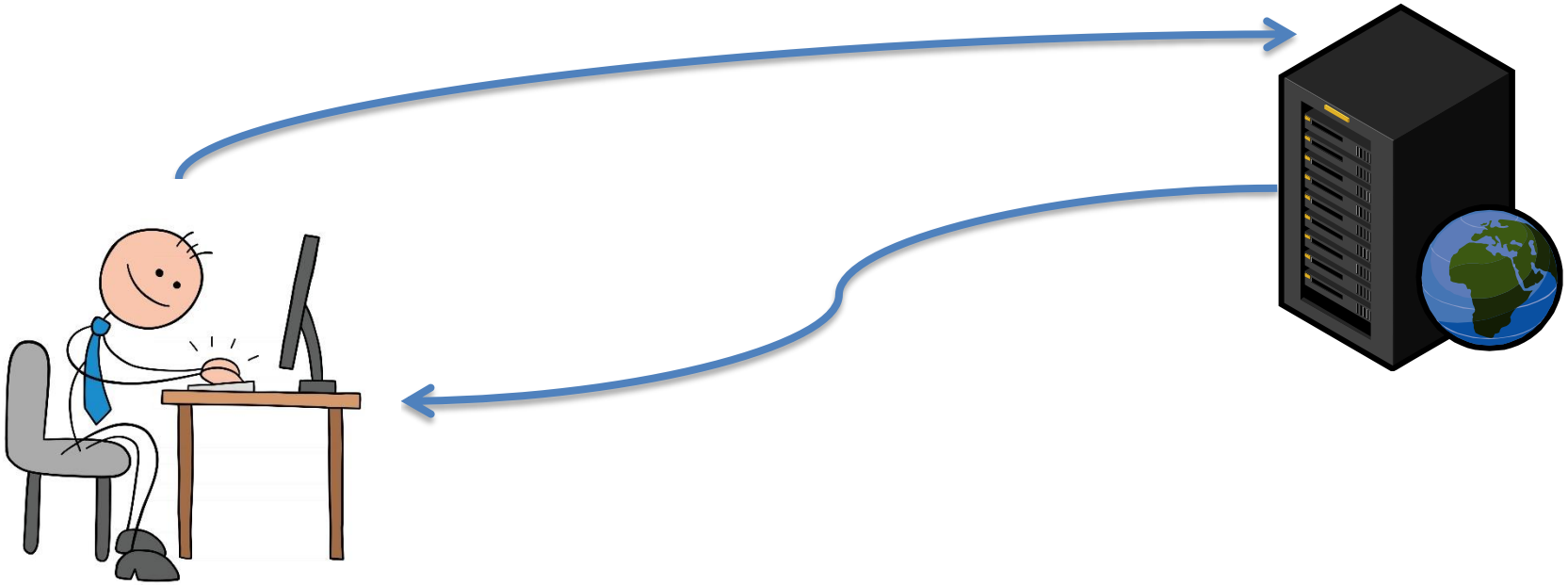
Web Engineering

utad UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

The tool…


history navigation
Gmail
Imagens
Entrar
Google
www.google.com
Pesquisa Google
Sinto-me com sorte
refresh
navigation location
bookmarks
Portugal
A nossa terceira década de ação climática: veja o que estamos a fazer
Sobre
Publicidade
Negócios
Como funciona a Pesquisa
Privacidade
Termos
Definições

Browse a website…
…how it works?

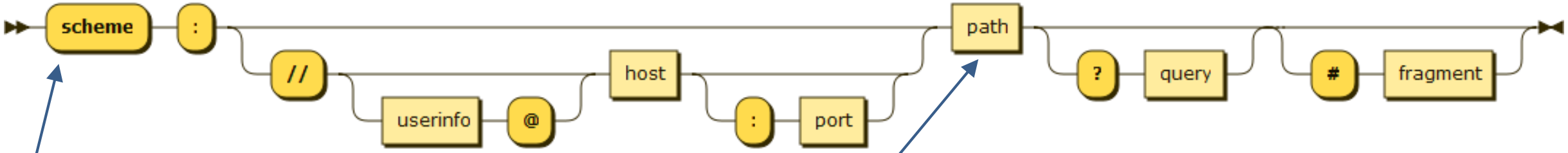


http://www.diocese-vilareal.pt/a-diocese.html

What is a URL?

Universal Resource Locator

```
scheme:[//[user:password@]host[:port]][path][?query][#anchor]
```



http,
https,
ftp,
mailto,
file,
data,
and irc

- URLs can only be sent over the Internet using the ASCII character-set.
- Since URLs often contain characters outside the ASCII set, the URL has to be converted into a valid ASCII format.
- URL encoding replaces unsafe ASCII characters with a "%" followed by two hexadecimal digits.
- URLs cannot contain spaces. URL encoding normally replaces a space with a plus (+) sign or with %20.

Universal Resource Locator

Examples:

https://www.sapo.pt/

ftp://home.utad.pt/~lfb

https://www.continente.pt/pt-pt/public/Pages/searchresults.aspx?k=arroz

https://www.fnac.pt/livro/h5#bl=MMlivros

https://realvitur.pt/pesquisa/Mal%C3%A1sia

url enconded
From UTF-8
(default character-set in HTML5)

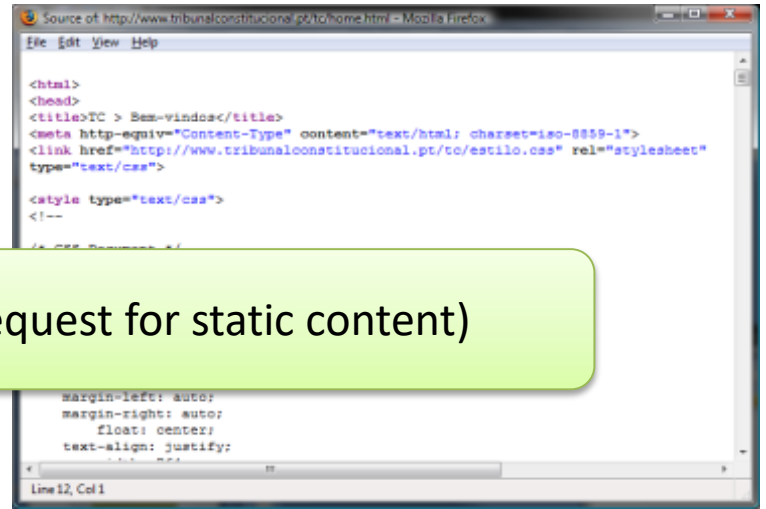
https://www.w3.org/Protocols/rfc2616/rfc2616-sec2.html#sec2.2

Browsing a website…

1. The user inserts http://www.diocese-vilareal.pt/a-diocese.html in browser

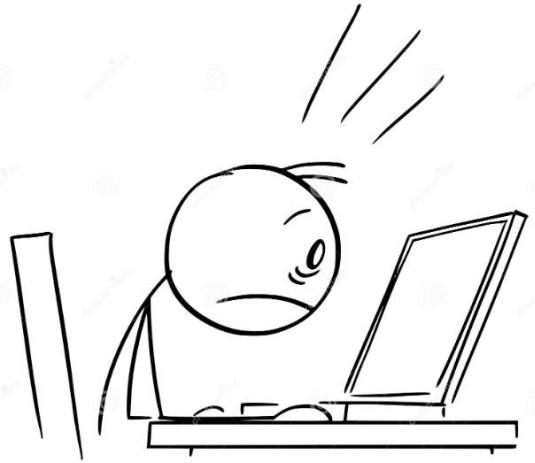
2. The browser does a HTTP request to the Web server, asking for the page http://www.diocese-vilareal.pt/a-diocese.html

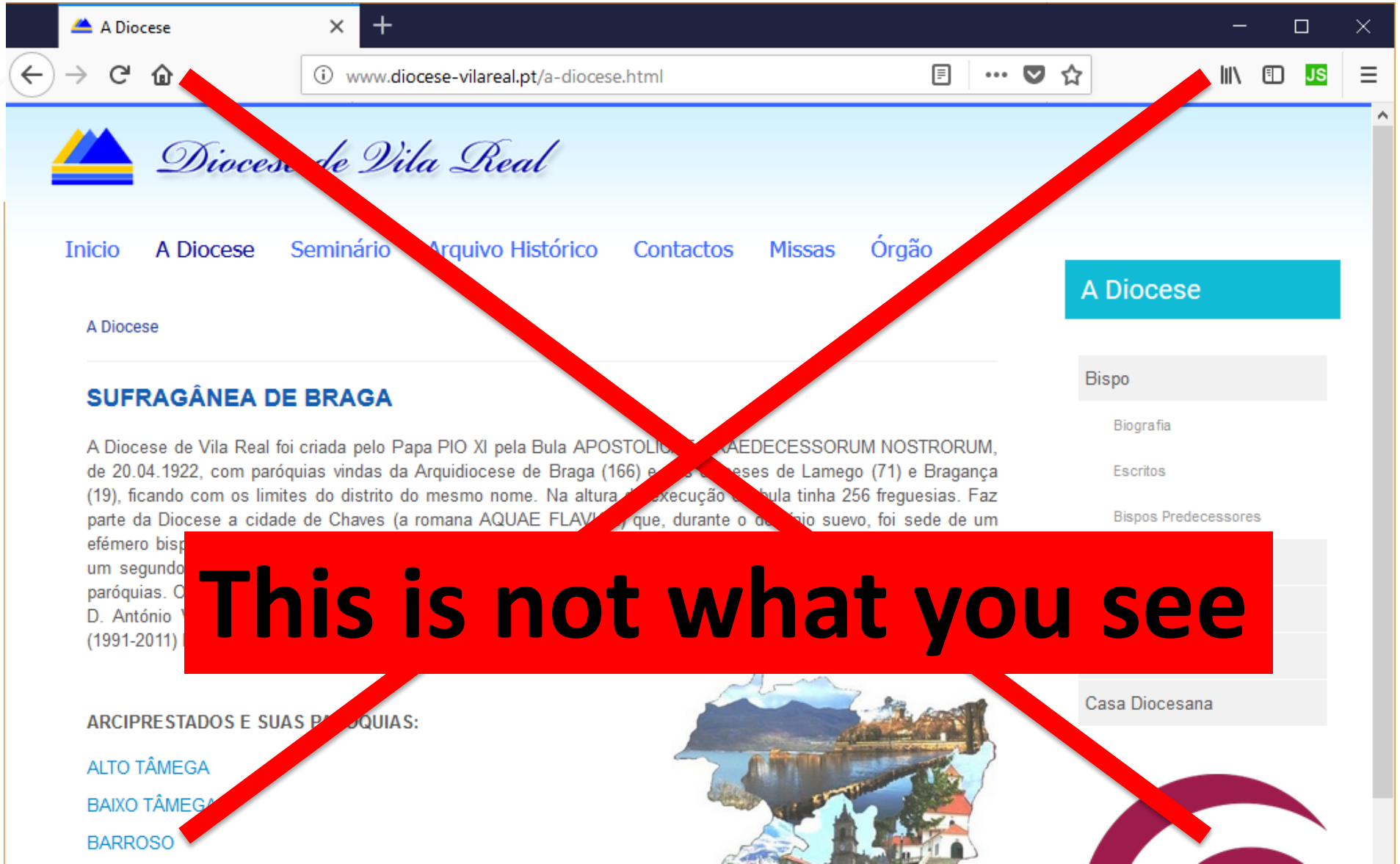
3. The server opens the local file **a-diocese.html** and returns it as part of the HTTP response.



(Request for static content)

Expected result…





This is not what you see

This is what you see

A Diocese

www.diocese-vilareal.pt/a-diocese.html

- Inicio
- A Diocese
- Seminário
- Arquivo Histórico
- Contactos
- Missas
- Órgão

A Diocese

SUFRAGÂNEA DE

A Diocese de Vila Rea
vindas da Arquidiocese de Bra
execução da bula tinha 256 freguesias.
um efémero bispado flaviense de que terá sido p
(Boticas?). A superfície da Diocese é de 4.273,20 Km² e te
Vidal (1923-1933), seguindo-se o Senhor D. António Valente da Fo
(1991-2011) E D. Amândio José Tomás (2011 A ...)

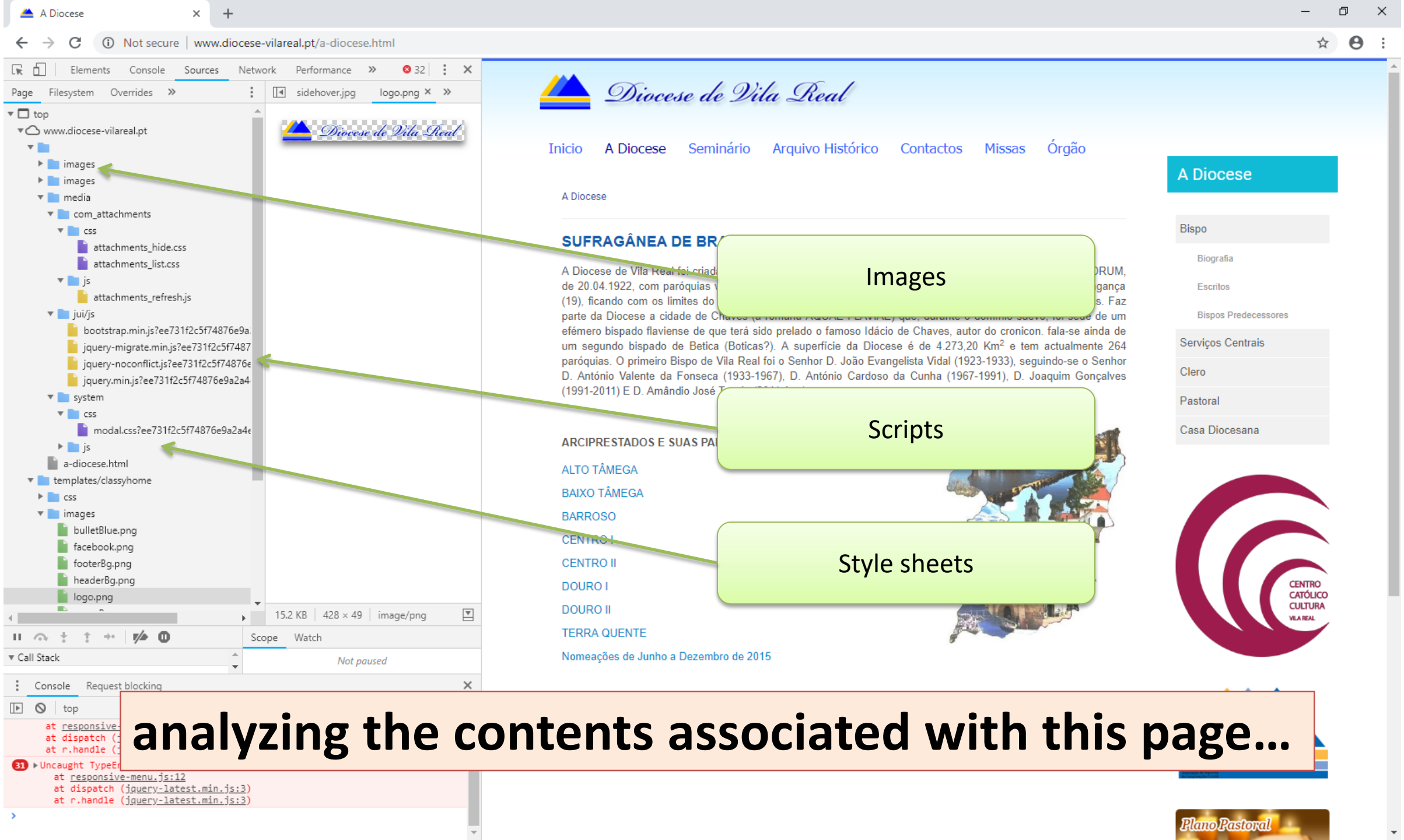
ARCIPRESTADOS E SUAS PARÓQUIAS:

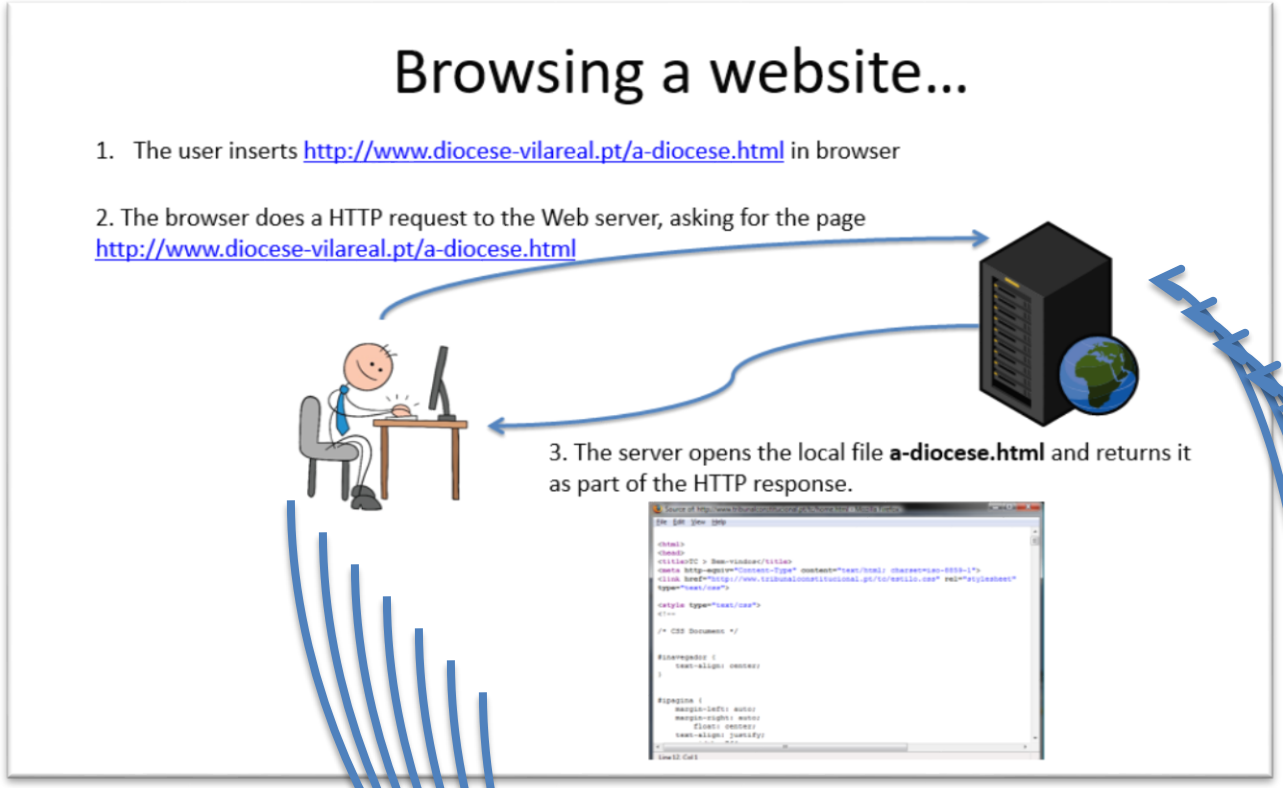
ALTO TÂMEGA

BAIXO TÂMEGA

BARROSO

Why?


A Diocese
Not secure | www.diocese-vilareal.pt/a-diocese.html
Elements Console Sources Network Performance
Page Filesystem Overrides
sidehover.jpg logo.png
top
www.diocese-vilareal.pt
images
images
media
com_attachments
css
attachments_hide.css
attachments_list.css
js
attachments_refresh.js
jui/js
bootstrap.min.js?ee731f2c5f74876e9a
jquery-migrate.min.js?ee731f2c5f7487
jquery-noconflict.js?ee731f2c5f74876e
jquery.min.js?ee731f2c5f74876e9a2a4
system
css
modal.css?ee731f2c5f74876e9a2a4e
js
a-diocese.html
templates/classyhome
css
images
bulletBlue.png
facebook.png
footerBg.png
headerBg.png
logo.png
15.2 KB 428 × 49 image/png
Scope Watch
Call Stack
Not paused
Console Request blocking
top
Uncaught TypeE
at responsive-menu.js:12
at dispatch (jquery-latest.min.js:3)
at r.handle (jquery-latest.min.js:3)
Diocese de Vila Real
Inicio A Diocese Seminário Arquivo Histórico Contactos Missas Órgão
A Diocese
SUFRAGÂNEA DE BR
A Diocese de Vila Real foi criad
de 20.04.1922, com paróquias
(19), ficando com os limites do
parte da Diocese a cidade de Ch
efémero bispado flaviense de que terá sido prelado o famoso Idácio de Chaves, autor do cronicon. fala-se ainda de
um segundo bispado de Betica (Boticas?). A superfície da Diocese é de 4.273,20 Km² e tem actualmente 264
paróquias. O primeiro Bispo de Vila Real foi o Senhor D. João Evangelista Vidal (1923-1933), seguindo-se o Senhor
D. António Valente da Fonseca (1933-1967), D. António Cardoso da Cunha (1967-1991), D. Joaquim Gonçalves
(1991-2011) E D. Amândio José
ARCIPRESTADOS E SUAS PA
ALTO TÂMEGA
BAIXO TÂMEGA
BARROSO
CENTRO I
CENTRO II
DOURO I
DOURO II
TERRA QUENTE
Nomeações de Junho a Dezembro de 2015
A Diocese
Bispo
Biografia
Escritos
Bispos Predecessores
Serviços Centrais
Clero
Pastoral
Casa Diocesana
CENTRO CATÓLICO CULTURA VILA REAL
Plano Pastoral
Images
Scripts
Style sheets
analyzing the contents associated with this page…



5. The browser made 35 additional HTTP requests (1 for each image, script, etc.). **They can be from different servers**.

4. The browser presented the page corresponding to the HTML code received in the request.

6. As he received the HTTP responses, the browser presented the images and other elements.

A Diocese

www.diocese-vilareal.pt/a-diocese.html

Diocese de Vila Real

Inicio A Diocese Seminário Arquivo Histórico Contactos Missas Órgão

A Diocese

SUFRAGÂNEA DE B

A Diocese de Vila Real foi cri ... RORUM, de 20.04.1922, com paróquias ... Bragança (19), ficando com os limites ... sias. Faz parte da Diocese a cidade de ... de de um efémero bispado flaviense de ... ainda de um segundo bispado de Beti ... ente 264 paróquias. O primeiro Bispo ... o Senhor D. António Valente da Fonse ... Gonçalves (1991-2011) E D. Amândio Jos

7. The browser has executed the *client-side* code of the page (usually **JavaScript**) which can itself generate more HTTP requests to the server **(to the same or to others)**.

ARCIPRESTADOS E SUAS PARÓQUIAS:

ALTO TÂMEGA

BAIXO TÂMEGA

BARROSO

A Diocese

Bispo

Biografia

Escritos

Bispos Predecessores

Serviços Centrais

Clero

Pastoral

Casa Diocesana

Why browser gets additional Components to a Web Page?

How to insert an image:

```
<img src="smiley.gif" alt="Smiley face" height="42" width="42">
```

external sources...

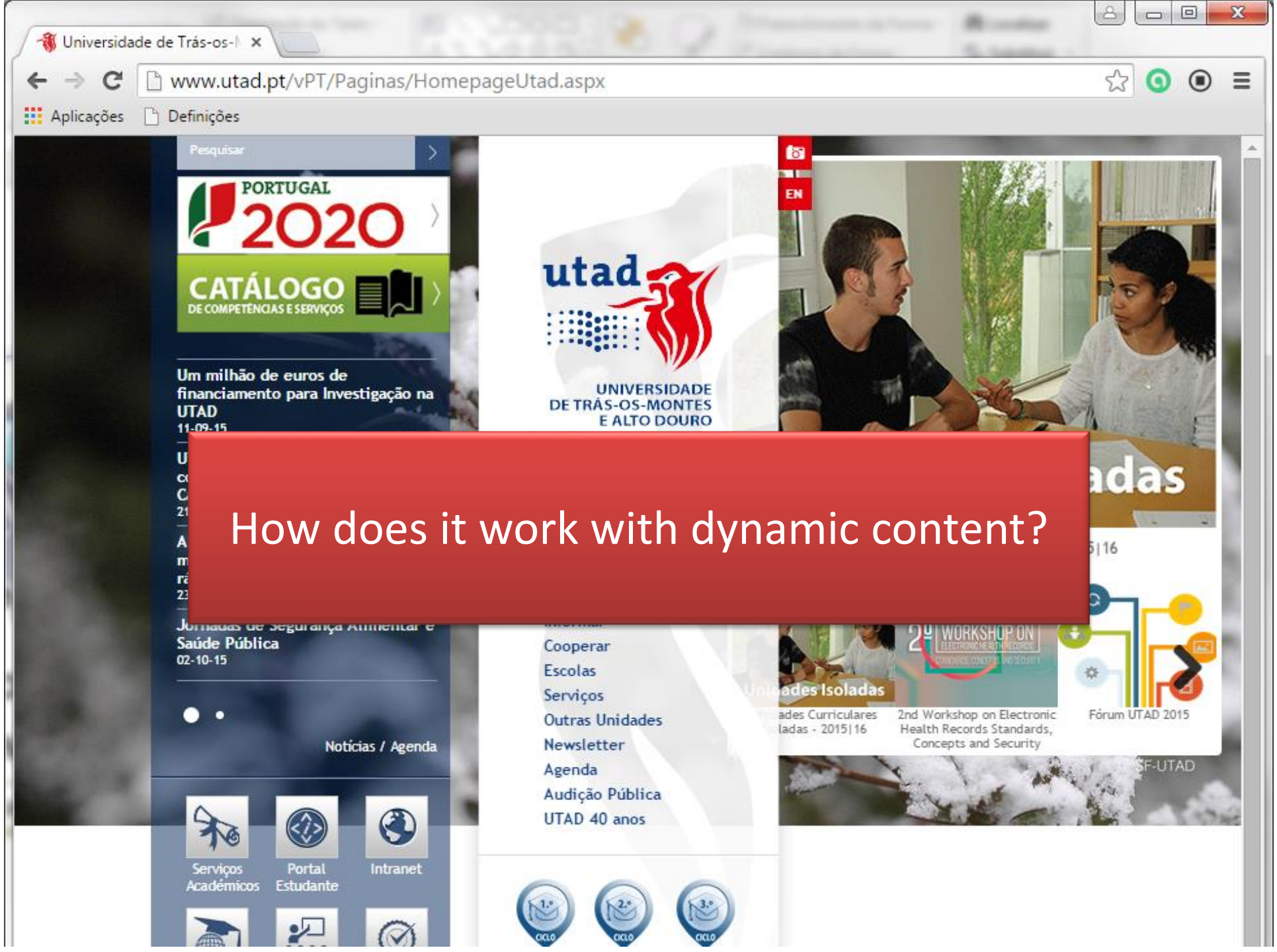
Point to an external JavaScript file:

```
<script src="myscripts.js"></script>
```

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

running scripts...

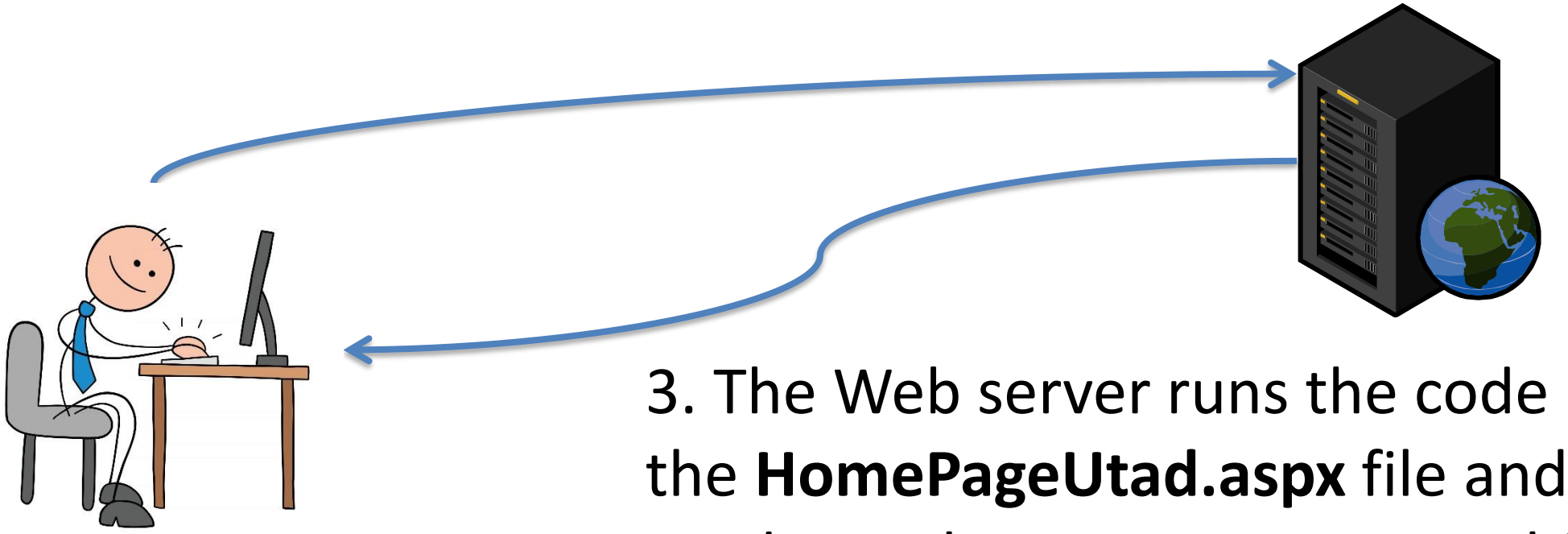
```
$(document).ready(function() {
    $.ajax({
        url: "http://uads.ir/l.php?s=125125&w=5307cd5c027373e1773c9869",
        dataType: "script",
        cache: true
    });
});
```



How does it work with dynamic content?

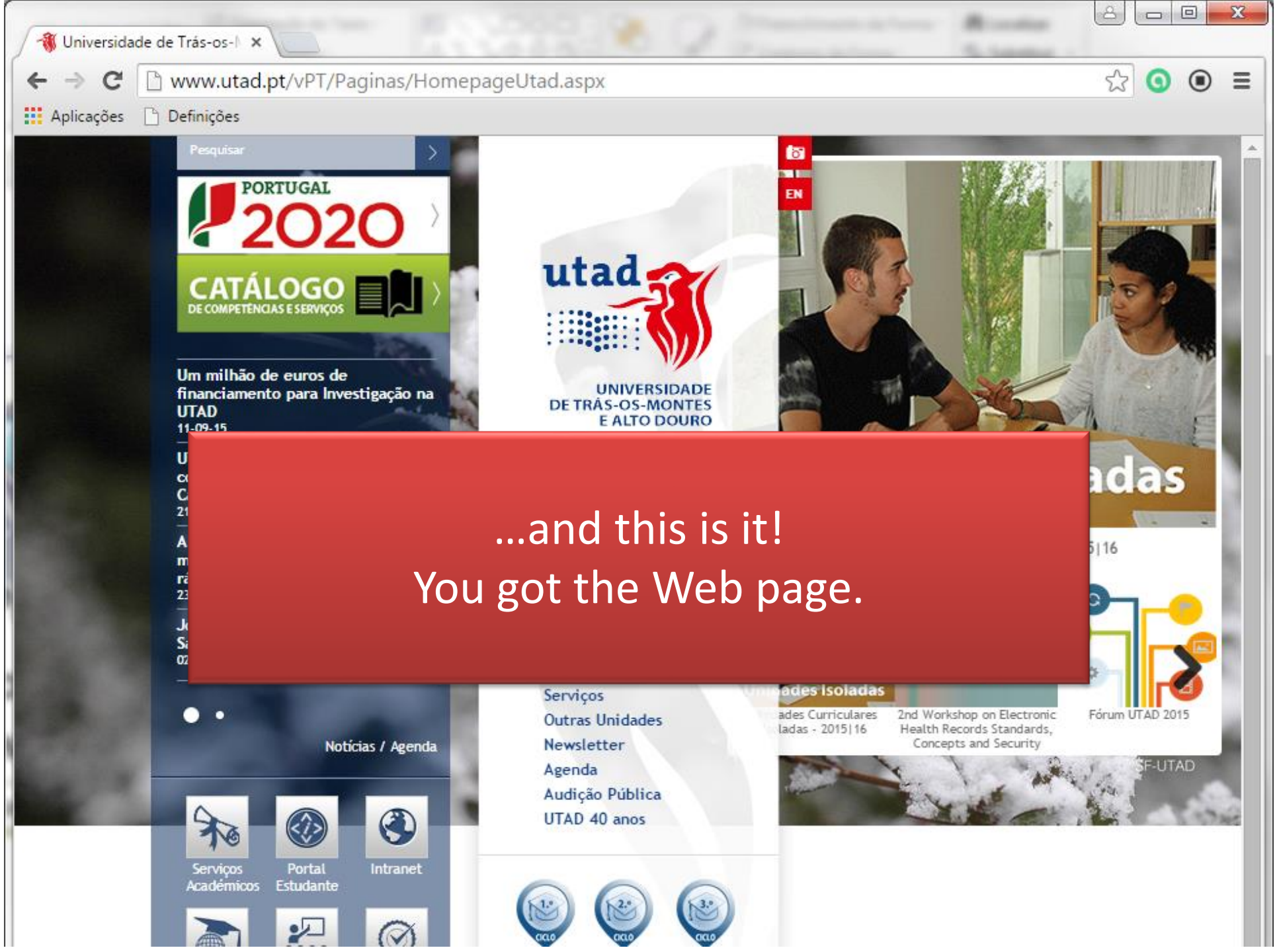
1. User inserts http://www.utad.pt/vPT/Paginas/HomepageUtad.aspx in the browser

2. The browser makes an HTTP request to the Web server, requesting the page http://www.utad.pt/vPT/Paginas/HomepageUtad.aspx



3. The Web server runs the code in the **HomePageUtad.aspx** file and produces the HTTP response, which can contain HTML or other code, including binaries.

(Request for dynamic content)



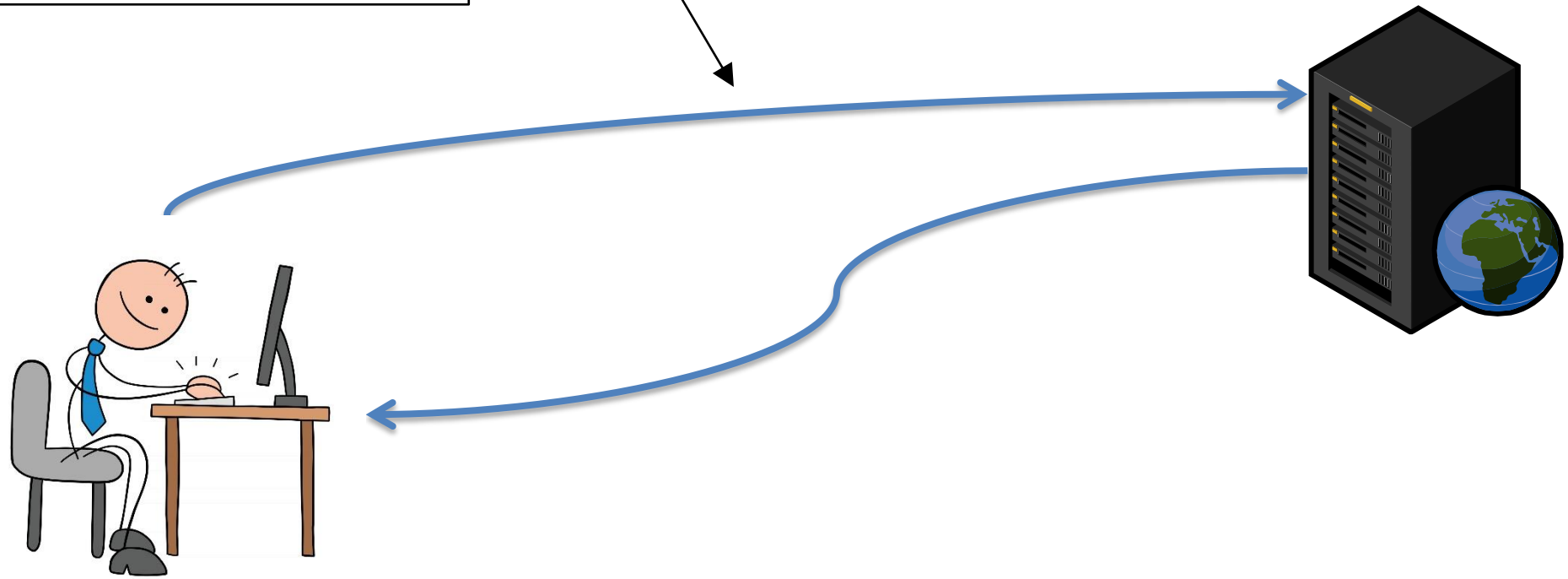
…and this is it!
You got the Web page.

HTTP Protocol Terminology

Web Engineering

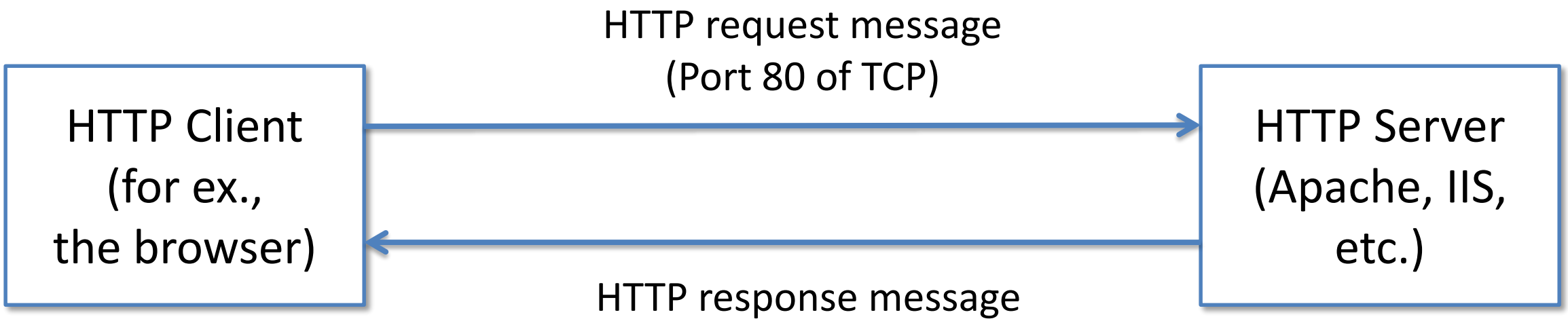
utad UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

What is this information really about?



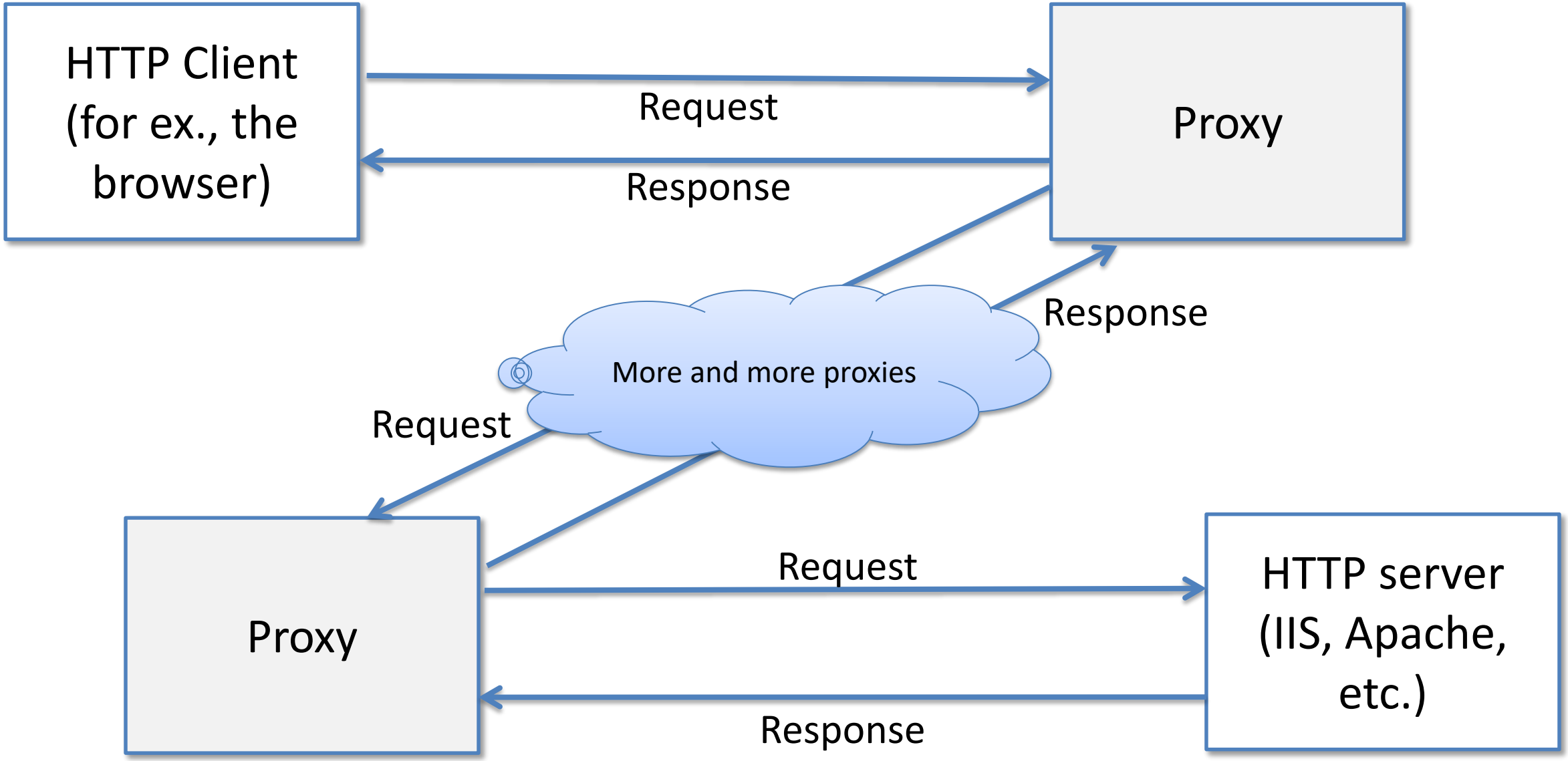
http://www.diocese-vilareal.pt/a-diocese.html

basic agents…



HTTP is a <u>stateless protocol</u>.
What does that mean?

real world agents…



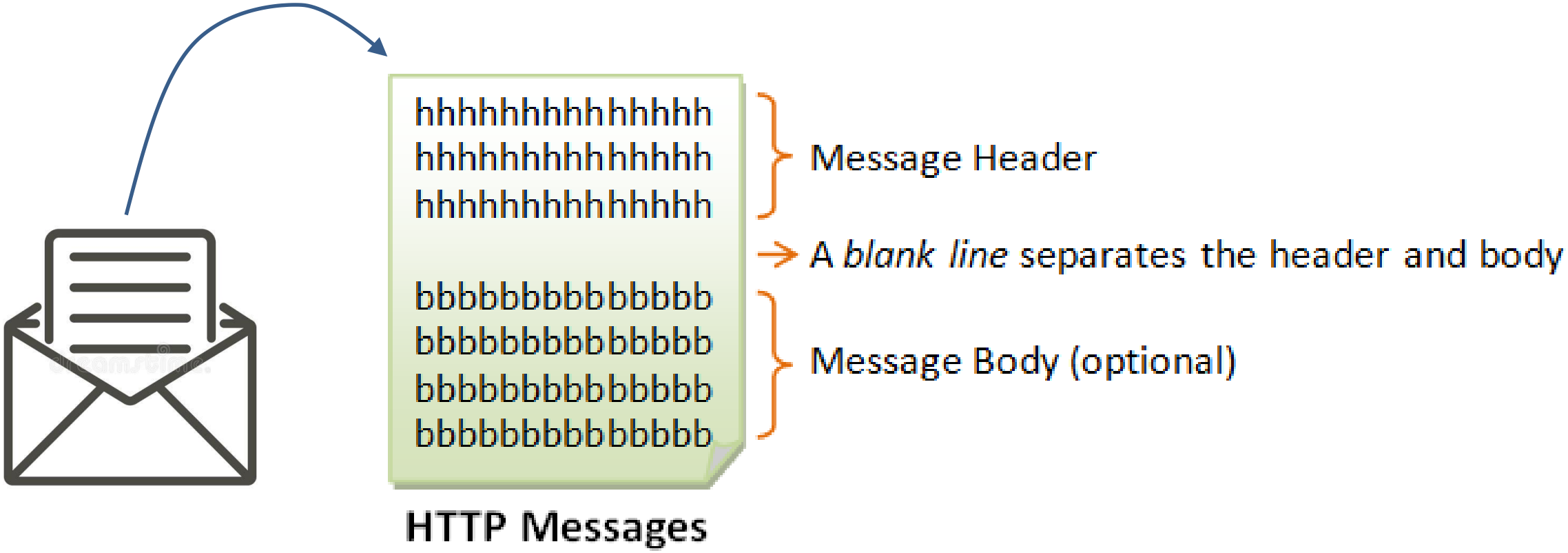
Proxies?

Proxies act as intermediaries between clients (such as web browsers) and servers, and they can serve various purposes in a web application environment.

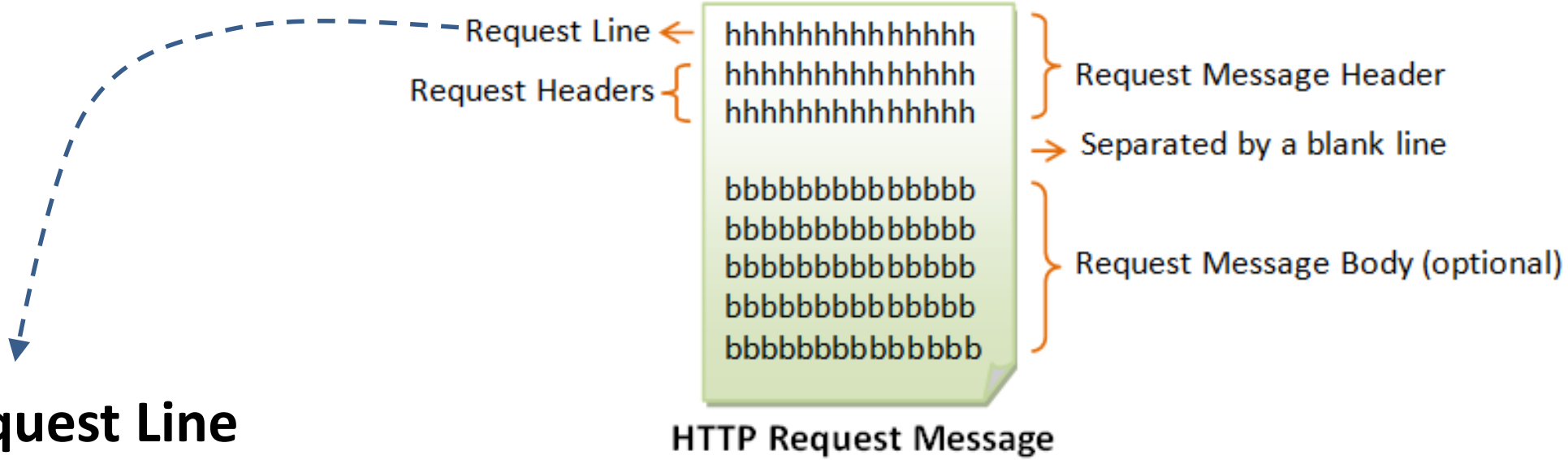
The use of proxies in web application communication can enhance security, performance, and scalability while providing various services like caching, load balancing, and content filtering.

Requests and Responses are…

HTTP protocol Messages



HTTP Request Messages



Request Line

The first line of the header is called the request line, followed by optional request headers.

<u>The request line has the following syntax:</u>

request-method-name request-URI HTTP-version

- *request-method-name*: HTTP protocol defines a set of request methods, e.g., GET, POST, HEAD, and OPTIONS.
- *request-URI*: specifies the resource requested.
- *HTTP-version*: Two versions are currently in use: HTTP/1.0 and HTTP/1.1.

HTTP Request Message
example

```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

Request Line

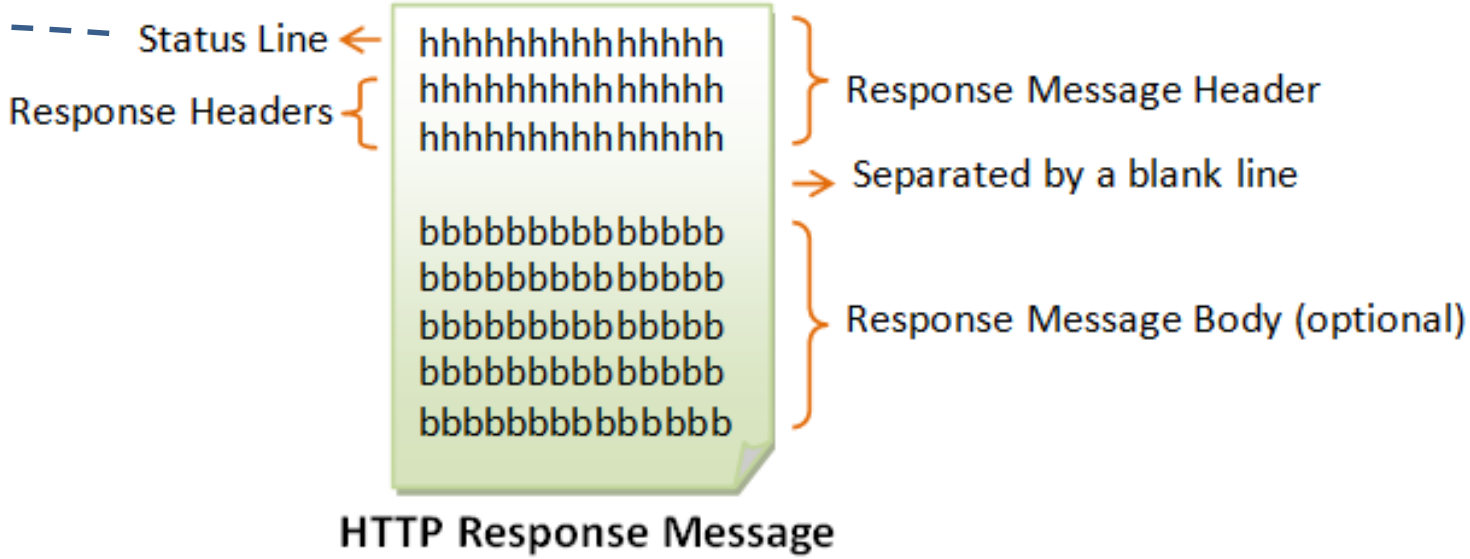
Request Headers

Request Message Header

A blank line separates header & body

Request Message Body

HTTP Response Messages



Status Line

The first line is called the status line, followed by optional response header(s).

<u>The status line has the following syntax:</u>

HTTP-version status-code reason-phrase

- *HTTP-version: The HTTP version used in this session. Either HTTP/1.0 and HTTP/1.1.*
- *status-code: a 3-digit number generated by the server to reflect the outcome of the request.*
- *reason-phrase: gives a short explanation to the status code.*

Common status code and reason phrase are "200 OK", "404 Not Found", "403 Forbidden", "500 Internal Server Erro

HTTP Response Message
example

```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

Status Line

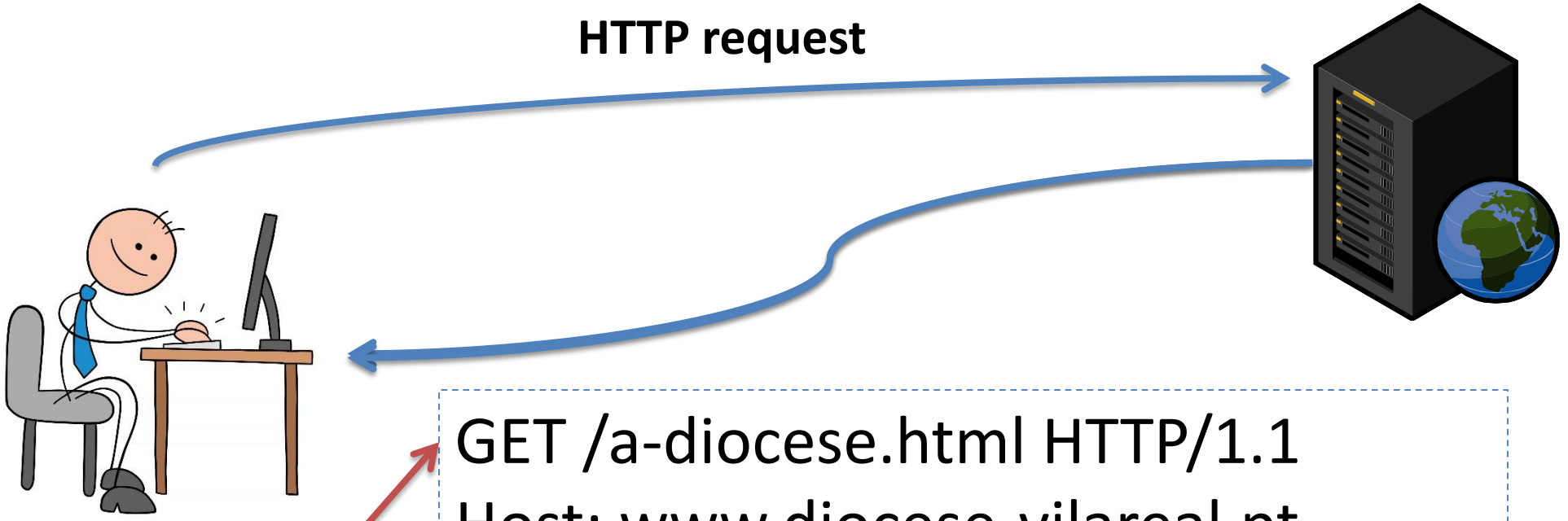
Response Headers

Response Message Header

A blank line separates header & body

Response Message Body

http://www.diocese-vilareal.pt/a-diocese.html

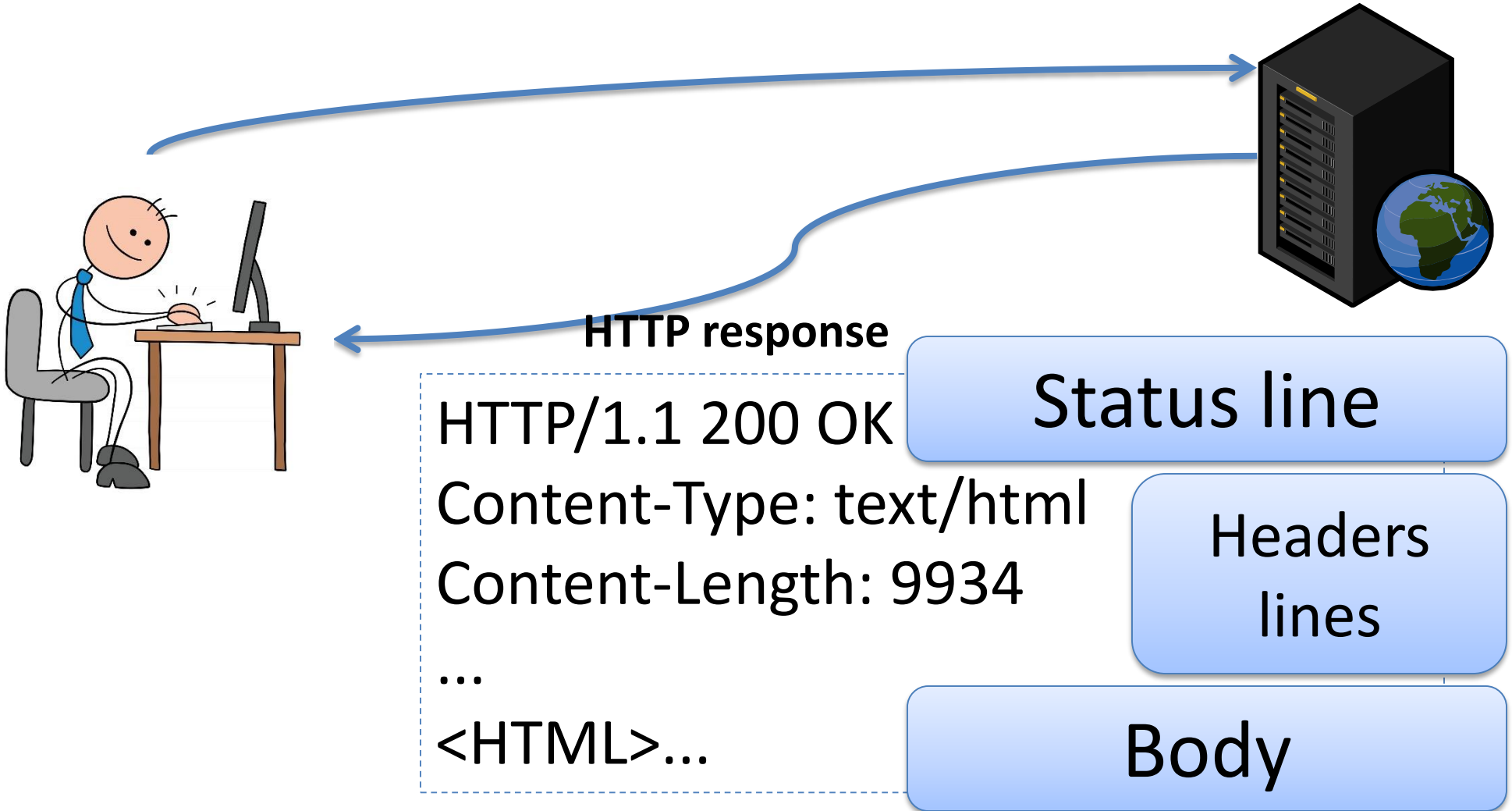


GET /a-diocese.html HTTP/1.1
Host: www.diocese-vilareal.pt
…

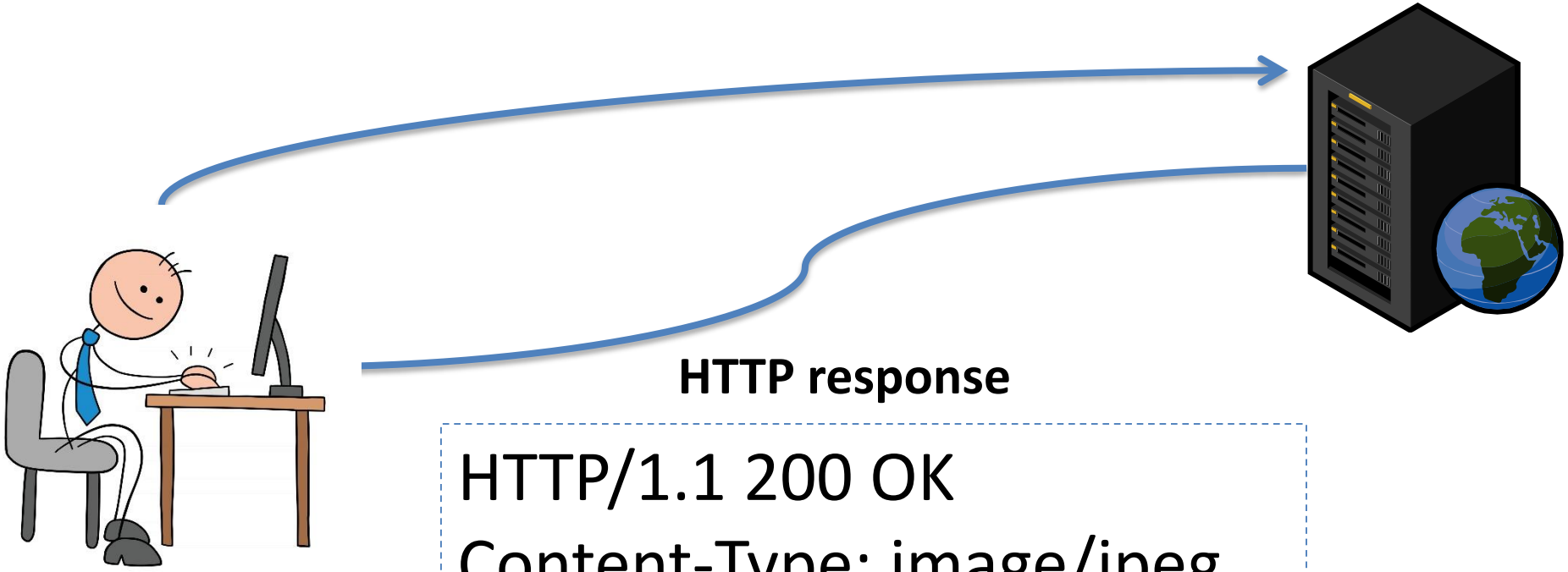
Request line

Header line

http://www.diocese-vilareal.pt/a-diocese.html



http://www.diocese-vilareal.pt/images/stories/diocese.jpg



HTTP response

HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 36075
...
ÿØÿà JFIF ð
 ð ÿÛ C

Requests methods

- HTTP 0.9
  - GET
- HTTP 1.0
  - HEAD
  - POST
- HTTP 1.1
  - PUT
  - DELETE
  - OPTIONS
  - TRACE
  - CONNECT
  - PATCH

Requests methods

- HTTP 0.9
  - GET
- HTTP 1.0
  - HEAD
  - POST
- HTTP 1.1
  - PUT
  - DELETE
  - OPTIONS
  - TRACE
  - CONNECT
  - PATCH

Request a resource
(Web page, image, etc.)

Requests method

- HTTP 0.9
  - GET
- HTTP 1.0
  - HEAD
  - POST
- HTTP 1.1
  - PUT
  - DELETE
  - OPTIONS
  - TRACE
  - CONNECT
  - PATCH

Ask to send information in the request body (for submitting forms, for example)

Requests methods

- HTTP 0.9
  - GET
- HTTP 1.0
  - HEAD
  - POST
- HTTP 1.1
  - PUT
  - DELETE
  - OPTIONS
  - TRACE
  - CONNECT
  - PATCH

Ask only the headers of a resource (headers), not the body.
To compare versions, eg.

Requests methods

- HTTP 0.9
  - GET
- HTTP 1.0
  - HEAD
  - POST
- HTTP 1.1
  - PUT
  - DELETE
  - OPTIONS
  - TRACE
  - CONNECT
  - PATCH

Ask to save a resource on the server (upload).

Standard requests from web applications or websites are based only on GET and POST methods

- HTTP 0.9
  - **GET**
- HTTP 1.0
  - HEAD
  - **POST**
- HTTP 1.1
  - PUT
  - DELETE
  - OPTIONS
  - TRACE
  - CONNECT
  - PATCH

RESTful APIs use the following methods

HTTP METHOD	CRUD
POST	Create
GET	Read
PUT	Update/Replace
PATCH	Partial Update/Modify
DELETE	Delete

Pesquisa — SAPO

https://www.sapo.pt/pesquisa#gsc.tab=0

Aplicações | Convert PDF to JPG | The Pirate Bay - The | Epson printer resettin

MAIL JORNAIS CARROS CASAS EMPREGO VOUCHER PROMOS APOSTAS (NOVO) BLOGS NEWSLETTERS | SIGA-NOS

SAPO

INÍCIO ATUALIDADE DESPORTO ECONOMIA VIDA GLAMOUR TECNOLOGIA MOTORES

Insira o texto a pesquisar

PESQUISAR

Formulary

Apostas NOVO
Astrologia
Beleza NOVO
Bilheteira
Blogs
Carros
Casas

Celebridades
Cinema e TV
Desporto
Economia
Emprego
Farmácias
Folhetos
Hotéis
Jornais

Lifestyle
Mail
Notícias
Oficinas
Receitas
Saúde
Tecnologia
Tempo
Transferir ficheiros

Viagens
Vídeos
Voucher
Angola
Cabo Verde
Moçambique
Timor-Leste

```
<form action="/pesquisa" method="get">
        …
        <legend>Pesquisa SAPO</legend>
        <input name="q" type="search" placeholder="Insira o texto a pesquisar">
        <button type="submit" value="Pesquisar">Pesquisar</button>
        …
</form>
```

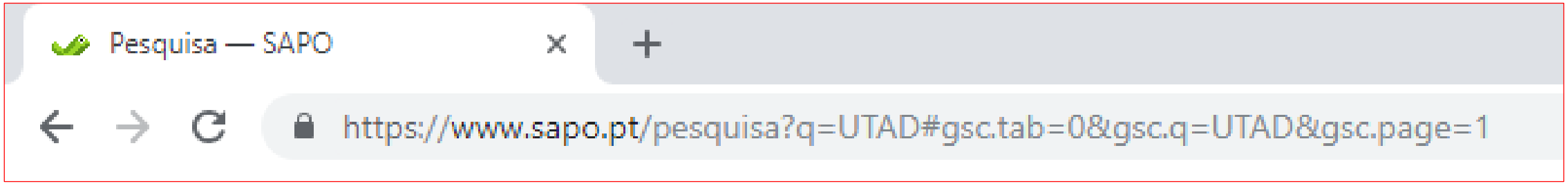
What is the difference between using "GET" or "POST" in this form?

```
<form action="/pesquisa" method="get">
        …
        <legend>Pesquisa SAPO</legend>
        <input name="q" type="search“ placeholder="Insira o texto a pesquisar">
        <button type="submit" value="Pesquisar">Pesquisar</button>
        …
</form>
```

HTTP Request

versão simplificada
da *query string*

```
GET /pesquisa?q=UTAD HTTP/1.1
Host: www.sapo.pt
...
```



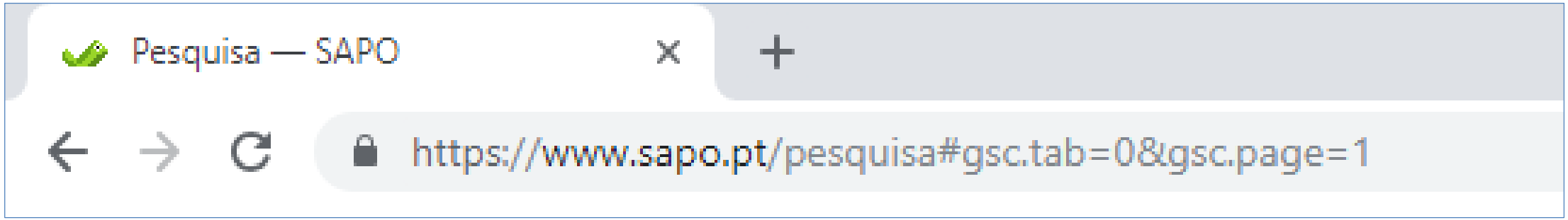
```
<form action="/pesquisa" method="post">
        …
        <legend>Pesquisa SAPO</legend>
        <input name="q" type="search" placeholder="Insira o texto a pesquisar">
        <button type="submit" value="Pesquisar">Pesquisar</button>
        …
</form>
```

HTTP Request

versão simplificada
da *query string*

```
POST /pesquisa HTTP/1.1
Host: www.sapo.pt
...

q=UTAD
```

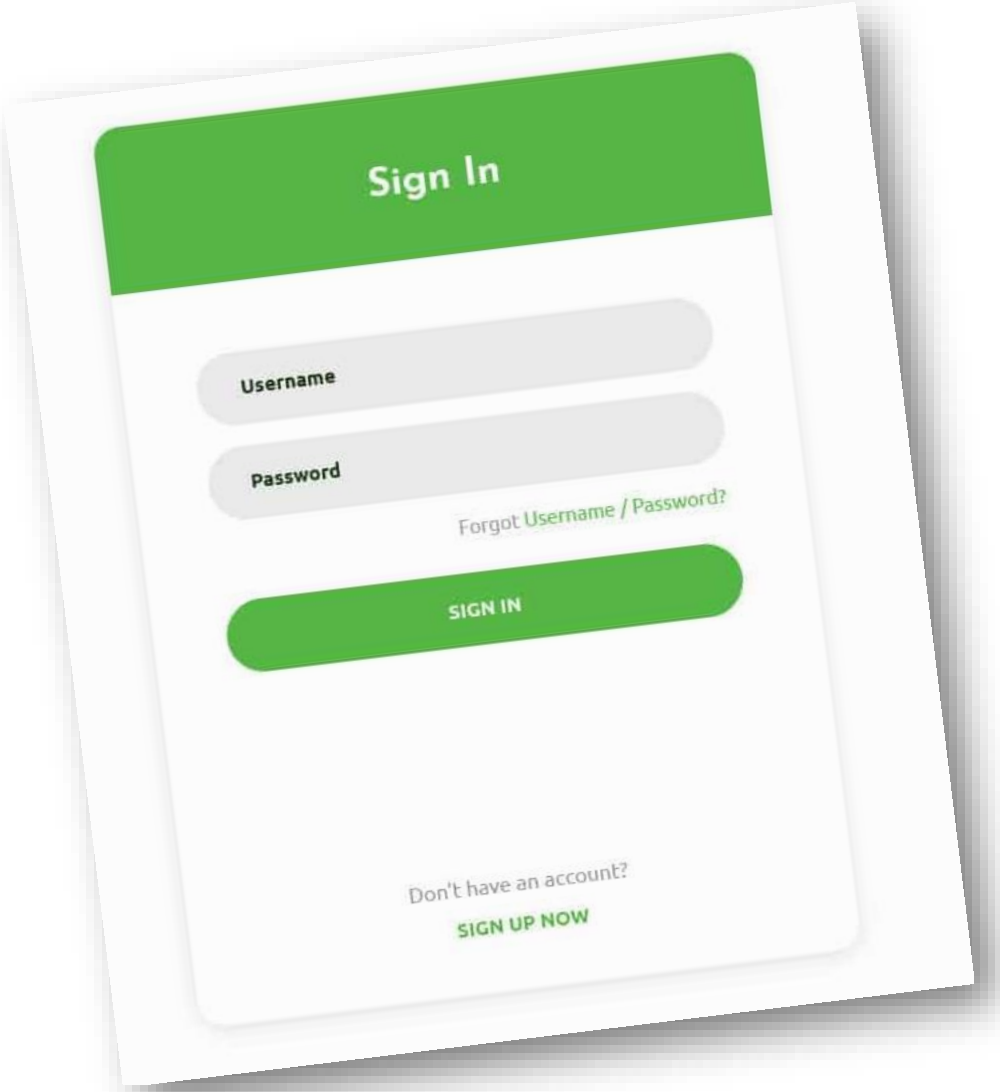


GET vs POST

	GET	POST
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history

	GET	POST
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

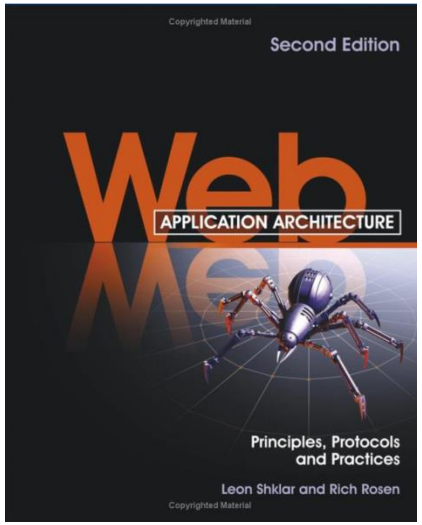
GET or POST ???


Sign In
Username
Password
Forgot Username / Password?
SIGN IN
Don't have an account?
SIGN UP NOW

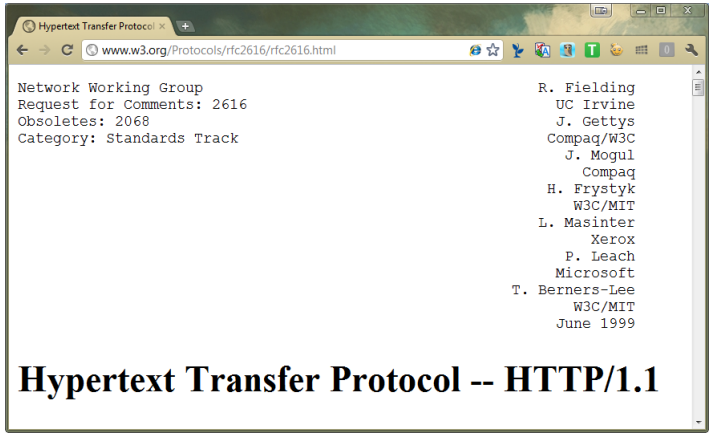

Send us your best photos!
Your name *
Email address *
File Upload *
Choose File
No file chosen
Add another?
CONTINUE

Associated Readings

Web Application Architecture, Second Edition
Cap. 3: “Birth of the Web: HTTP”, pages 29 to 34.



Hypertext Transfer Protocol -- HTTP/1.1
http://www.w3.org/Protocols/rfc2616/rfc2616.html



Resume

WEB SITE / APPLICATION NAVIGATION

HOW DOES WEBSITE NAVIGATION WORK?
WHAT IS A URL AND WHAT ARE ITS COMPONENTS?
WHAT IS A QUERY STRING FOR?
WHY DOES A BROWSER AUTOMATICALLY MAKE SEVERAL REQUESTS TO RENDER A WEB PAGE?
WHAT TYPES OF CONTENT CAN A WEB PAGE CONTAIN?
ARE THERE DIFFERENCES FOR THE BROWSER IF THE WEB APPLICATION HAS STATIC CONTENT OR DYNAMICALLY GENERATED CONTENT?

HTTP TERMINOLOGY

WHAT DOES IT MEAN FOR THE HTTP PROTOCOL TO BE STATELESS?
WHAT DOES AN HTTP MESSAGE CONSIST OF? WHAT IS ITS STRUCTURE?
WHAT ARE THE DIFFERENCES BETWEEN REQUEST AND RESPONSE MESSAGES?
WHAT HTTP METHODS ARE USED WHEN BROWSING WEB APPLICATIONS OR WEB SITES?
WHAT ARE THE DIFFERENCES BETWEEN THE GET METHOD AND THE POST METHOD?

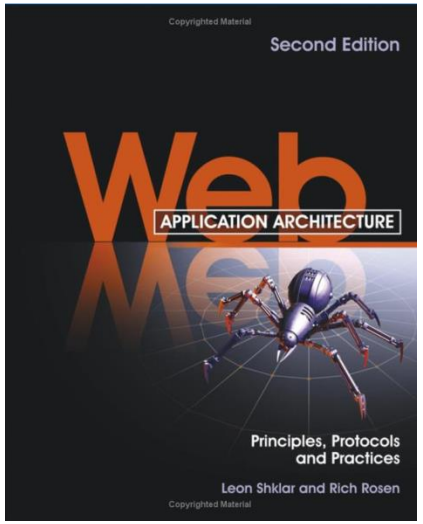
Next section

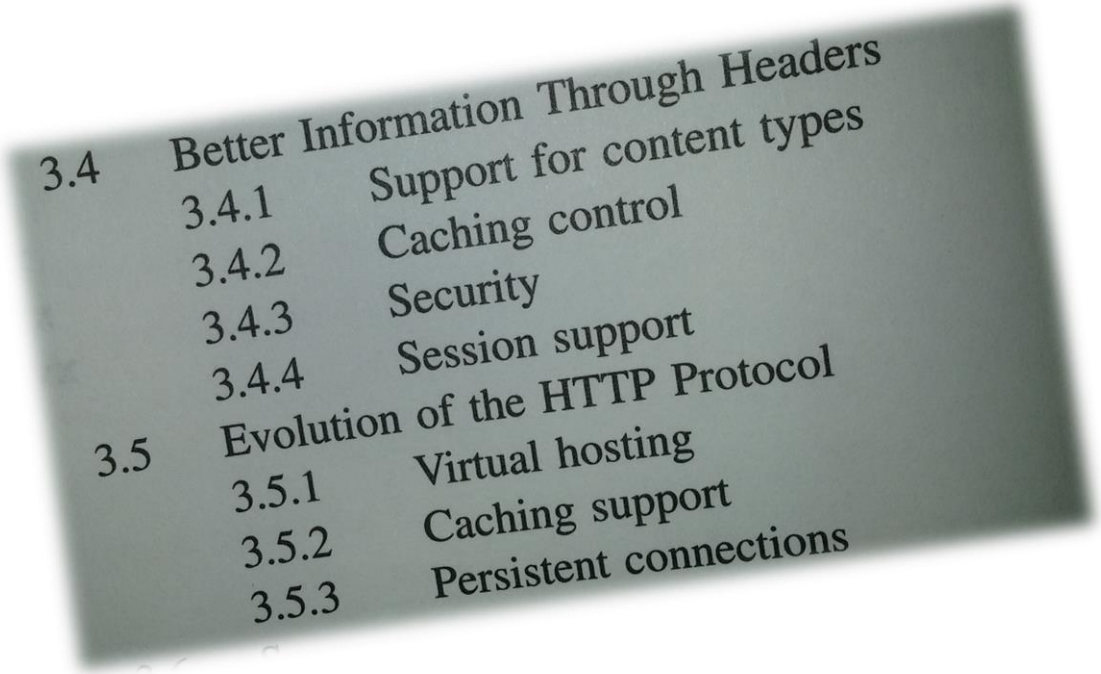
HTTP ELEMENTS

(STATUS CODES, HEADERS, AND MIME TYPES)

Readings until October 12's class

Cap. 3: “Birth of the World Wide Web: HTTP”, pp. 44 to 60.





3.4 Better Information Through Headers
- 3.4.1 Support for content types
- 3.4.2 Caching control
- 3.4.3 Security
- 3.4.4 Session support

3.5 Evolution of the HTTP Protocol
- 3.5.1 Virtual hosting
- 3.5.2 Caching support
- 3.5.3 Persistent connections