

RESENDING METHOD AND DEVICE FOR HYPERTEXT TRANSFER PROTOCOL REQUEST, AND CLIENT TERMINAL

The present application claims priority to Chinese Patent Application No. 201510335208.3
5 filed on June 16, 2015 and entitled " RESENDING METHOD AND DEVICE FOR HYPERTEXT
TRANSFER PROTOCOL REQUEST, AND CLIENT TERMINAL ", which is incorporated herein
by reference in its entirety.

Technical Field

The present application relates to the technical field of browsers and similar containers, and in
10 particular, to a resending method and device for a Hypertext Transfer Protocol request and a client
terminal.

Background Art

A Hypertext transfer protocol (HTTP) request is a way of recording monitoring data of a client
terminal to a server terminal. When a user performs an operation on a current web page on a client
15 terminal, for example, clicking a link on the web page or logging in to an account on the web page,
the client terminal will monitor the operation and send monitoring data to a server through an
HTTP request.

In an actual application, when web page redirection occurs on the client terminal, for example,
redirection from a first web page to a second web page, the client terminal will offload the content
20 of the first web page and load the content of the second web page. During the process that the client
terminal offloads the first web page, if an HTTP request sent in a first web page state fails to be
sent or has not yet been completely sent, the HTTP request may quite possibly be cancelled, and
thus the server terminal cannot receive monitoring data carried in the HTTP request, thereby
resulting in data loss.

25 Summary of the Invention

The present application provides a resending method and device for a Hypertext Transfer Protocol request and a client terminal, such that an HTTP request that fails to be sent due to web page redirection can be resent, thereby solving the problem in the prior art.

In a first aspect of an embodiment of the present application, a resending method for a
5 Hypertext Transfer Protocol request is provided, which includes:

sending, by a client terminal, a Hypertext Transfer Protocol HTTP request to a server, and recording information of the HTTP request;

if it is determined that an HTTP request that fails to be sent exists in a first web page state when redirection from the first web page to a second web page needs to occur, acquiring information of
10 the HTTP request that fails to be sent;

transferring, to the second web page, the information of the HTTP request that fails to be sent;
and

after redirection to the second web page, resending the HTTP request that fails to be sent according to the information of the HTTP request that fails to be sent.

15 In a second aspect of an embodiment of the present application, a resending device for a Hypertext Transfer Protocol request is provided, which includes:

a first sending unit configured to send a Hypertext Transfer Protocol HTTP request to a server;

a first recording unit configured to record information of the HTTP request;

an acquisition unit configured to, if it is determined that an HTTP request that fails to be sent
20 exists in a first web page state when redirection from the first web page to a second web page needs to occur, acquire information of the HTTP request that fails to be sent;

a transferring unit configured to transfer, to the second web page, the information of the HTTP request that fails to be sent; and

a second sending unit configured to, after redirection to the second web page, resend the HTTP
25 request that fails to be sent according to the information of the HTTP request that fails to be sent.

In a third aspect of an embodiment of the present application, a client terminal is provided, which includes:

a processor; and a memory configured to store executable instructions of the processor,
where the processor is configured to:

send a Hypertext Transfer Protocol HTTP request to a server, and record information of the HTTP request;

5 if it is determined that an HTTP request that fails to be sent exists in a first web page state when redirection from the first web page to a second web page needs to occur, acquire information of the HTTP request that fails to be sent;

transfer, to the second web page, the information of the HTTP request that fails to be sent; and
after redirection to the second web page, resend the HTTP request that fails to be sent according

10 to the information of the HTTP request that fails to be sent.

In the embodiment of the present application, when sending an HTTP request to the server, the client terminal records information of the HTTP request; when redirection from a first web page to a second web page needs to occur, the client terminal determines whether an HTTP request that fails to be sent exists in a first web page state; if yes, the client terminal acquires information of all
15 the HTTP requests that fail to be sent, and transfers the information to the second web page, such that after redirection to the second web page, the client terminal can resend the HTTP requests that fail to be sent according to the transferred information of the HTTP requests that fail to be sent in the first web page state, thereby avoiding data loss on the server terminal and solving the problem in the prior art.

20 **Brief Description of the Drawings**

FIG. 1 is a flowchart of a resending method for a Hypertext Transfer Protocol request according to an embodiment of the present application;

FIG. 2 is a flowchart of a resending method for a Hypertext Transfer Protocol request according to another embodiment of the present application;

25 FIG. 3 is a hardware structure diagram of an apparatus provided with a resending device for a Hypertext Transfer Protocol request according to the present application; and

FIG. 4 is a block diagram of a resending device for a Hypertext Transfer Protocol request according to an embodiment of the present application.

Detailed Description

Exemplary embodiments will be described in detail herein, and examples thereof are shown in the accompanying drawings. In the following description involving the accompanying drawings, the same numerals in different accompanying drawings denote the same or similar elements, unless specified otherwise. Implementations described in the following exemplary embodiments do not represent all implementations consistent with the present application. In contrast, they are merely examples of devices and methods consistent with some aspects of the present application described in detail in the appended claims.

Terms used in the present application are merely used for describing specific embodiments, instead of limiting the present application. Singular forms "a(n)", "said", and "the" used in the present application and the appended claims also include plural forms, unless clearly specified in the context that other meanings are denoted. It should be further understood that the term "and/or" used in this text refers to and includes any or all possible combinations of one or more associated items listed.

It should be understood that although terms such as "first", "second", and "third" may be used to describe various kinds of information in the present application, these kinds of information should not be limited to the terms. These terms are merely used to distinguish information of the same type from each other. For example, without departing from the scope of the present application, the first information may also be referred to as the second information, and similarly, the second information may also be referred to as the first information. Depending on the context, the word "if" used herein may be explained as "when...", "as...", or "in response to determination".

HTTP is an application-layer protocol in a Transmission Control Protocol/Internet Protocol (TCP/IP) model. When a browser and a server need to communicate with each other, a TCP connection is established first, and the browser sends request information to the server.

Specifically, when a user performs an operation on a current web page on a client terminal, for example, clicking a link on the web page or logging in to an account on the web page, the client terminal will monitor the operation and send corresponding HTTP request information to a server for recording.

5 The server returns corresponding response information after receiving the HTTP request information. If the client terminal receives normal response information returned by the server, it indicates that the HTTP request is sent successfully. If the client terminal receives abnormal response information returned by the server or does not receive response information returned by the server, it indicates that the HTTP request fails to be sent. The reason why the HTTP request
10 fails to be sent may be that the request is cancelled by a browser before being completely sent, and the server does not receive the HTTP request at all, resulting in data loss.

An embodiment of the present application provides a resending method for a Hypertext Transfer Protocol request. The method can record information of all the HTTP requests that fail to be sent before web page redirection, and transfer the information to a next web page after
15 redirection, such that after redirection to the next web page, the client terminal can resend the HTTP requests that fail to be sent according to the information, thereby solving the problem of data loss on the server terminal.

FIG. 1 is a flowchart of a resending method for a Hypertext Transfer Protocol request according to an embodiment of the present application. As shown in FIG. 1, the method may include the
20 following steps:

Step S100: A client terminal sends an HTTP request to a server, and records information of the HTTP request.

Step S101: When the client terminal determines that an HTTP request that fails to be sent exists in a first web page state when redirection from the first web page to a second web page needs to
25 occur, the client terminal acquires information of the HTTP request that fails to be sent.

Step S102: The client terminal transfers, to the second web page, the information of the HTTP request that fails to be sent.

Step S103: After redirection to the second web page, the client terminal resends the HTTP request that fails to be sent according to the information of the HTTP request that fails to be sent.

In the embodiment of the present application, when sending an HTTP request to the server, the client terminal records information of the HTTP request; when redirection from a first web page to a second web page needs to occur, the client terminal determines whether an HTTP request that fails to be sent exists in a first web page state; if yes, the client terminal acquires information of all the HTTP requests that fail to be sent, and transfers the information to the second web page, such that after redirection to the second web page, the client terminal can resend the HTTP requests that fail to be sent according to the transferred information of the HTTP requests that fail to be sent in the first web page state, thereby avoiding data loss on the server terminal and solving the problem in the prior art.

FIG. 2 is a flowchart of a resending method for a Hypertext Transfer Protocol request according to another embodiment of the present application. As shown in FIG. 2, the method may include the following steps:

Step S200: A client terminal monitors an operation performed by a user on a first web page, sends an HTTP request to a server, and records information of the HTTP request.

The HTTP request carries monitoring data of the operation on the first web page.

The first web page here may refer to a web page currently opened by the client terminal. When a user performs an operation on the first web page on the client terminal, for example, clicking a link on the first web page or logging in to an account on the first web page, the client terminal will monitor the operation and send monitoring data to the server terminal in the form of an HTTP request.

It should be noted that in the embodiment of the present application, when sending an HTTP request, the client terminal records complete information of the HTTP request, including a request method, a Uniform Resource Locator (URL) address, the HTTP protocol version and the like, such that after determining that the HTTP request fails to be sent, the client terminal can resend the

HTTP request according to the recorded complete information of the HTTP request, thereby avoiding data loss on the server terminal.

Step S201: The client terminal records the HTTP request in a list of requests that fail to be sent.

In the embodiment of the present application, when sending an HTTP request to the server, the client terminal writes information of the HTTP request into a list of requests that fail to be sent. The list of requests that fail to be sent is stored in the client terminal.

Step S202: When receiving a normal response message returned by the server, the client terminal deletes the HTTP request from the list of requests that fail to be sent; otherwise, when the client terminal receives an abnormal response message returned by the server or does not receive a response message returned by the server, the client terminal maintains the HTTP request recorded in the list of requests that fail to be sent.

Specifically, in general cases, the following situations exist when the server returns a response message for an HTTP request sent by the client terminal:

The status code of the response message is 200, and the message is a normal response message, indicating that the HTTP request is sent successfully; the status code of the response message is not 200, and the message is an abnormal response message, indicating that the HTTP request fails to be sent; no response message is received, which may be caused by the fact that web page redirection occurs before the HTTP request is completely sent, such that the HTTP request is directly cancelled by the client terminal, the server terminal does not receive the HTTP request at all, and this situation indicates that the HTTP request fails to be sent.

To ensure data integrity on the server terminal, data needs to be resent for the HTTP request that fails to be sent in any of the above situations. Therefore, when determining that the HTTP request fails to be sent, the client terminal maintains related information of the HTTP request in the list of requests that fail to be sent, such that on a web page which is subsequently redirected to, the client terminal can resend the HTTP requests that fail to be sent according to records in the list of requests that fail to be sent.

Step S203: When detecting that a current web page needs to be redirected from a first web page to a second web page, the client terminal judges whether the list of requests that fail to be sent is empty, if yes, Step S205 is directly performed; and if not, Step S204 is performed.

Step S204: The client terminal records information of HTTP requests in the list of requests that fail to be sent, and transfers the information to the second web page; Step S205 is performed.

In an actual application, when web page redirection occurs on the client terminal, for example, redirection from a first web page to a second web page, the client terminal will offload the content of the first web page and load the content of the second web page. During the process that the client terminal offloads the first web page, if an HTTP request sent in a first web page state fails to be sent or has not been completely sent, the HTTP request may quite possibly be cancelled, and the server terminal cannot receive monitoring data carried in the HTTP request, resulting in data loss.

In the embodiment of the present application, when detecting that a current web page needs to be redirected, for example, from a first web page to a second web page, the client terminal checks the list of requests that fail to be sent before web page redirection to determine whether the list of requests is empty.

If the list of requests is empty, it indicates that all the HTTP requests sent in the first web page state are sent successfully, the phenomenon of data loss does not exist on the server terminal, data does not need to be resent, and redirection to the second web page can be directly performed.

If the list of requests is not empty, it indicates that the HTTP requests included in the list of requests fail to be sent or may not be completely sent in the first web page state, and the server may not receive the monitoring data carried in the HTTP requests. To prevent data loss, the client terminal transfers information of the HTTP requests included in the list of requests to the second web page, such that after redirection to the second web page, the client terminal can resend the HTTP requests according to the information of the HTTP requests.

It should be noted that in the embodiment of the present application, the list of requests that fail to be sent records complete information of all the HTTP requests that fail to be sent, thereby ensuring that the HTTP requests are resent successfully.

Further, in the embodiment of the present application, the step of recording, by the client terminal, information of HTTP requests in the list of requests that fail to be sent, and transferring the information to the second web page may include: recording, by the client terminal, information of HTTP requests in the list of requests that fail to be sent, and writing the information in a window name corresponding to the second web page.

If the redirection from the first web page to the second web page is redirection in a same window, for example, an operation of clicking "opening a link on a new tab page" by a user or performing a login operation by a user after submitting an account and a password on the first web page, the window name corresponding to the second web page is identical to the window name corresponding to the first web page, that is, the name of the window currently opened by the client terminal. At this time, only the information of the HTTP requests in the list of requests that fail to be sent needs to be written in the name of the currently opened window.

If the redirection from the first web page to the second web page is redirection between different windows, for example, an operation of clicking "opening a link in a new window" by a user, the window name corresponding to the second web page is different from the window name corresponding to the first web page. At this time, the client terminal acquires, according to a redirection instruction, the window name corresponding to the second web page, that is, the name of the window to be redirected to, and writes the information of the HTTP requests in the list of requests that fail to be sent in the name of the window to be redirected to.

Step S205: After redirection to the second web page, the client terminal acquires, from the first web page, the information of the HTTP requests in the list of requests that fail to be sent, and resends the HTTP requests according to the information of the HTTP requests.

In the embodiment of the present application, in Step S204, the information of the HTTP requests in the list of requests that fail to be sent has been written in the window name corresponding to the second web page. When web page redirection is completed on the client terminal, that is, redirection to the second web page has been performed, the client terminal can acquire, from the window name corresponding to the second web page, the information of the

HTTP requests, corresponding to the first web page, in the list of requests that fail to be sent. The information of the HTTP requests is the information of all the HTTP requests that fail to be sent on the first web page, that is, the information of the HTTP requests from which the server does not receive the monitoring data in the first web page state. The client terminal can resend the HTTP requests that fail to be sent in the first web page state according to the information of the HTTP requests, thereby ensuring integrity of data received by the server terminal and solving the problem in the prior art.

In the embodiment of the present application shown in FIG. 2, an HTTP request is recorded in a list of requests that fail to be sent. When receiving a normal response message returned by the server, the client terminal deletes the HTTP request from the list of requests that fail to be sent; otherwise, when the client terminal receives an abnormal response message returned by the server or does not receive a response message returned by the server, the client terminal maintains the HTTP request recorded in the list of requests that fail to be sent; when detecting that a current web page needs to be redirected from a first web page to a second web page, the client terminal determines whether an HTTP request that fails to be sent exists in a first web page state by judging whether the list of requests that fail to be sent is empty; if the list of requests that fail to be sent is not empty, the client terminal transfers, to the second web page, information of all the HTTP requests in the list of requests that fail to be sent, such that the client terminal can resend the HTTP requests that fail to be sent in the first web page state according to the information of the HTTP requests transferred to the second web page.

In another embodiment of the present application, an HTTP request may also be recorded in an HTTP request cache pool. When receiving a normal response message returned by the server, the client terminal deletes the HTTP request from the HTTP request cache pool; otherwise, when the client terminal receives an abnormal response message returned by the server or does not receive a response message returned by the server, the client terminal maintains the HTTP request recorded in the HTTP request cache pool; when detecting that a current web page needs to be redirected from a first web page to a second web page, the client terminal determines whether an

HTTP request that fails to be sent exists in a first web page state by judging whether the HTTP request cache pool is empty; if the HTTP request cache pool is not empty, the client terminal transfers, to the second web page, information of all the HTTP requests in the HTTP request cache pool, such that the client terminal can resend the HTTP requests that fail to be sent in the first web page state according to the information of the HTTP requests transferred to the second web page.

In another embodiment of the present application, an HTTP request may also be recorded in a list of HTTP requests, and the state of the HTTP request is marked as failing to be sent successfully. When receiving a normal response message returned by the server, the client terminal modifies the state of the HTTP request in the list of HTTP requests as being sent successfully; otherwise, when the client terminal receives an abnormal response message returned by the server or does not receive a response message returned by the server, the client terminal maintains the state of the HTTP request in the list of HTTP requests as failing to be sent successfully; when detecting that a current web page needs to be redirected from a first web page to a second web page, the client terminal determines whether an HTTP request that fails to be sent exists in a first web page state by judging whether the HTTP request with its state being marked as failing to be sent successfully exists in the list of HTTP requests; if the HTTP request with its state being marked as failing to be sent successfully exists, the client terminal transfers, to the second web page, information of the HTTP request with its state being marked as failing to be sent successfully, such that the client terminal can resend the HTTP requests that fail to be sent in the first web page state according to the information of the HTTP requests transferred to the second web page.

It should be noted that in the embodiment of the present application, the client terminal can update the list of requests that fail to be sent whenever receiving the normal response information returned by the server terminal, and delete, from the list of requests, the HTTP request corresponding to the normal response information.

Definitely, in an actual application, if the client terminal updates the list of requests whenever receiving the normal response information returned by the server terminal, the client terminal will operate too frequently, and in the embodiment of the present application, when determining to

perform web page redirection, the client terminal can initiate checking of request completeness and uniformly update the list of requests. In consideration of the embodiment shown in FIG. 2, when determining to perform web page redirection, the client terminal judges, among all the HTTP requests sent in the current web page state, the specific HTTP requests for which the normal response information returned by the server is received, and deletes, from the list of requests that fail to be sent, the HTTP requests for which the normal response information is received.

Corresponding to the method embodiment of the present application, the present application further provides embodiments of a device and a server.

The embodiment of a resending device for a Hypertext Transfer Protocol request according to the present application can be applied in a server. The device embodiment may be implemented by software, hardware, or a combination thereof. Taking software implementation for example, the resending device, as a logical device, is formed in a manner that a processor of an apparatus provided with the resending device reads corresponding computer program instructions in a non-volatile memory into a memory for running therein. From the aspect of hardware, FIG. 3 is a hardware structure diagram of an apparatus provided with a resending device for a Hypertext Transfer Protocol request according to the present application. In addition to the processor, memory, network interface, and non-volatile memory shown in FIG. 3, the apparatus provided with the device in the embodiment may also include other hardware according to the actual functions of the apparatus, for example, as for a client terminal apparatus, the apparatus may include a camera, a touch screen, a communication component and the like; while for a server, the apparatus may include a forwarding chip responsible for processing packets and the like.

FIG. 4 is a block diagram of a resending device for a Hypertext Transfer Protocol request according to an embodiment of the present application. The device can be applied in a server.

The device may include: a first sending unit 400, a first recording unit 401, an acquisition unit 402, a transferring unit 403, and a second sending unit 404.

The first sending unit 400 is configured to send a Hypertext Transfer Protocol HTTP request to a server.

The first recording unit 401 is configured to record information of the HTTP request.

The acquisition unit 402 is configured to, if it is determined that an HTTP request that fails to be sent exists in a first web page state when redirection from the first web page to a second web page needs to occur, acquire information of the HTTP request that fails to be sent.

- 5 The transferring unit 403 is configured to transfer, to the second web page, the information of the HTTP request that fails to be sent.

The second sending unit 404 is configured to, after redirection to the second web page, resend the HTTP request that fails to be sent according to the information of the HTTP request that fails to be sent.

- 10 The HTTP request that fails to be sent is an HTTP request for which successful response information returned by the server is not received.

The device may further include: a second recording unit.

The second recording unit is configured to record, in the first web page state, all the HTTP requests for which normal response information returned by the server is not received.

- 15 The second recording unit may include: a first request recording sub-unit and a first deleting sub-unit.

The first request recording sub-unit is configured to, when an HTTP request is sent, record the HTTP request in a list of requests that fail to be sent.

- 20 The first deleting sub-unit is configured to delete the HTTP request from the list of requests that fail to be sent when normal response information returned by the server is received.

The second recording unit may include: a second request recording sub-unit and a second deleting sub-unit.

The second request recording sub-unit is configured to, when an HTTP request is sent, record the HTTP request in an HTTP request cache pool.

- 25 The second deleting sub-unit is configured to delete the HTTP request from the HTTP request cache pool when normal response information returned by the server is received.

The transferring unit may include: a writing sub-unit.

The writing sub-unit is configured to write the information of the HTTP request that fails to be sent in a window name corresponding to the second web page.

For the implementation of the functions and effects of each unit in the device, reference may be made to the implementation of the corresponding steps in the method, and the details will not
5 be repeated herein.

The present application further provides a client terminal, including: a processor; and a memory configured to store executable instructions of the processor.

The processor is configured to: send a Hypertext Transfer Protocol HTTP request to a server, and record information of the HTTP request; if it is determined that an HTTP request that fails to
10 be sent exists in a first web page state when redirection from the first web page to a second web page needs to occur, acquire information of the HTTP request that fails to be sent; transfer, to the second web page, the information of the HTTP request that fails to be sent; after redirection to the second web page, resend the HTTP request that fails to be sent according to the information of the HTTP request that fails to be sent.

15 In the embodiment of the present application, when sending an HTTP request to the server, the client terminal records information of the HTTP request; when redirection from a first web page to a second web page needs to occur, the client terminal determines whether an HTTP request that fails to be sent exists in a first web page state; if yes, the client terminal acquires information of all the HTTP requests that fail to be sent, and transfers the information to the second web page, such
20 that after redirection to the second web page, the client terminal can resend the HTTP requests that fail to be sent according to the transferred information of the HTTP requests that fail to be sent in the first web page state, thereby avoiding data loss on the server terminal and solving the problem in the prior art.

It should be noted that the method and device according to the embodiments of the present
25 application can be applied to a browser or a container similar to a browser, for example, a WebView container generally used in a mobile device.

The device embodiment is basically corresponding to the method embodiment, such that for related parts, reference may be made to the description of the parts in the method embodiment. The device embodiment described above is merely exemplary, where units described as separated parts may be or may not be physically separated, parts displayed as units may be or may not be physical units, and they may be located at the same place or distributed to multiple network units. The objective of the solutions of the present application may be implemented by selecting a part of or all modules of the device according to actual requirements. Those of ordinary skill in the art may understand and implement the present application without creative efforts.

Those skilled in the art can easily obtain other implementations of the present application after considering the specification and practicing the invention disclosed herein. The present application is intended to cover any variations, usages, or adaptive changes of the present application, and these variations, usages, or adaptive changes follow general principles of the present application and include common knowledge or conventional technical means in this technical field that are not disclosed in the present application. The specification and embodiments are considered as merely exemplary, and the real scope and spirit of the present application are defined by the following claims.

It should be understood that the present application is not limited to precise structures that have been described above and shown in the accompanying drawings, and can have various modifications and variations without departing from the scope of the present application. The scope of the present application is merely limited by the appended claims.

CLAIMS

1. A resending method for a Hypertext Transfer Protocol request, the method comprising:
sending, by a client terminal, a Hypertext Transfer Protocol HTTP request to a server, and
recording information of the HTTP request;

if it is determined that an HTTP request that fails to be sent exists in a first web page state
when redirection from the first web page to a second web page needs to occur, acquiring
information of the HTTP request that fails to be sent;

transferring, to the second web page, the information of the HTTP request that fails to be sent;
and

after redirection to the second web page, resending the HTTP request that fails to be sent
according to the information of the HTTP request that fails to be sent.

2. The method of claim 1, wherein the HTTP request that fails to be sent is an HTTP request
for which successful response information returned by the server is not received.

3. The method of claim 2, further comprising:
recording, in the first web page state, all the HTTP requests for which normal response
information returned by the server is not received.

4. The method of claim 3, wherein the step of recording all the HTTP requests for which
normal response information returned by the server is not received comprises:

when the client terminal sends an HTTP request, recording the HTTP request in a list of
requests that fail to be sent; and

deleting the HTTP request from the list of requests that fail to be sent when normal response
information returned by the server is received.

5. The method of claim 3, wherein the step of recording all the HTTP requests for which
normal response information returned by the server is not received comprises:

when the client terminal sends an HTTP request, recording the HTTP request in an HTTP
request cache pool; and

deleting the HTTP request from the HTTP request cache pool when normal response

information returned by the server is received.

6. The method of claim 1, wherein the step of transferring, to the second web page, the information of the HTTP request that fails to be sent comprises:

writing the information of the HTTP request that fails to be sent in a window name corresponding to the second web page.

7. A resending device for a Hypertext Transfer Protocol request, comprising:

a first sending unit configured to send a Hypertext Transfer Protocol HTTP request to a server;

a first recording unit configured to record information of the HTTP request;

an acquisition unit configured to, if it is determined that an HTTP request that fails to be sent exists in a first web page state when redirection from the first web page to a second web page needs to occur, acquire information of the HTTP request that fails to be sent;

a transferring unit configured to transfer, to the second web page, the information of the HTTP request that fails to be sent; and

a second sending unit configured to, after redirection to the second web page, resend the HTTP request that fails to be sent according to the information of the HTTP request that fails to be sent.

8. The resending device for a Hypertext Transfer Protocol request of claim 7, wherein the HTTP request that fails to be sent is an HTTP request for which successful response information returned by the server is not received.

9. The resending device for a Hypertext Transfer Protocol request of claim 8, further comprising:

a second recording unit configured to record, in the first web page state, all the HTTP requests for which normal response information returned by the server is not received.

10. The resending device for a Hypertext Transfer Protocol request of claim 9, wherein the second recording unit comprises:

a first request recording sub-unit configured to, when an HTTP request is sent, record the

HTTP request in a list of requests that fail to be sent; and

a first deleting sub-unit configured to delete the HTTP request from the list of requests that fail to be sent when normal response information returned by the server is received.

11. The resending device for a Hypertext Transfer Protocol request of claim 9, wherein the second recording unit comprises:

a second request recording sub-unit configured to, when an HTTP request is sent, record the HTTP request in an HTTP request cache pool; and

a second deleting sub-unit configured to delete the HTTP request from the HTTP request cache pool when normal response information returned by the server is received.

12. The resending device for a Hypertext Transfer Protocol request of claim 7, wherein the transferring unit comprises:

a writing sub-unit configured to write the information of the HTTP request that fails to be sent in a window name corresponding to the second web page.

13. A client terminal, comprising:

a processor; and a memory configured to store executable instructions of the processor, wherein the processor is configured to:

send a Hypertext Transfer Protocol HTTP request to a server, and record information of the HTTP request;

if it is determined that an HTTP request that fails to be sent exists in a first web page state when redirection from the first web page to a second web page needs to occur, acquire information of the HTTP request that fails to be sent;

transfer, to the second web page, the information of the HTTP request that fails to be sent; and

after redirection to the second web page, resend the HTTP request that fails to be sent according to the information of the HTTP request that fails to be sent.