



EBook Gratis

APRENDIZAJE http-headers

Free unaffiliated eBook created from
Stack Overflow contributors.

#http-headers

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con los encabezados http.....	2
Observaciones.....	2
Examples.....	2
Solicitud HTTP.....	2
Respuesta HTTP.....	2
Capítulo 2: Aceptar petición).....	3
Introducción.....	3
Sintaxis.....	3
Parámetros.....	3
Observaciones.....	3
Examples.....	3
Tipo de HTML solamente.....	3
Coincidir con todos los tipos de texto.....	4
text / html y application / xml con una preferencia text / html.....	4
Preferencia por un tipo sobre otro.....	4
Capítulo 3: Aceptar-Charset: (Solicitud).....	6
Introducción.....	6
Sintaxis.....	6
Parámetros.....	6
Observaciones.....	6
Examples.....	6
Solo acepta UTF-8.....	6
Solo acepta UTF-8 e iso-8859-1.....	7
Solo se acepta UTF-8, iso-8859-1 con preferencia.....	7
Acepta cualquier juego de caracteres pero tiene preferencia por algunos tipos.....	7
Capítulo 4: Aceptar-Codificación: (Solicitud).....	9
Introducción.....	9
Sintaxis.....	9
Parámetros.....	9

Observaciones.....	9
Examples.....	9
Solicitar gzip.....	10
Solicitar gzip y desinflar.....	10
Solicite compres pero prefiero gzip.....	10
Sin preferencia por el tipo de codificación.....	10
Capítulo 5: Aceptar-Idioma: (Solicitud).....	12
Introducción.....	12
Sintaxis.....	12
Parámetros.....	12
Observaciones.....	12
Examples.....	12
solo inglés.....	12
Inglés americano o inglés básico.....	13
Inglés americano o inglés básico.....	13
Coincide con cualquier idioma.....	13
Capítulo 6: Rangos de aceptación: (Respuesta).....	15
Introducción.....	15
Sintaxis.....	15
Parámetros.....	15
Observaciones.....	15
Examples.....	16
Servidor soporta rangos.....	16
Solicitud: " http://example.com ".....	16
Respuesta:.....	16
El servidor no admite rangos.....	16
Solicitud: " http://example.com ".....	16
Respuesta:.....	16
Capítulo 7: X-Request-ID.....	18
Introducción.....	18
Sintaxis.....	18
Observaciones.....	18

Examples.....	18
nginx.....	18
Heroku.....	19
Django.....	19
ID de solicitud (solicitud / respuesta).....	19
Creditos.....	21

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [http-headers](#)

It is an unofficial and free http-headers ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official http-headers.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con los encabezados http

Observaciones

Los encabezados HTTP son una parte importante de la comunicación HTTP. Cada solicitud HTTP y respuesta HTTP generalmente contienen múltiples encabezados. Los intermediarios, como los apoderados, a menudo interpretan algunos de los encabezados y transmiten o filtran otros.

Examples

Solicitud HTTP

Una solicitud HTTP simple para el recurso `/index.html` . El host `www.example.com` se especifica en el encabezado HTTP `Host` .

```
GET /index.html HTTP/1.1
Host: www.example.com
```

Respuesta HTTP

Una posible respuesta a la solicitud anterior. La respuesta contiene los encabezados HTTP `Date` , `Content-Type` `Content-Encoding` `Content-Length` .

```
HTTP/1.1 200 OK
Date: Wed, 21 Jun 2017 10:58:03 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 150
```

```
<response body>
```

El cuerpo de la respuesta se envía después de los encabezados, separados por una línea en blanco.

Lea [Empezando con los encabezados http en línea](https://riptutorial.com/es/http-headers/topic/10573/empezando-con-los-encabezados-http): <https://riptutorial.com/es/http-headers/topic/10573/empezando-con-los-encabezados-http>

Capítulo 2: Aceptar petición)

Introducción

¿Qué tipo de `Content-Type` acepta el cliente?

Sintaxis

- Aceptar: `MIMETYPE / MIMESubtype; QualityFactor`
- Aceptar: `MIMETYPE / MIMESubtype; QualityFactor, MIMETYPE / MIMESubtype; QualityFactor, ...`

Parámetros

Parámetro	Descripción
Tipo MIME	La primera mitad del tipo mime. Esto también puede ser un <code>*/*</code> para todos los tipos
MIMESubtype	La segunda mitad del tipo mime o un <code>*</code> para todos los subtipos (es decir, <code>image/*</code>)
Factor de calidad	El factor de calidad en el formato <code>;q=0.8</code> (opcional)

Observaciones

Los tipos de contenido son tipos MIME (es decir, `text/html`) separados por comas con un factor de calidad opcional (usando a `;q=`) que se usa en la preferencia de los clientes para usar este tipo. El factor de calidad tiene un valor de 0 a 1, ya que cuanto mayor sea el número, mayor será la preferencia por ese tipo.

Si el servidor no puede encontrar un tipo aceptable para responder, debe enviar una respuesta 406 (no aceptable).

Examples

Tipo de HTML solamente

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
```

```
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará Contenido-Tipos de texto / html

Coincidir con todos los tipos de texto

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará tipos de contenido de cualquiera de los tipos de texto / * de tipos MIME. Por ejemplo, texto / html, texto / plano, texto / css.

text / html y application / xml con una preferencia text / html

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html;q=1.0,application/xml;q=0.9
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará Contenido-Tipos de texto / html y application / xml pero prefiere texto / html

Preferencia por un tipo sobre otro

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará todos los tipos de tipos de contenido, pero prefiere text / html y application / xml.

Lea Aceptar petición) en línea: <https://riptutorial.com/es/http-headers/topic/10614/aceptar-peticion->

Capítulo 3: Aceptar-Charset: (Solicitud)

Introducción

`Accept-Charset` le dice al servidor qué juegos de caracteres acepta el cliente.

Sintaxis

- Aceptar-Charset: tipo; QualityFactor
- Aceptar-Charset: tipo; QualityFactor, tipo; QualityFactor, tipo; QualityFactor, ...

Parámetros

Parámetro	Descripción
tipo	Un nombre de conjunto de caracteres. Esto también puede ser un * para todos los juegos de caracteres.
Factor de calidad	El factor de calidad en el formato ;q=0.8 (opcional)

Observaciones

`Accept-Charset` toma una serie de juegos de caracteres e incluye una preferencia opcional por la que debe usar el servidor. El conjunto de caracteres es uno de la lista de conjuntos de caracteres disponibles en el registro de "Conjuntos de caracteres" de la IANA. Por ejemplo `UTF-8`.

El conjunto de caracteres está separado por comas con un factor de calidad opcional (usando a ;q=) que se usa la preferencia del cliente para usar este tipo. El factor de calidad tiene un valor de 0 a 1, ya que cuanto mayor sea el número, mayor será la preferencia por ese tipo.

Si este encabezado no está incluido, el cliente aceptará cualquier conjunto de caracteres.

El servidor utiliza `Content-Type` para informar al cliente qué conjunto de caracteres está utilizando.

Si el servidor no puede encontrar un conjunto de caracteres aceptable para responder, debe enviar una respuesta 406 (no aceptable) o ignorar este encabezado y no realizar ninguna negociación de contenido.

Examples

Solo acepta UTF-8

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xml;q=0.9,*/*;q=0.8
Accept-Charset: UTF-8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará solo conjuntos de caracteres UTF-8.

Solo acepta UTF-8 e iso-8859-1

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xml;q=0.9,*/*;q=0.8
Accept-Charset: UTF-8, iso-8859-1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará solo los conjuntos de caracteres UTF-8 e iso-8859-1.

Solo se acepta UTF-8, iso-8859-1 con preferencia.

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xml;q=0.9,*/*;q=0.8
Accept-Charset: UTF-8, iso-8859-1;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará los conjuntos de caracteres UTF-8 e iso-8859-1, pero prefiere UTF-8 (que tiene un factor de calidad de 1.0).

Acepta cualquier juego de caracteres pero tiene preferencia por algunos tipos.

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xml;q=0.9,*/*;q=0.8
Accept-Charset: UTF-8, iso-8859-1;q=0.8, *;q=0.5
Accept-Language: en-US,en;q=0.5
```

```
Accept-Encoding: gzip, deflate  
Connection: keep-alive
```

El cliente aceptará cualquier conjunto de caracteres, pero prefiere UTF-8 y luego iso-8859-1 si UTF-8 no está disponible.

Lea **Aceptar-Charset: (Solicitud)** en línea: <https://riptutorial.com/es/http-headers/topic/10613/aceptar-charset---solicitud->

Capítulo 4: Aceptar-Codificación: (Solicitud)

Introducción

`Accept-Encoding` le dice al servidor qué codificación acepta el cliente. La codificación se utiliza principalmente para la compresión.

Sintaxis

- Aceptar-Codificación: Codificación; QualityFactor
- Aceptar-Codificación: Codificación; QualityFactor, tipo; QualityFactor, tipo; QualityFactor, ...

Parámetros

Parámetro	Descripción
Codificación	El tipo de codificación a utilizar. Esto también puede ser un * para decir que el cliente no tiene preferencia a qué codificación usar
Factor de calidad	El factor de calidad en el formato <code>;q=0.8</code> . Si se establece en 0, significa "no aceptable". (Opcional)

Observaciones

`Accept-Encoding` toma una cantidad de codificación e incluye una preferencia opcional por la que el servidor debería usar. La codificación es una de la lista de codificaciones disponibles en el registro de IANA. Por ejemplo `gzip`.

La codificación está separada por comas con un factor de calidad opcional (usando a `;q=`) que se utiliza la preferencia de los clientes para usar esta codificación. El factor de calidad tiene un valor de 0 a 1; cuanto mayor sea el número, mayor será la preferencia por esa codificación.

Si este encabezado no está incluido, el cliente no establece ninguna preferencia por la codificación. No significa que el cliente admita todas las codificaciones.

Un valor de `identity` siempre es aceptable a menos que lo rechace con `identity;q=0`.

El servidor utiliza `Content-Encoding` para informar al cliente qué codificación está utilizando.

Si el servidor no puede encontrar un conjunto de caracteres aceptable para responder, debe enviar una respuesta 406 (no aceptable) o ignorar este encabezado y no realizar ninguna negociación de contenido.

Examples

Solicitar gzip

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará gzip y codificación de identidad.

Solicitar gzip y desinflar.

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: compress, gzip
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente solo aceptará codificaciones gzip, compress y de identidad.

Solicite compres pero prefiero gzip

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip;q=1.0, compress;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará gzip, comprimir y codificación de identidad, pero prefiere gzip (que tiene un factor de calidad de 1.0).

Sin preferencia por el tipo de codificación.

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
```

```
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: *
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente no tiene preferencia por el tipo de codificación.

Lea **Aceptar-Codificación: (Solicitud)** en línea: <https://riptutorial.com/es/http-headers/topic/10615/acceptar-codificacion---solicitud->

Capítulo 5: Aceptar-Idioma: (Solicitud)

Introducción

`Accept-Language` le dice al servidor qué idioma (como el inglés) acepta el cliente.

Sintaxis

- Aceptar Idioma: Idioma; Calidad Factor
- Aceptar idioma: idioma; factor de calidad, idioma; factor de calidad, ...
- Aceptar-Idioma: *

Parámetros

Parámetro	Descripción
Idioma	Que idioma es aceptable.
Factor de calidad	El factor de calidad en el formato <code>;q=0.8</code> (opcional)
*	Coincide con cualquier idioma

Observaciones

`Accept-Language` toma varios idiomas e incluye una preferencia opcional para el servidor. El idioma es uno de la lista de disponibles en la página Registro de subetiquetas de idioma de la IANA. Por ejemplo, `en` es inglés, y `en-US` es inglés en EE.UU.

El idioma está separado por comas con un factor de calidad opcional (usando a `;q=`) que se usa la preferencia del cliente para usar este idioma. El factor de calidad tiene un valor de 0 a 1, ya que cuanto mayor sea el número, más preferencia tendrá ese idioma.

Si este encabezado no está incluido, el cliente aceptará cualquier idioma.

El servidor utiliza `Content-Language` para informar al cliente qué idioma está utilizando.

Examples

solo inglés

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
```



```
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html
Accept-Language: en
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente solo aceptará contenido-idioma del inglés.

Inglés americano o inglés básico

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html
Accept-Language: en-US, en; q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará el inglés pero prefiere el inglés estadounidense.

Inglés americano o inglés básico

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html
Accept-Language: da, en-gb; q=0.8, en; q=0.7
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente prefiere el danés, pero también aceptará el inglés británico, o si no está disponible el inglés básico.

Coincide con cualquier idioma

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html
Accept-Language: *
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

El cliente aceptará cualquier idioma.

Lea Aceptar-Idioma: (Solicitud) en línea: <https://riptutorial.com/es/http->

Capítulo 6: Rangos de aceptación: (Respuesta)

Introducción

`Accept-Ranges` le dice al cliente que este servidor admite rangos para este recurso (archivo).

Sintaxis

- Rangos de aceptación: `RangeType`
- Rangos de aceptación: ninguno

Parámetros

Parámetro	Descripción
<code>RangeType</code>	Ese tipo de rangos son compatibles. Esto es actualmente solo <code>bytes</code> o <code>none</code> .
ninguna	El servidor no soporta rangos en este recurso

Observaciones

`Accept-Ranges` es parte del sistema de rangos. El sistema de rangos permite al cliente solicitar solo parte de un archivo en lugar de tener que descargar todo el archivo.

Por ejemplo, si un cliente solo necesita los últimos 100 bytes de un archivo 10M, puede solicitar que el servidor solo envíe datos desde la compensación 10485660 a 10485760.

`Accept-Ranges` se envía desde el servidor para indicar al cliente si admite rangos. Esto solo se aplica a este recurso en particular (archivo), otros archivos pueden aceptar diferentes tipos de rango.

Sólo dos valores están definidos actualmente, `bytes` y `none`. Los `bytes` valores significa que puede solicitar rangos de bytes (el desplazamiento y el final estarán en bytes). Un valor de 'ninguno' significa que el servidor no admite rangos.

Los clientes tienen la libertad de solicitar solicitudes de rango de bytes sin verificar si el servidor admite rangos.

El cliente usa `Range` para solicitar un rango del servidor y el servidor responde con un estado de 206 (Contenido parcial) si está enviando el rango de bytes o 200 (ok) si va a enviar todo el archivo.

Examples

Servidor soporta rangos

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Respuesta:

```
HTTP/1.1 200 OK
Date: Sat, 01 Jan 2000 01:00:00 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1h PHP/5.4.31
Keep-Alive: timeout=5, max=97
Connection: Keep-Alive
Content-Type: text/html
Accept-Ranges: bytes
Content-Length: 500
```

El servidor no admite rangos

Solicitud: " <http://example.com> "

```
GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Respuesta:

```
HTTP/1.1 200 OK
Date: Sat, 01 Jan 2000 01:00:00 GMT
Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1h PHP/5.4.31
Keep-Alive: timeout=5, max=97
Connection: Keep-Alive
Content-Type: text/html
Accept-Ranges: none
Content-Length: 500
```

Lea Rangos de aceptación: (Respuesta) en línea: <https://riptutorial.com/es/http->

Capítulo 7: X-Request-ID

Introducción

El encabezado `X-Request-ID` se puede usar para rastrear solicitudes individuales a un servicio web (como una API REST) desde el cliente al servidor y sus backends.

Sintaxis

- X-Request-ID: <valor>

Observaciones

Un cliente puede enviar un encabezado HTTP `X-Request-ID: some-value`. El servidor debe utilizar el valor provisto y proporcionarlo en cualquier solicitud que realice a los servicios de back-end con el fin de atender la solicitud inicial. Al enviar la respuesta, el servidor devolverá el mismo encabezado al cliente. Para fines de seguimiento, el servidor incluirá el valor en sus registros, para permitir la correlación de solicitudes y respuestas con los registros correspondientes.

Examples

nginx

Los proxies inversos pueden detectar si un cliente proporciona un encabezado de X-Request-ID, y pasarlo al servidor backend. Si no se proporciona dicho encabezado, puede proporcionar un valor aleatorio.

```
map $http_x_request_id $reqid {
    default    $http_x_request_id;
    ""        $request_id;
}
```

El código anterior almacena el Id. De solicitud en la variable `$reqid` desde donde se puede usar posteriormente en los registros.

```
log_format trace '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" "$http_user_agent" '
                '"$http_x_forwarded_for" $reqid';
```

También debe pasarse a los servicios de backend.

```
location @proxy_to_app {
    proxy_set_header X-Request-ID $reqid;
    proxy_pass http://backend;
    access_log /var/log/nginx/access_trace.log trace;
}
```

Heroku

Heroku siempre pasará un encabezado `X-Request-ID` enviado por el cliente, o generará el suyo propio.

Consulte la documentación en [ID de solicitud HTTP](#) .

Django

Cuando se usa Django como marco de servicios web, el paquete `django-log-request-id` se puede usar para analizar y registrar las ID de solicitud.

Ajustes

```
MIDDLEWARE_CLASSES = (
    'log_request_id.middleware.RequestIDMiddleware',
    # ... other middleware goes here
)

LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'filters': {
        'request_id': {
            '()': 'log_request_id.filters.RequestIDFilter'
        }
    },
    'formatters': {
        'standard': {
            'format': '%(levelname)-8s [%(asctime)s] [%(request_id)s] %(name)s: %(message)s'
        },
    },
    'handlers': {
        'console': {
            'level': 'DEBUG',
            'class': 'logging.StreamHandler',
            'filters': ['request_id'],
            'formatter': 'standard',
        },
    },
    'loggers': {
        'myapp': {
            'handlers': ['console'],
            'level': 'DEBUG',
            'propagate': False,
        },
    },
}
```

ID de solicitud (solicitud / respuesta)

El mismo encabezado de `X-Request-ID` puede ser enviado por un cliente en una solicitud, o por un servidor en una respuesta.

```
X-Request-ID: f9ed4675f1c53513c61a3b3b4e25b4c0
```

El valor no tiene ningún significado inherente, pero es solo un token para identificar solicitudes y respuestas correlacionadas.

Lea X-Request-ID en línea: <https://riptutorial.com/es/http-headers/topic/10581/x-request-id>

Creditos

S. No	Capítulos	Contributors
1	Empezando con los encabezados http	Community , Stefan Kögl
2	Aceptar petición)	Paul Hutchinson
3	Aceptar-Charset: (Solicitud)	Paul Hutchinson
4	Aceptar-Codificación: (Solicitud)	Paul Hutchinson
5	Aceptar-Idioma: (Solicitud)	Paul Hutchinson
6	Rangos de aceptación: (Respuesta)	Paul Hutchinson
7	X-Request-ID	Stefan Kögl