CLAIMS

1. A logging method, comprising:

intercepting, by a server using a first asynchronous thread, an Application Programming Interface (API) calling request sent by a client;

generating, by the server, log data based on the API calling request, and storing the log data in a log storage queue, the log storage queue being configured to store log data to be filtered for storage in a database; and

implementing, by the server, real-time recording of a scene state using a service thread in response to a modification request in the API calling request,

wherein the method further comprises:

filtering, by the server, the log data in the log storage queue based on a preset first filtering parameter, and storing the filtered log data in the database in batches.

2. The logging method of claim 1, further comprising:

obtaining, by the server, user permission information using the first asynchronous thread according to a request header obtained by parsing the API calling request, and authenticating the user permission information to obtain an authentication result,

wherein generating, by the server, the log data based on the API calling request comprises:

in a case where the authentication result represents successful authentication, generating, by the server, the log data using the first asynchronous thread based on the API calling request.

3. The logging method of any one of claims 1-2, wherein generating, by the server, the log data based on the API calling request comprises:

extracting, by the server, first data from a request header obtained by parsing each calling request in the API calling request, the first data at least comprising user identity data and resource path data;

extracting, by the server, second data from a request body and request line obtained by parsing each calling request, the second data at least comprising request method data and request content data; and

combining, by the server, the first data and the second data to obtain a piece of log data corresponding to each calling request.

4. The logging method of claim 1, wherein at least one piece of log data is stored in the log storage queue, and each piece of log data comprises a log parameter; and

filtering, by the server, the log data in the log storage queue based on the preset first filtering parameter and storing the filtered log data in the database in batches comprises:

acquiring, by the server, the at least one piece of log data from the log storage queue,

matching, by the server, the log parameter of the each piece of log data in the at least one piece of log data and the first filtering parameter to obtain a matched matching parameter,

determining, by the server, the filtered log data according to at least one piece of target log data corresponding to the matching parameter, and

when the amount of the filtered log data reaches a preset amount, storing, by the server, the filtered log data in the database.

5. The logging method of claim 4, wherein acquiring, by the server, the at least one piece of log data from the log storage queue comprises:

reading, by the server, the log data from the log storage queue according to a preset frequency; and

in a case where present log data is read, continuing, by the server, to read a next piece

of log data from the log storage queue until a last piece of log data stored in the log storage queue is read, thereby acquiring, by the server, the at least one piece of log data.

6. The logging method of claim 4, wherein the matching, by the server, parameter comprises at least one of:

a method parameter, a resource path resource, a resource path prefix parameter, a non-target parameter in a parameter blacklist, or a target parameter in a parameter whitelist.

7. The logging method of claim 4, wherein determining, by the server, the filtered log data according to the at least one piece of target log data corresponding to the matching parameter comprises:

determining, by the server, candidate log data according to the at least one piece of target log data corresponding to the matching parameter; and

determining, by the server, the filtered log data based on the candidate log data.

8. The logging method of claim 7, wherein determining, by the server, the candidate log data according to the at least one piece of target log data corresponding to the matching parameter comprises at least one of:

in a case where the matching parameter is the method parameter, determining, by the server, at least one piece of target log data corresponding to the method parameter in the at least one piece of log data as the candidate log data;

in a case where the matching parameter is the resource path parameter, determining, by the server, at least one piece of target log data corresponding to the resource path parameter in the at least one piece of log data as the candidate log data;

in a case where the matching parameter is the resource path prefix parameter, determining, by the server, at least one piece of target log data corresponding to the resource path prefix parameter in the at least one piece of log data as the candidate log data;

in a case where the matching parameter is the at least one non-target parameter in the parameter blacklist, determining, by the server, log data, except at least one piece of target log data corresponding to the at least one non-target parameter, in the at least one piece of log data as the candidate log data; or

in a case where the matching parameter is the at least one target parameter in the parameter whitelist, determining, by the server, at least one piece of target log data corresponding to the at least one target parameter in the at least one piece of log data as the candidate log data.

9. The logging method of claim 1 or any one of claims 4-7, further comprising:

receiving, by the server, a filtering parameter setting instruction containing the preset first filtering parameter; and

modifying, by the server, a preset first initial filtering parameter to the preset first filtering parameter in response to the filtering parameter setting instruction.

10. The logging method of any one of claims 1-9, wherein implementing, by the server, real-time recording of the scene state using the service thread in response to the modification request in the API calling request comprises:

modifying, by the server, a working parameter of the service thread using the service thread in response to the modification request in the API calling request to obtain a modified working parameter;

determining, by the server, a real-time state of a scene based on the modified working parameter and received real-time scene data acquired by an acquisition device; and

comparing, by the server, the real-time state and a preset state to obtain a comparison result, and recording the comparison result in real time.

11. The logging method of claim 10, further comprising:

in a case where the comparison result represents that the real-time state is abnormal, outputting, by the server, prompt information.

12. An electronic device for use in a server, comprising:

a memory, configured to store an executable computer program; and

a processor, configured to execute the executable computer program stored in the memory to implement following operations:

intercepting, using a first asynchronous thread, an Application Programming Interface (API) calling request sent by a client;

generating log data based on the API calling request, and storing the log data in a log storage queue, the log storage queue being configured to store log data to be filtered for storage in a database; and

implementing real-time recording of a scene state using a service thread in response to a modification request in the API calling request,

wherein the processor is further configured to implement following operations:

filtering the log data in the log storage queue based on a preset first filtering parameter, and storing the filtered log data in the database in batches.

13. The electronic device of claim 12, wherein the processor is further configured to implement following operations:

obtaining user permission information using the first asynchronous thread according to a request header obtained by parsing the API calling request, and authenticating the user

permission information to obtain an authentication result,

wherein generating the log data based on the API calling request comprises:

in a case where the authentication result represents successful authentication, generating the log data using the first asynchronous thread based on the API calling request.

14. The electronic device of any one of claims 11-13, wherein the operation of generating the log data based on the API calling request comprises:

extracting first data from a request header obtained by parsing each calling request in the API calling request, the first data at least comprising user identity data and resource path data;

extracting second data from a request body and request line obtained by parsing each calling request, the second data at least comprising request method data and request content data; and

combining the first data and the second data to obtain a piece of log data corresponding to each calling request.

15. The electronic device of claim 12, wherein at least one piece of log data is stored in the log storage queue, and each piece of log data comprises a log parameter; and

the operations of filtering the log data in the log storage queue based on the preset first filtering parameter and storing the filtered log data in the database in batches comprise:

acquiring the at least one piece of log data from the log storage queue,

matching the log parameter of the each piece of log data in the at least one piece of log data and the first filtering parameter to obtain a matched matching parameter,

determining the filtered log data according to at least one piece of target log data corresponding to the matching parameter, and

when the amount of the filtered log data reaches a preset amount, storing the filtered log data in the database.

16. The electronic device of claim 12 or 15, wherein the processor is further configured to implement following operations:

receiving a filtering parameter setting instruction containing the preset first filtering parameter; and

modifying a preset first initial filtering parameter to the preset first filtering parameter in response to the filtering parameter setting instruction.

17. A computer-readable storage medium, storing a computer program, configured to be executed by a processor of a server to implement following operations:

intercepting, using a first asynchronous thread, an Application Programming Interface (API) calling request sent by a client;

generating log data based on the API calling request, and storing the log data in a log storage queue, the log storage queue being configured to store log data to be filtered for storage in a database; and

implementing real-time recording of a scene state using a service thread in response to a modification request in the API calling request,

wherein the operations further comprise:

filtering the log data in the log storage queue based on a preset first filtering parameter, and storing the filtered log data in the database in batches.

18. A computer program, comprising computer-readable codes which, when executed in an electronic device for use in a server, cause a processor in the electronic device to implement following operations:

intercepting, using a first asynchronous thread, an Application Programming Interface (API) calling request sent by a client;

generating log data based on the API calling request, and storing the log data in a log storage queue, the log storage queue being configured to store log data to be filtered for storage in a database; and

implementing real-time recording of a scene state using a service thread in response to a modification request in the API calling request,

wherein the operations further comprise:

filtering the log data in the log storage queue based on a preset first filtering parameter, and storing the filtered log data in the database in batches.