APPLICATION PROGRAMMING INTERFACE ACCESS METHOD AND APPARATUS, STORAGE MEDIUM, AND COMPUTER SYSTEM

TECHNICAL FIELD

[0001] This specification relates to the field of computer technologies, and in particular, to an application programming interface access method and apparatus, a storage medium, and a computer system.

BACKGROUND

[0002] When an application system may be invoked by a client or another system, application programming interfaces (APIs) usually need to be exposed. These APIs represent various function points of the application system. If a mediator is added between two systems or between a client and a system to coordinate the invocation of the API, so that all requests entering the system pass through the mediator, the mediator is an application programming interface gateway (API Gateway). The API gateway can help better manage and control API invocations, to improve system security and performance. However, the existing manner of accessing an API by using an API gateway still has the problem of not being able to meet different access requirements of different users for API interfaces.

SUMMARY

[0003] Embodiments of this specification provide an application programming interface access method and apparatus, a storage medium, and a computer system, so as to partially resolve the foregoing problems in the prior art.

[0004] This specification provides an application programming interface access method, executed by an application programming interface gateway, including: receiving a first request sent by a target user for a target application programming interface, where the first request includes a target user field; and determining a target factor field corresponding to the target user field based

on a pre-stored field correspondence, where the field correspondence indicates a correspondence between a factor field provided in the target application programming interface and a user field that matches the target user; and obtaining field content of the target factor field from a serving end corresponding to the target application programming interface based on the target factor field, and sending the field content of the target factor field to the target user as field content of the target user field.

[0005] This specification provides a computer system, including a memory, a processor, and a computer program stored in the memory and running on the processor. When executing the program, the processor implements the following steps: receiving a first request sent by a target user for a target application programming interface, where the first request includes a target user field; and determining a target factor field corresponding to the target user field based on a pre-stored field correspondence, where the field correspondence indicates a correspondence between a factor field provided in the target application programming interface and a user field that matches the target user; and obtaining field content of the target factor field from a serving end corresponding to the target application programming interface based on the target factor field, and sending the field content of the target factor field to the target user as field content of the target user field.

[0006] This specification provides an application programming interface access apparatus. The apparatus includes: a receiving unit, configured to: receive a first request sent by a target user for a target application programming interface, where the first request includes a target user field; and determine a target factor field corresponding to the target user field based on a pre-stored field correspondence, where the field correspondence indicates a correspondence between a factor field provided in the target application programming interface and a user field that matches the target user; and a sending unit, configured to: obtain field content of the target factor field from a serving end corresponding to the target application programming interface based on the target factor field, and send the field content of the target factor field to the target user as field content of the target user field.

[0007] This specification provides a computer-readable nonvolatile storage medium, where the storage medium stores a computer program, and when the computer program is executed by a processor, the foregoing application programming interface access method is implemented.

[0008] According to the foregoing at least one technical solution used in the embodiments of

this specification, at least one of the following beneficial effects can be achieved: Different users can access a same application programming interface by using user fields that are respectively defined by the users, thereby improving user experience of accessing the application programming interface by the users.

BRIEF DESCRIPTION OF DRAWINGS

[0009] The accompanying drawings described herein are used to provide further understanding of the specification and constitute a part of the specification. The example embodiments of the specification and descriptions thereof are used to explain the specification, and do not constitute an undue limitation on the specification. In the accompanying drawings:

[0010] FIG. 1 is a schematic diagram of an application interface access solution;

[0011] FIG. 2 is a schematic diagram of an application programming interface access method in this specification;

[0012] FIG. 3 is a schematic flowchart of a multi-mode data processing method in this specification;

[0013] FIG. 4A is a schematic diagram of receiving a user field input by a user in this specification;

[0014] FIG. 4B is a schematic diagram of receiving a user field input by another user in this specification;

[0015] FIG. 5A is a schematic diagram of receiving a factor field selected by a user in this specification;

[0016] FIG. 5B is a schematic diagram of receiving a factor field selected by another user in this specification;

[0017] FIG. 6 is a schematic flowchart of obtaining field content of a factor field in this specification;

[0018] FIG. 7 is a schematic diagram of a structure of an application programming interface access apparatus in this specification; and

[0019] FIG. 8 is a schematic diagram of a structure of a computer system according to an embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0020] To make the objectives, technical solutions, and advantages of the specification clearer, the following clearly and completely describes the technical solutions of the specification with reference to specific embodiments of the specification and corresponding accompanying drawings. It is clear that the described embodiments are merely some but not all of embodiments of this specification. All other embodiments obtained by a person of ordinary skill in the art based on embodiments of this specification without creative efforts shall fall within the protection scope of this specification.

[0021] As described above, when an application system may be invoked by a client or another system, application programming interfaces (APIs) usually need to be exposed. These APIs represent various function points of the application system. If a mediator is added between two systems or between a client and a system to coordinate the invocation of the API, so that all requests entering the system can pass through the mediator, the mediator is an application programming interface gateway (API Gateway). FIG. 1 is a schematic diagram of an application interface access solution. As shown in FIG. 1, for example, all clients that wish to access different APIs on a serving end send their access requests to an API gateway, and the API gateway forwards their access requests to different application programming interfaces on the serving end for processing, and after receiving processing results sent from the serving end, returns the processing results to the respective clients.

[0022] The API gateway can help better manage and control API invocations, to improve system security and performance. However, the existing manner of accessing an API by using an API gateway still has the problem of not being able to meet different access requirements of different users for API interfaces. For example, when different users access a same API, they expect to obtain values of different fields, or different users expect to set names of fields in the API interface according to definitions of the different users. The existing solution of accessing an API through an API gateway is difficult to achieve the foregoing technical effect desired by the user.

[0023] To resolve the foregoing problem, in this application, a correspondence between a user field named by the user and an original field (or referred to as a factor field) in the system may be pre-stored. FIG. 2 is a schematic diagram of an application programming interface access method

in this specification. As shown in FIG. 2, for example, when the API receives a request for accessing the API sent by a user through, for example, a client, the access request from the client may include a user field whose field content is desired to be obtained. An API gateway may change the user field into a factor field according to a pre-stored correspondence between a user field and a factor field, and send the factor field to a serving end of the API. After obtaining field data of the factor field, the serving end replaces the factor field with the corresponding user field, and returns the user field and the field data to the client.

[0024] In the conventional solution using an API, all users access a unified interface, and all accessible fields in the interface are set and cannot be changed for each user. Therefore, values of different fields cannot be obtained when different users access the API; and different users cannot obtain a value of a same field in the API by using different field names that are desired by the different users. In the application programming interface access method provided in this application, the correspondence between a user field of an API and a factor field is pre-stored. When receiving the request for accessing the API sent by the client, the API gateway may convert the user field in the request into a factor field, and obtain the field content of the factor field from the API serving end by using the factor field. In addition, the field content of the factor field is used as field content of the user field, and is sent back to the client. Therefore, actually, different users can access a same API by using different user fields defined by the users, thereby improving user experience of accessing the API by the users.

[0025] In addition, different users may select different fields from the same API for the users to access the API. In this way, the same API may have different field combinations for different users, so that access requirements of different users for differentiated field combinations are met, and user experience of accessing the API by the users is further improved.

[0026] The following describes in detail the technical solutions provided in the embodiments of this specification with reference to the accompanying drawings.

[0027] FIG. 3 is a schematic flowchart of a service prediction method in this specification. The method is performed by a first party, and may specifically include the following steps.

[0028] S300: Receive a first request sent by a target user for a target application programming interface, where the first request includes a target user field; and determine a target factor field corresponding to the target user field based on a pre-stored field correspondence, where the field correspondence indicates a correspondence between a factor field provided in the target

application programming interface and a user field that matches the target user. The application programming interface (API) is usually a software interface that is used by an invocation party to access and invoke various functions of an application system. In different embodiments, the target application programming interface may be a programming interface configured to invoke different specific service functions or system functions of different service systems. In different embodiments, the target user may be different specific users, for example, may be an individual user or a user belonging to a target group.

[0029] After the request (the first request) sent by the target user for the target application programming interface is received, the request includes one or more data fields that the target user wishes to access through the target application programming interface. A data field is each component of a data structure that a user wishes to return by accessing an API. An API is typically used to transfer data between different software application programs (for example, between a client application and a server end application, or between different system serving end applications). The data is typically returned in a structured format (e.g., Jason and XML) and contains a plurality of fields, each representing specific data or information. In this specification, these fields are also referred to as fields of an application programming interface. A factor field refers to an original field included in the API, that is, a field predefined by the API provider. For example, the API provider may generally set a name or an identifier of the original field. In different embodiments, the target application programming interface may include different factor fields.

[0030] A user field may be a field that matches the factor field and is set by the target user. A matched user field that is set by the target user for a factor field, and a correspondence or an association relationship between the user field and the factor field may be pre-stored. In different embodiments, specific manners of obtaining and storing the foregoing association relationship may be different. In an embodiment, a user field that is input by the target user and corresponding to a factor field provided in the target application programming interface may be obtained, and a correspondence between the factor field and the user field is stored. Specifically, in an embodiment, a factor field may be further displayed to the user, and then a user field that matches the factor field and that is input by the user is received. FIG. 4A is a schematic diagram of receiving a user field input by a user in this specification. As shown in FIG. 4A, the target application interface may include, for example, a plurality of factor fields, for example, a factor field 1, a factor field 2,

and so on. These factor fields may be displayed to a user (for example, a user A), and then user fields that are input by the user A and that respectively match the factor fields are received, for example, a user field a1 that matches the factor field 1, a user field a2 that matches the factor field 2, and so on. Different users may set different user fields for matching a same field in the target application programming interface. FIG. 4B is a schematic diagram of receiving a user field input by another user in this specification. As shown in FIG. 4B, factor fields may be displayed to a user B, and then user fields that are input by the user B and that respectively match the factor fields are received, for example, a user field b1 that matches the factor field 1, a user field b2 that matches the factor field 2, and so on. For a same factor field, different user fields are set for different users, so that the different users can access the same API according to desired field names or identifiers, and obtain field content of the field in the API, thereby improving user experience. In a specific embodiment, when a factor field is displayed to a user, a name, an identifier, a field description, and the like of the factor field may be specifically displayed, so that the user determines a corresponding user field as the user expects.

[0031] In an embodiment, a correspondence between some factor fields of the target application programming interface and user fields that match the target user may be pre-stored. The some factor fields may be determined by the target user from all factor fields of the target application programming interface, and the some factor fields include the target user field. FIG. 5A is a schematic diagram of receiving a factor field selected by a user in this specification. As shown in FIG. 5A, the target application interface may include, for example, a plurality of factor fields, for example, a factor field 1, a factor field 2, ..., and a factor field 10. All the factor fields may be displayed to a user (for example, a user A), and then the factor field 1 to the factor field 5 that are selected by the user A are received as factor fields of the target application programming interface for the user A. Different users may select different factor field sets for the target application programming interface. FIG. 5B is a schematic diagram of receiving a factor field selected by another user in this specification. As shown in FIG. 5B, all the factor fields that are the same as those in FIG. 5A may be displayed to a user (for example, a user B), and then the factor field 1, the factor field 2, the factor field 6, and the factor field 7 that are selected by the user B are received as factor fields of the target application programming interface for the user B. In a specific embodiment, when the factor field set is displayed to the user, an association between the factor fields may be specifically displayed, so that the user determines, according to a relationship

between different factor fields, a factor field that the user wishes to reserve. For a same API, different factor field sets are specified for different users, so that different users can access the API according to required field sets and obtain field content of a required field, without needing to obtain field content that is not required in the API. Not only user experience is improved, but also computing resources consumed for data processing and network resources consumed for data transmission are reduced.

[0032] In some scenarios, according to a service corresponding to the target application programming interface, all factor fields of the target application programming interface may include mandatory factor fields (that is, factor fields that need to be obtained for accessing the target application programming interface) and optional factor fields (that is, factor fields that may or may not be obtained for accessing the target application programming interface). In this scenario, the factor fields selected by the user may include all the mandatory factor fields and all or a part of the optional factor fields. For example, as shown in FIG. 5A and FIG. 5B, for example, the factor field 1 and the factor field 2 are mandatory fields. Therefore, both the user A and the user B select them as factor fields used by themselves. The factor field 3 to the factor field 10 are optional fields. Therefore, the user A may select only the factor field 3, the factor field 4, and the factor field 5 as factor fields used by the user A, and the user B may select the factor field 6 and the factor field 7 as factor fields used by the user B.

[0033] The application programming interface gateway is a mediator of the client or another system access API, and access requests for the API may all pass through the application programming interface gateway. In an embodiment, the application programming interface access method in this application may be performed by the application programming interface gateway. In some scenarios, the application programming interface gateway may be configured to coordinate access to, for example, a plurality of application programming interfaces for a plurality of services. Therefore, in an embodiment, the application programming interface gateway may be configured to receive requests for a plurality of application programming interfaces, and the plurality of application programming interfaces may include a target application programming interface.

[0034] S302: Obtain field content of the target factor field from a serving end corresponding to the target application programming interface based on the target factor field, and send the field content of the target factor field to the target user as field content of the target user field.

[0035] In different embodiments, the serving end of the target application programming interface may be a serving end component of the application programming interface, or a server that runs the foregoing serving end component. In different embodiments, a specific type of the server that runs the serving end component may also be different. In an embodiment, the server may be, for example, one or more of a physical server, a logical server, or a cloud server. In another embodiment, the serving end of the target application programming interface may alternatively be a cluster of the foregoing servers that run the serving end component.

[0036] Further, the foregoing step S302 of obtaining the field content of the target factor field from the serving end corresponding to the target application programming interface in FIG. 3 may be specifically shown in FIG. 4, that is, S400: Send a second request to the serving end corresponding to the target application programming interface, where the second request includes the target factor field. Data is generally organized or stored in a factor field at the serving end of the target application programming interface. Therefore, a request for field content may be sent to the serving end according to the factor field. S402: Receive the field content of the target factor field that is sent by the serving end. In different specific embodiments, specific field content of the target factor field may be different according to different target factor fields, which is not limited in this specification.

[0037] The foregoing is the application programming interface access method provided in the embodiments of this specification. Based on the same idea, this specification further provides a corresponding application programming interface access apparatus. As shown in FIG. 7, the application programming interface access apparatus includes: an obtaining module 702, configured to: receive a first request sent by a target user for a target application programming interface, where the first request includes a target user field; and determine a target factor field corresponding to the target user field based on a pre-stored field correspondence, where the field correspondence indicates a correspondence between a factor field provided in the target application programming interface and a user field that matches the target user; and a construction module 704, configured to: obtain field content of the target factor field from a serving end corresponding to the target application programming interface based on the target factor field, and send the field content of the target factor field to the target user as field content of the target user field.

[0038] For ease of description, the foregoing apparatus is described separately based on

modules according to functions. Certainly, during implementation of this specification, functions of modules may be implemented in one or more software and/or hardware devices.

[0039] The specification also provides a computer-readable non-transitory storage medium, wherein the storage medium stores a computer program, which, when executed by a processor, can be used to perform one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Herein, the computer-readable non-transitory storage medium or media may include one or more semiconductor-based or other integrated circuits (ICs) (such, as for example, field-programmable gate arrays (FPGAs) or application-specific ICs (ASICs)), hard disk drives (HDDs), hybrid hard drives (HHDs), optical discs, optical disc drives (ODDs), magneto-optical discs, magneto-optical drives, floppy diskettes, floppy disk drives (FDDs), magnetic tapes, solid-state drives (SSDs), RAM-drives, SECURE DIGITAL cards or drives, any other suitable computer-readable non-transitory storage media, or any suitable combination of two or more of these, where appropriate. A computer-readable non-transitory storage medium may be volatile, non-volatile, or a combination of volatile and non-volatile, where appropriate.

[0040] The specification also provides a computer system. FIG. 8 illustrates an example computer system **800**. In particular embodiments, one or more computer systems **800** perform one or more steps of one or more methods described or illustrated herein. In particular embodiments, one or more computer systems **800** provide functionality described or illustrated herein. In particular embodiments, software running on one or more computer systems **800** performs one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Particular embodiments include one or more portions of one or more computer systems **800**. Herein, reference to a computer system may encompass a computing device, and vice versa, where appropriate. Moreover, reference to a computer system may encompass one or more computer systems, where appropriate.

[0041] This disclosure contemplates any suitable number of computer systems **800**. This disclosure contemplates computer system **800** taking any suitable physical form. As example and not by way of limitation, computer system **800** may be an embedded computer system, a system-on-chip (SOC), a single-board computer system (SBC) (such as, for example, a computer-on-module (COM) or system-on-module (SOM)), a desktop computer system, a laptop or notebook computer system, an interactive kiosk, a mainframe, a mesh of computer systems, a mobile

telephone, a personal digital assistant (PDA), a server, a tablet computer system, or a combination of two or more of these. Where appropriate, computer system **800** may include one or more computer systems **800**; be unitary or distributed; span multiple locations; span multiple machines; span multiple data centers; or reside in a cloud, which may include one or more cloud components in one or more networks. Where appropriate, one or more computer systems **800** may perform without substantial spatial or temporal limitation one or more steps of one or more methods described or illustrated herein. As an example and not by way of limitation, one or more computer systems **800** may perform in real time or in batch mode one or more steps of one or more methods described or illustrated herein. One or more computer systems **800** may perform at different times or at different locations one or more steps of one or more methods described or illustrated herein, where appropriate.

[0042] In particular embodiments, computer system **800** includes a processor **802**, memory **804**, storage **806**, an input/output (I/O) interface **808**, a communication interface **810**, and a bus **812**. Although this disclosure describes and illustrates a particular computer system having a particular number of particular components in a particular arrangement, this disclosure contemplates any suitable computer system having any suitable number of any suitable components in any suitable arrangement.

[0043] In particular embodiments, processor **802** includes hardware for executing instructions, such as those making up a computer program. As an example and not by way of limitation, to execute instructions, processor **802** may retrieve (or fetch) the instructions from an internal register, an internal cache, memory **804**, or storage **806**; decode and execute them; and then write one or more results to an internal register, an internal cache, memory **804**, or storage **806**. In particular embodiments, processor **802** may include one or more internal caches for data, instructions, or addresses. This disclosure contemplates processor **802** including any suitable number of any suitable internal caches, where appropriate. As an example and not by way of limitation, processor **802** may include one or more instruction caches, one or more data caches, and one or more translation lookaside buffers (TLB s). Instructions in the instruction caches may be copies of instructions in memory **804** or storage **806**, and the instruction caches may speed up retrieval of those instructions by processor **802**. Data in the data caches may be copies of data of **80** in memory **804** or storage **806** for instructions executing at processor **802** to operate on; the results of previous instructions executed at processor **802** for access by subsequent instructions executing at processor

802 or for writing to memory **804** or storage **806**; or other suitable data. The data caches may speed up read or write operations by processor **802**. The TLBs may speed up virtual-address translation for processor **802**. In particular embodiments, processor **802** may include one or more internal registers for data, instructions, or addresses. This disclosure contemplates processor **802** including any suitable number of any suitable internal registers, where appropriate. Where appropriate, processor **802** may include one or more arithmetic logic units (ALUs); be a multi-core processor; or include one or more processors **802**. Although this disclosure describes and illustrates a particular processor, this disclosure contemplates any suitable processor.

[0044] In particular embodiments, memory **804** includes main memory for storing instructions for processor **802** to execute or data for processor **802** to operate on. As an example and not by way of limitation, computer system **800** may load instructions from storage **806** or another source (such as, for example, another computer system **800**) to memory **804**. Processor **802** may then load the instructions from memory **804** to an internal register or internal cache. To execute the instructions, processor **802** may retrieve the instructions from the internal register or internal cache and decode them. During or after execution of the instructions, processor **802** may write one or more results (which may be intermediate or final results) to the internal register or internal cache. Processor **802** may then write one or more of those results to memory **804**. In particular embodiments, processor **802** executes only instructions in one or more internal registers or internal caches or in memory **804** (as opposed to storage **806** or elsewhere) and operates only on data in one or more internal registers or internal caches or in memory **804** (as opposed to storage **806** or elsewhere). One or more memory buses (which may each include an address bus and a data bus) may couple processor **802** to memory **804**. Bus **812** may include one or more memory buses, as described below. In particular embodiments, one or more memory management units (MMUs) reside between processor **802** and memory **804** and facilitate accesses to memory **804** requested by processor **802**. In particular embodiments, memory **804** includes random access memory (RAM). This RAM may be volatile memory, where appropriate. Where appropriate, this RAM may be dynamic RAM (DRAM) or static RAM (SRAM). Moreover, where appropriate, this RAM may be single-ported or multi-ported RAM. This disclosure contemplates any suitable RAM. Memory **804** may include one or more memories **804**, where appropriate. Although this disclosure describes and illustrates particular memory, this disclosure contemplates any suitable memory.

[0045] In particular embodiments, storage **806** includes mass storage for data or instructions.

As an example and not by way of limitation, storage **806** may include a hard disk drive (HDD), a floppy disk drive, flash memory, an optical disc, a magneto-optical disc, magnetic tape, or a Universal Serial Bus (USB) drive or a combination of two or more of these. Storage **806** may include removable or non-removable (or fixed) media, where appropriate. Storage **806** may be internal or external to computer system **800**, where appropriate. In particular embodiments, storage **806** is non-volatile, solid-state memory. In particular embodiments, storage **806** includes read-only memory (ROM). Where appropriate, this ROM may be mask-programmed ROM, programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (EEPROM), electrically alterable ROM (EAROM), or flash memory or a combination of two or more of these. This disclosure contemplates mass storage **806** taking any suitable physical form. Storage **806** may include one or more storage control units facilitating communication between processor **802** and storage **806**, where appropriate. Where appropriate, storage **806** may include one or more storages **806**. Although this disclosure describes and illustrates particular storage, this disclosure contemplates any suitable storage.

[0046] In particular embodiments, I/O interface **808** includes hardware, software, or both, providing one or more interfaces for communication between computer system **800** and one or more I/O devices. Computer system **800** may include one or more of these I/O devices, where appropriate. One or more of these I/O devices may enable communication between a person and computer system **800**. As an example and not by way of limitation, an I/O device may include a keyboard, keypad, microphone, monitor, mouse, printer, scanner, speaker, still camera, stylus, tablet, touch screen, trackball, video camera, another suitable I/O device or a combination of two or more of these. An I/O device may include one or more sensors. This disclosure contemplates any suitable I/O devices and any suitable I/O interfaces **808** for them. Where appropriate, I/O interface **808** may include one or more device or software drivers enabling processor **802** to drive one or more of these I/O devices. I/O interface **808** may include one or more I/O interfaces **808**, where appropriate. Although this disclosure describes and illustrates a particular I/O interface, this disclosure contemplates any suitable I/O interface.

[0047] In particular embodiments, communication interface **810** includes hardware, software, or both providing one or more interfaces for communication (such as, for example, packet-based communication) between computer system **800** and one or more other computer systems **800** or one or more networks. As an example and not by way of limitation, communication interface **810**

may include a network interface controller (NIC) or network adapter for communicating with an Ethernet or other wire-based network or a wireless NIC (WNIC) or wireless adapter for communicating with a wireless network, such as a WI-FI network. This disclosure contemplates any suitable network and any suitable communication interface **810** for it. As an example and not by way of limitation, computer system **800** may communicate with an ad hoc network, a personal area network (PAN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), or one or more portions of the Internet or a combination of two or more of these. One or more portions of one or more of these networks may be wired or wireless. As an example, computer system **800** may communicate with a wireless PAN (WPAN) (such as, for example, a BLUETOOTH WPAN), a WI-FI network, a WI-MAX network, a cellular telephone network (such as, for example, a Global System for Mobile Communications (GSM) network), or other suitable wireless network or a combination of two or more of these. Computer system **800** may include any suitable communication interface **810** for any of these networks, where appropriate. Communication interface **810** may include one or more communication interfaces **810**, where appropriate. Although this disclosure describes and illustrates a particular communication interface, this disclosure contemplates any suitable communication interface.

[0048] In particular embodiments, bus **812** includes hardware, software, or both coupling components of computer system **800** to each other. As an example and not by way of limitation, bus **812** may include an Accelerated Graphics Port (AGP) or other graphics bus, an Enhanced Industry Standard Architecture (EISA) bus, a front-side bus (FSB), a HYPERTRANSPORT (HT) interconnect, an Industry Standard Architecture (ISA) bus, an INFINIBAND interconnect, a low-pin-count (LPC) bus, a memory bus, a Micro Channel Architecture (MCA) bus, a Peripheral Component Interconnect (PCI) bus, a PCI-Express (PCIe) bus, a serial advanced technology attachment (SATA) bus, a Video Electronics Standards Association local (VLB) bus, or another suitable bus or a combination of two or more of these. Bus **812** may include one or more buses **812**, where appropriate. Although this disclosure describes and illustrates a particular bus, this disclosure contemplates any suitable bus or interconnect.

[0049] It should be further noted that the term "include", "comprise" or any other variant thereof is intended to cover non-exclusive inclusion, so that a process, method, commodity, or device that includes a series of elements includes not only those elements but also other elements that are not explicitly listed, or includes elements inherent in such a process, method, commodity,

or device. In the absence of more restrictions, an element defined by the statement "includes a..." does not rule out a case that a process, method, commodity, or device including the element has another same element.

[0050] The embodiments in the present specification are described in a progressive way. For same or similar parts of the embodiments, references can be made to the embodiments mutually. Each embodiment focuses on a difference from other embodiments. Particularly, the system embodiments are briefly described since they are basically similar to the method embodiments. For related parts, references can be made to related descriptions in the method embodiments.

[0051] The foregoing descriptions are merely embodiments of this specification, and are not intended to limit this specification. A person skilled in the art may make various modifications or changes in this specification. Any modification, equivalent replacement, or improvement made without departing from the spirit and principle of this specification shall fall within the protection scope of the claims of this specification.

CLAIMS

What is claimed is:

1. An application programming interface access method, wherein the method comprises:

receiving a first request sent by a target user for a target application programming interface, wherein the first request comprises a target user field; and determining a target factor field corresponding to the target user field based on a pre-stored field correspondence, wherein the field correspondence indicates a correspondence between a factor field provided in the target application programming interface and a user field that matches the target user; and

obtaining field content of the target factor field from a serving end corresponding to the target application programming interface based on the target factor field, and sending the field content of the target factor field to the target user as field content of the target user field.

2. The method according to claim 1, wherein the obtaining field content of the target factor field from a serving end corresponding to the target application programming interface based on the target factor field comprises:

sending a second request to the serving end corresponding to the target application programming interface, wherein the second request comprises the target factor field; and

receiving the field content of the target factor field that is sent by the serving end.

3. The method according to claim 1, wherein the pre-stored correspondence between the factor field provided in the target application programming interface and the user field that matches the target user comprises: a correspondence between some factor fields of the target application programming interface and user fields that match the target user, and the some factor fields are determined by the target user from all factor fields of the target application programming interface; and

the some factor fields comprise the target user field.

4. The method according to claim 3, wherein all the factor fields of the target application programming interface comprise mandatory factor fields and optional factor fields; and the some factor fields comprise the mandatory factor fields and all or a part of the optional factor fields.

5. The method according to claim 1, wherein the correspondence between the factor field provided in the target application programming interface and the user field that matches the target user is pre-stored in the following manner:

obtaining a user field that is input by the target user and corresponding to a factor field provided in the target application programming interface; and

storing a correspondence between the factor field and the user field.

6. The method according to claim 1, wherein the target user comprises an individual user or a user belonging to a target group.

7. The method according to claim 1, executed by an application programming interface gateway.

8. The method according to claim 7, wherein the application programming interface gateway is configured to receive requests for a plurality of application programming interfaces, and the plurality of application programming interfaces comprise the target application programming interface.

9. A computer system, comprising a memory, a processor, and a computer program stored in the memory and running on the processor, wherein when executing the program, the processor implements the following steps:

receiving a first request sent by a target user for a target application programming interface, wherein the first request comprises a target user field; and determining a target factor field corresponding to the target user field based on a pre-stored field correspondence, wherein the field correspondence indicates a correspondence between a factor field provided in the target application programming interface and a user field that matches the target user; and

obtaining field content of the target factor field from a serving end corresponding to the target application programming interface based on the target factor field, and sending the field content of the target factor field to the target user as field content of the target user field.

10. The computer system according to claim 9, wherein the obtaining field content of the target factor field from a serving end corresponding to the target application programming interface based on the target factor field comprises:

sending a second request to the serving end corresponding to the target application programming interface, wherein the second request comprises the target factor field; and receiving the field content of the target factor field that is sent by the serving end.

11. The computer system according to claim 9, wherein the pre-stored correspondence between the factor field provided in the target application programming interface and the user field that matches the target user comprises: a correspondence between some factor fields of the target

application programming interface and user fields that match the target user, and the some factor fields are determined by the target user from all factor fields of the target application programming interface; and

the some factor fields comprise the target user field.

12. The computer system according to claim 11, wherein all the factor fields of the target application programming interface comprise mandatory factor fields and optional factor fields; and the some factor fields comprise the mandatory factor fields and all or a part of the optional factor fields.

13. The computer system according to claim 9, wherein the correspondence between the factor field provided in the target application programming interface and the user field that matches the target user is pre-stored in the following manner:

obtaining a user field that is input by the target user and corresponding to a factor field provided in the target application programming interface; and

storing a correspondence between the factor field and the user field.

14. The computer system according to claim 9, wherein the target user comprises a user belonging to a target group.

15. The computer system according to claim 9, used as an application programming interface gateway.

16. The computer system according to claim 15, wherein the application programming interface gateway is configured to receive requests for a plurality of application programming interfaces, and the plurality of application programming interfaces comprise the target application programming interface.

17. An application programming interface access apparatus, wherein the apparatus comprises:

a receiving unit, configured to: receive a first request sent by a target user for a target application programming interface, wherein the first request comprises a target user field; and determine a target factor field corresponding to the target user field based on a pre-stored field correspondence, wherein the field correspondence indicates a correspondence between a factor field provided in the target application programming interface and a user field that matches the target user; and

a sending unit, configured to: obtain field content of the target factor field from a serving end corresponding to the target application programming interface based on the target factor field, and

send the field content of the target factor field to the target user as field content of the target user field.

18. A computer-readable nonvolatile storage medium, wherein the storage medium stores a computer program, and when the computer program is performed by a processor, the method according to any one of claims 1 to 8 is implemented.