

SERVICE PROCESSING METHOD, APPARATUS, AND DEVICE FOR SMARTWATCH

TECHNICAL FIELD

[0001] This specification relates to the field of internet of things technologies, and in particular, to a service processing method, apparatus, and device for a smartwatch.

BACKGROUND

[0002] With the widespread use of smartphones, people increasingly use the smartphones to make electronic payments. In recent years, smartwatches are also gradually emerging, with a growing user base. Compared with the smartphones, the smartwatches provide more convenience in some payment scenarios. Based on this, some smartwatches technically draw on the experience and achievements of mobile phone payments. By integrating two-dimensional code payment and transit code functions into smartwatches, users can easily and conveniently complete some payment operations in life without taking out their smartphones, but by simply raising their wrists and using their smartwatches.

[0003] Currently, communication manners of smartwatches on the market include Bluetooth, Wi-Fi, eSIM, etc. Many smartwatches are based on a real-time operating system (RTOS), and include only a Bluetooth communication module. This severely limits payment experience of the smartwatches. Typically, for example, after payments are made, users can only briefly understand, on the smartwatches, that a payment operation is completed. Due to the lack of means to communicate with a serving end, detailed payment result information of a payment serving end cannot be directly obtained on the smartwatches.

[0004] Based on this, a solution that can improve a communication capability of such a smartwatch that includes only a Bluetooth communication module to improve payment experience of the smartwatch is required.

SUMMARY

[0005] One or more embodiments of this specification provide a service processing method, apparatus, and device for a smartwatch and a storage medium, to resolve the following technical

problem: A solution that can improve a communication capability of such a smartwatch that includes only a Bluetooth communication module to improve payment experience of the smartwatch is required.

[0006] To solve the above-mentioned technical problem, the one or more embodiments of this specification are implemented as follows:

[0007] One or more embodiments of this specification provide a service processing method for a smartwatch, applied to a watch end on the smartwatch. The method includes:

[0008] obtaining Socket communication trigger data;

[0009] performing encapsulation by using a short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet;

[0010] sending the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs encapsulation by using the short-range wireless communication protocol and based on Socket communication response data returned by the serving end, to obtain a second data packet; and

[0011] receiving the second data packet returned from the mobile phone end by performing short-range wireless communication with the smartphone, and parsing the second data packet to obtain the Socket communication response data.

[0012] One or more embodiments of this specification provide a service processing apparatus for a smartwatch, applied to a watch end on the smartwatch. The apparatus includes:

[0013] a trigger data obtaining module, configured to obtain Socket communication trigger data;

[0014] a trigger data encapsulation module, configured to perform encapsulation by using a short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet;

[0015] a data packet sending module, configured to send the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs encapsulation by using the short-range wireless communication protocol and based on Socket communication response data returned by the serving end, to obtain a second data packet; and

[0016] a response data obtaining module, configured to: receive the second data packet returned

from the mobile phone end by performing short-range wireless communication with the smartphone, and parse the second data packet to obtain the Socket communication response data.

[0017] One or more embodiments of this specification provide a service processing device for a smartwatch, applied to a watch end on the smartwatch. The device includes:

[0018] at least one processor; and

[0019] a memory communicatively connected to the at least one processor.

[0020] The memory stores instructions that can be executed by the at least one processor, and the instructions are executed by the at least one processor, to enable the at least one processor to perform the following operations:

[0021] obtaining Socket communication trigger data;

[0022] performing encapsulation by using a short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet;

[0023] sending the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs encapsulation by using the short-range wireless communication protocol and based on Socket communication response data returned by the serving end, to obtain a second data packet; and

[0024] receiving the second data packet returned from the mobile phone end by performing short-range wireless communication with the smartphone, and parsing the second data packet to obtain the Socket communication response data.

[0025] One or more embodiments of this specification provide a nonvolatile computer storage medium, applied to a watch end on a smartwatch. The medium stores computer-executable instructions, and the computer-executable instructions are set to perform the following operations:

[0026] obtaining Socket communication trigger data;

[0027] performing encapsulation by using a short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet;

[0028] sending the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs encapsulation by using the short-range wireless communication protocol and based on Socket

communication response data returned by the serving end, to obtain a second data packet; and

[0029] receiving the second data packet returned from the mobile phone end by performing short-range wireless communication with the smartphone, and parsing the second data packet to obtain the Socket communication response data.

[0030] The above-mentioned at least one technical solution used in one or more embodiments of this specification can achieve the following beneficial effects: A Socket communication solution is used as a basic internet access communication protocol of a smartwatch, a smartphone near the smartwatch is used as a proxy, and a capability of short-range wireless communication (Bluetooth communication is used as an example) between a watch end and a mobile phone end and a capability of remote communication between the mobile phone end and a serving end are used. Based on cooperation of the three parties, data related to Socket communication between the watch end and the serving end is encapsulated, parsed, and intermediately transmitted, to indirectly implement Socket communication between the watch end on the smartwatch and the serving end, so as to improve a communication capability of the smartwatch. In this way, service data can be flexibly exchanged between the watch end and the serving end, to help improve payment experience of the smartwatch that includes only a Bluetooth communication module.

BRIEF DESCRIPTION OF DRAWINGS

[0031] To describe the technical solutions in the embodiments of this specification or in the conventional technology more clearly, the following briefly describes the accompanying drawings required for describing the embodiments or the conventional technology. Clearly, the accompanying drawings in the following descriptions merely show some embodiments of this specification, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0032] FIG. 1 is a schematic flowchart of a service processing method for a smartwatch according to one or more embodiments of this specification;

[0033] FIG. 2 is a schematic diagram of a proxy link for completing Socket communication between a watch end and a serving end by using a mobile phone end as a proxy in a scenario according to one or more embodiments of this specification;

[0034] FIG. 3 is a schematic flowchart of completing proxy link processing of a single Socket API in a scenario according to one or more embodiments of this specification;

[0035] FIG. 4 is a schematic diagram of customizing a parameter related to a Socket() function according to one or more embodiments of this specification;

[0036] FIG. 5 is a schematic flowchart of an additional task processing solution of a mobile phone end involved in a watch proxy link in an application scenario according to one or more embodiments of this specification;

[0037] FIG. 6 is a schematic flowchart of a dynamic selection solution for a proxy mobile phone end in an application scenario according to one or more embodiments of this specification;

[0038] FIG. 7 is a schematic structural diagram of a service processing apparatus for a smartwatch according to one or more embodiments of this specification; and

[0039] FIG. 8 is a schematic structural diagram of a service processing device for a smartwatch according to one or more embodiments of this specification.

DESCRIPTION OF EMBODIMENTS

[0040] Embodiments of this specification provide a service processing method, apparatus, and device for a smartwatch, and a storage medium.

[0041] To make a person skilled in the art better understand the technical solutions in this specification, the following clearly and comprehensively describes the technical solutions in the embodiments of this specification with reference to the accompanying drawings in the embodiments of this specification. Clearly, the described embodiments are merely some but not all of the embodiments of this specification. All other embodiments obtained by a person of ordinary skill in the art based on the embodiments of this specification without creative efforts shall fall within the protection scope of this specification.

[0042] To resolve the problem in the background and improve a communication capability of a smartwatch, the applicant makes some attempts, for example, maintains communication by keeping a sports application of a watch manufacturer alive, or encapsulates HTTP protocol proxy communication at a mobile phone end. However, there are still some problems in these attempts. For example, the former attempt listed is still not flexible enough for a watch end, only messages can be received, and there is no capability of actively obtaining information. In addition, the sports application needs to continuously keep alive to maintain real-time information transmission, resulting in slightly higher resource costs. In the latter attempt listed, HTTP message packets are in a uniform and fixed format and the packet is cumbersome, which is not friendly to Bluetooth communication. In addition, HTTP protocol communication does not have a bidirectional communication function.

[0043] To resolve these problems, the applicant proposes a more optimized solution in this specification, to enable a smartwatch to complete internet access communication by using a mobile phone as a proxy, and selects Socket communication as a basic internet access communication

protocol of the smartwatch.

[0044] Compared with the previous attempt HTTP communication, Socket communication is a lower-level, connection-oriented transport layer communication manner in which a connection is usually established by using a TCP protocol (this manner is mainly used as an example for description) or a connectionless UDP protocol can be used. In Socket communication, no specific data format is stipulated, and a data format and structure can be freely defined. Therefore, two communication parties can independently negotiate and parse data. An overall protocol is simple and encapsulation content is simple. Therefore, the packet is relatively friendly to Bluetooth communication. In addition, in Socket communication, data transmission can be directly performed after a connection is established between the two parties, and bidirectional communication is supported. When a connection is established, information can be actively pushed, without the need for a client to send a request to a server each time.

[0045] However, most RTOS smartwatches cannot directly implement a network connection. Therefore, the RTOS smartwatches encapsulate parameters for Socket communication by using existing Bluetooth-related protocols of the RTOS smartwatches, and then forward the parameters to mobile phone ends for parsing, to complete proxy internet access.

[0046] Based on such an overall idea, the following continues to describe the solutions of this specification in detail.

[0047] FIG. 1 is a schematic flowchart of a service processing method for a smartwatch according to one or more embodiments of this specification. The procedure can be applied to a watch end on the smartwatch, and has more prominent advantages particularly for a smartwatch that supports only a short-range wireless communication manner such as Bluetooth communication but does not support a network remote communication manner. The procedure is performed by the watch end, and further involves a mobile phone end on a smartphone and a serving end on a server. The watch end and the mobile phone end can be clients of corresponding applications, or embedded application programs (for example, applets) embedded in other application clients. Particularly for the watch end, the watch end can be lighter (with simplified functions, etc.) than the mobile phone end.

[0048] The procedure in FIG. 1 includes the following steps.

[0049] S102: Obtain Socket communication trigger data.

[0050] In one or more embodiments of this specification, the Socket communication trigger data can include data that is used to indirectly perform Socket communication between the watch end and the serving end by using the mobile phone end as a proxy, and is mainly a parameter for Socket communication, etc. The Socket communication trigger data can be specifically used to trigger establishment of a Socket communication connection, or is used to trigger an active request for data

at the serving end based on a connection after the connection is established, and in a case of no connection, can be used to trigger active sending of data to the serving end.

[0051] In one or more embodiments of this specification, an example in which the Socket communication trigger data is used to trigger establishment of a connection is used. A Socket API parameter structure can be predesigned. Then, a Socket API parameter is encapsulated by using the structure, to obtain a Socket API structure object; and serialization processing is performed on the Socket API structure object, to obtain the Socket communication trigger data. The Socket API is mainly represented as one or more functions, for example, Socket(), Connect(), Send(), and Receive().

[0052] In different specific uses, for functions currently involved, appropriate Socket API parameters can be flexibly customized and values can be assigned accordingly.

[0053] S104: Perform encapsulation by using a short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet.

[0054] In one or more embodiments of this specification, the smartwatch does not support direct network remote communication, and supports only one or more types of short-range wireless communication. Typically, only Bluetooth communication is supported, as mentioned in the background. In this case, the short-range wireless communication protocol supported by the smartwatch can be a Bluetooth protocol or a Bluetooth low energy (BLE) protocol, and preferably, is the BLE protocol, which is more friendly to the solutions of this specification, and helps ensure battery life extension of the smartwatch.

[0055] The smartwatch uses a nearby smartphone (for example, a smartphone held by a user wearing the smartwatch) as a proxy, and needs to transmit the Socket communication trigger data to the smartphone. The smartwatch can perform short-range wireless communication with the smartphone. Therefore, the Socket communication trigger data is correspondingly encapsulated to meet a requirement of short-range wireless communication data transmission. For example, the watch end can encapsulate the Socket communication trigger data into packets by using the BLE protocol, to obtain a plurality of data packets. For ease of distinction, such data packets are referred to as first data packets.

[0056] S106: Send the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs encapsulation by using the short-range wireless communication protocol and based on Socket communication response data returned by the serving end, to obtain a second data packet.

[0057] In one or more embodiments of this specification, the watch end and the mobile phone end can belong to a same application, or can belong to different applications. In the latter case, authorization between applications can be performed in advance, so that normal interaction can be performed.

[0058] By using the short-range wireless communication protocol, the mobile phone end can at least completely obtain, by parsing the first data packet, the Socket communication trigger data (referred to as raw data for distinction) previously encapsulated by the watch end, and the mobile phone end can directly perform Socket communication with the corresponding serving end based on the raw data. Alternatively, the mobile phone end can perform more complex processing on the raw data based on a requirement, to obtain distinct new data, use the new data as the obtained Socket communication trigger data, and perform Socket communication with the corresponding serving end based on the Socket communication trigger data.

[0059] In one or more embodiments of this specification, the watch end and the corresponding serving end can belong to a same application or different applications in a cooperative relationship.

[0060] In the payment scenario mentioned in the background, the watch end and the serving end can belong to payment applications or payment channel applications. The payment application is, for example, a third-party payment application, and the payment channel application is, for example, an application of a bank with which a third-party payment application cooperates. When a third-party payment is made, deduction can be performed from a bank card issued by the bank, to complete the payment. For example, the watch end and the serving end belong to a same third-party payment application; or the watch end belongs to a third-party payment application, and the serving end belongs to an application of a bank that cooperates with the third-party payment application; and so on.

[0061] In one or more embodiments of this specification, how the serving end responds depends on a current use of the Socket communication trigger data. If the mobile phone end and the serving end do not respond to establish a connection, a connection can be established based on the Socket communication trigger data. If a connection is established, the serving end can perform corresponding service processing based on a service involved in the Socket communication trigger data, to obtain a service processing result, and return the service processing result to the mobile phone end through the connection in a form of Socket communication response data.

[0062] S108: Receive the second data packet returned from the mobile phone end by performing short-range wireless communication with the smartphone, and parse the second data packet to obtain the Socket communication response data.

[0063] In one or more embodiments of this specification, after obtaining the Socket

communication response data returned by the serving end, the mobile phone end can similarly completely encapsulate the Socket communication response data, and further return the Socket communication response data to the watch end, or can perform more complex processing, encapsulate data that is not exactly the same as the raw response data, and return the data to the watch end. Certainly, either manner can be used provided that it is ensured that a current service requirement of the watch end is met based on the Socket communication response data obtained by the watch end by parsing the second data packet. The latter more complex processing manner can further help the mobile phone end to additionally meet a service requirement of the mobile phone end. An example is provided below for description.

[0064] Based on such a short-range communication process and a Socket communication process indirectly completed based on a proxy, in a payment scenario, even for a smartwatch that includes only a Bluetooth communication module, a watch end on the smartwatch can conveniently obtain detailed payment result information from a payment serving end in a timely manner, and display the detailed payment result information to a user.

[0065] According to the method in FIG. 1, a Socket communication solution is used as a basic internet access communication protocol of a smartwatch, a smartphone near the smartwatch is used as a proxy, and a capability of short-range wireless communication (Bluetooth communication is used as an example) between a watch end and a mobile phone end and a capability of remote communication between the mobile phone end and a serving end are used. Based on cooperation of the three parties, data related to Socket communication between the watch end and the serving end is encapsulated, parsed, and intermediately transmitted, to indirectly implement Socket communication between the watch end on the smartwatch and the serving end, so as to improve a communication capability of the smartwatch. In this way, service data can be flexibly exchanged between the watch end and the serving end, to help improve payment experience of the smartwatch that includes only a Bluetooth communication module.

[0066] Based on the method in FIG. 1, this specification further provides some specific implementation solutions and extension solutions of the method, which are further described below.

[0067] In one or more embodiments of this specification, in the above-mentioned example manner of implementing active triggering and returning a response, the watch end can actively obtain information from the serving end. Similarly, the watch end can actively send information to the serving end in this manner. Certainly, bidirectional communication can also be implemented based on Socket communication, and it is not necessary for the watch end to actively initiate interaction.

[0068] A specified connection used for the above-mentioned Socket communication performed with the serving end by using the mobile phone end as a proxy identity is referred to as a watch proxy

connection. It can be learned from the above-mentioned descriptions that the watch proxy connection is triggered to be established based on short-range wireless communication between the watch end and the mobile phone end. After the watch proxy connection is established, the mobile phone end may not need to actively request the serving end, but passively receive, by performing short-range wireless communication with the smartphone, information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection.

[0069] For example, when a watch payment is made through the watch end, the watch payment can be specifically completed through two-dimensional code scanning interaction or short-range wireless communication between the smartphone and a target checkout device. After deduction is successfully performed, the serving end actively pushes detailed payment result information corresponding to the watch payment through the watch proxy connection, and the mobile phone end on the smartphone sends the detailed payment result information to the watch end through short-range wireless communication. The detailed payment result information can include information such as an actual payment amount and/or an account balance after the payment, so that the user more clearly understands a change involved in the payment.

[0070] It can be learned from the above-mentioned descriptions that this specification provides a proxy link for completing Socket communication between the watch end and the serving end by using the mobile phone end as a proxy. For a more intuitive view, refer to FIG. 2. FIG. 2 is a schematic diagram of the proxy link in a scenario according to one or more embodiments of this specification. In this scenario, the serving end is specifically a payment serving end. The smartwatch can include only a Bluetooth communication module, and the watch end on the smartwatch is referred to as a Bluetooth smartwatch end.

[0071] In the link shown in FIG. 2, communication is performed between the Bluetooth smartwatch end and the mobile phone end by using, for example, the BLE protocol. In this case, the mobile phone end is used as a proxy of the Bluetooth smartwatch end to perform, for example, TCP protocol-based Socket communication, with the serving end. This can intuitively reflect a reference logic link design idea for transitioning from a network protocol (for example, a TCP/IP protocol) to a Bluetooth-related protocol (for example, a Bluetooth protocol/BLE protocol).

[0072] In the link, protocol layer locations and uses of functions involved in Socket communication at each end are shown. In the link, a spatial dimension shows flow of data between the three ends based on different protocols, and a time dimension shows establishment of a Socket communication connection. Further, data exchange can continue to be performed based on the connection by sending a request and returning a response, and the connection can be closed as required. For example, for the Bluetooth smartwatch end and the mobile phone end, functions such

as Socket() and Connect() can be involved in a phase of establishing the Socket communication connection, and functions such as Send(), Receive(), and Close() can be involved in a subsequent phase; and for the payment serving end, functions such as Socket(), Bind(), Listen(), and Accept() can be involved in the phase of establishing the Socket communication connection, and functions such as Receive(), Send(), and Close() can be involved in the subsequent phase.

[0073] Further, one or more embodiments of this specification further provide a schematic flowchart of completing proxy link processing of a single Socket API in a scenario, as shown in FIG. 3. This can be understood with reference to the above-mentioned descriptions and the following example, and some content is not described.

[0074] The procedure in FIG. 3 mainly includes processes such as serialization and deserialization of a Socket parameter, encapsulation into packets based on the BLE protocol, and packet assembly. Intuitively, a specific customized socket() function is used as an example for description. FIG. 4 shows a definition of a parameter related to the socket() function.

[0075] In FIG. 4, three parameter fields used for the socket() function are defined, including a 4-byte "domain" (representing a protocol domain) field, an 8-byte "type" (representing a socket type) field, and a 4-byte "protocol" (representing a transmission protocol) field. In addition, a data structure "RetSocket" is defined to return response data to the watch end.

[0076] Based on these definitions, for example, the socket() function used in the procedure in FIG. 3 is defined as follows:

[0077] "int socket (int domain, int type, int protocol)".

[0078] Specifically, the procedure mainly includes the following steps.

[0079] The socket() function is used as an example. In the first step in establishing a communication connection, the watch end used as a client needs to send basic information such as the protocol domain, the socket type, and the transmission protocol to the serving end. The serving end creates a socket object based on the parameter information provided by the watch end, to establish a communication infrastructure for both parties. Therefore, the parameter information of the socket() function needs to be correctly transmitted across platforms to the mobile phone end, and the mobile phone end initiates a connection to the serving end on behalf of the watch end. The mobile phone end transmits an invocation result of the socket() function across platforms to the watch end, to complete a complete data transmission process.

[0080] Serialization protocols that can be used include Protocol Buffers, JSON, Thrift, etc. The Protocol Buffers protocol is used as an example. The Protocol Buffers protocol is an efficient binary serialization protocol, provides a convenient manner to serialize and deserialize structured data, and is designed to enable data to be transmitted between different platforms and languages without errors.

By using this protocol, a complex data structure can be converted into a compact and efficient binary format, to implement efficient data transmission and storage.

[0081] Based on this, an example serialization and deserialization process is as follows:

[0082] 1. The following steps need to be completed at the watch end: First, a structure and a field Socket of a message need to be defined, and compilation is performed to generate code in a target language. Then, a field value of the message can be set by using a generated class or structure. Once the field value of the message is set, a serialization method can be invoked to convert the message data into a binary format. In this way, the data can be transmitted and stored in a binary form between different platforms.

[0083] 2. By using the BLE protocol at the watch end, the serialized binary data is split into packets and sent to the mobile phone end. After performing packet assembly, the mobile phone end restores the binary data to a message object by using a deserialization method. In a deserialization process, the binary data is converted into a message object with a corresponding field value based on a message structure Socket and a field sequence defined in advance at the watch end. Then, the mobile phone end invokes the socket() function on behalf of the watch end, and sends a request to the serving end.

[0084] 3. After receiving the socket request sent by the watch end, the serving end performs a series of processing, and returns a result to the mobile phone end. After receiving the returned result, the mobile phone end sets the result into the defined data structure RetSocket; and then serializes the data, splits the data into packets by using the BLE protocol, and then sends the data to the watch end.

[0085] 4. The watch end performs parsing based on the defined data structure RetSocket, to obtain the result. In this case, the watch end completes the first step of Socket communication, and therefore subsequent connection establishment can be performed.

[0086] Other functions involved below can also be processed in a similar manner.

[0087] According to the solutions in FIG. 2 and FIG. 3, a BLE internet access specification of the Bluetooth watch end can be established, and a procedure system can be improved, to improve user experience. By using the Socket communication solution as a basic internet access communication protocol of the smartwatch, the two communication parties can independently negotiate and parse data. An overall protocol is simple and encapsulation content is simple. In addition, in Socket communication, data transmission can be directly performed after a connection is established between the two parties. When a connection is established, information can be actively pushed, without the need for the watch end to send a request to the serving end each time. Therefore, the problems in the background and the attempt solutions mentioned above can be resolved.

[0088] In one or more embodiments of this specification, additional processing resources need to

be consumed when the mobile phone end is used as a proxy. To fully use these processing resources, the mobile phone end uses a communication requirement between the mobile phone end and the serving end to complete proxy work. In this case, the mobile phone end gains more active control, and avoids additional burden on the mobile phone end. In addition, the watch end has more choices for the mobile phone end on the smartphone, which helps optimize global resource configuration.

[0089] Based on this idea, it is considered that the mobile phone end should not merely forward data from the other two ends in a proxy process, but performs the more complex processing mentioned above on the data, to meet an original communication requirement of the mobile phone end. Specifically, one or more embodiments of this specification provide a schematic flowchart of an additional task processing solution of a mobile phone end involved in a watch proxy link in an application scenario, as shown in FIG. 5.

[0090] The procedure in FIG. 5 includes the following steps.

[0091] S502: The watch end sends the first data packet indicating a first service task of the watch end to the mobile phone end on the smartphone.

[0092] The first service task is a service task that needs to be currently completed by the watch end, for example, establishing a connection to the serving end and transmitting corresponding service data (for example, payment service data) based on the connection.

[0093] S504: The mobile phone end parses the first data packet to obtain the Socket communication trigger data, and performs Socket communication with the corresponding serving end based on the Socket communication trigger data and data indicating a second service task of the mobile phone end.

[0094] In one or more embodiments of this specification, the second service task is a service task of the mobile phone end, can be unrelated to the watch end, and may not be in a direct relationship with the first service task. The first service task and the second service task can belong to different services (for example, the first service task can belong to a payment service, and the second service task can belong to an instant messaging service), and the watch end and the mobile phone end can even belong to different applications. In conclusion, even if the second service task is not executed, based on the proxy link described above, the first service task can be normally executed to complete the above-mentioned Socket communication.

[0095] In this case, a main objective of the mobile phone end is actually to complete the second service task of the mobile phone end, while being incidentally used as a proxy to help the watch end to complete the first service task of the watch end.

[0096] S506: The mobile phone end receives the Socket communication response data returned by the serving end, where the Socket communication response data includes first task result data

obtained by executing the first service task by the serving end and second task result data obtained by executing the second service task.

[0097] In one or more embodiments of this specification, the serving end is responsible for executing the two types of tasks, and it can be unnecessary to clearly determine specific ends that need the two types of tasks. Therefore, the task result data of the two types of tasks can be uniformly placed in the Socket communication response data, and returned to the mobile phone end by using one message (that is, the Socket communication response data).

[0098] S508: The mobile phone end returns the second data packet obtained through encapsulation after the second task result data is intercepted from the Socket communication response data to the watch end, so that the watch end obtains the remaining first task result data.

[0099] The mobile phone end mainly needs to obtain the second task result data in the Socket communication response data. In addition, this part of data can preferably not be exposed to the watch end, especially in a case in which the two ends do not belong to a same application. Similarly, the mobile phone end can ensure, based on a predetermined security policy, that although the second service task and/or the second task result data need/needs to be transmitted by the watch end, corresponding plaintext is not exposed to the watch end. Based on this, the mobile phone end can intercept the second task result data from the Socket communication response data, perform encapsulation based on the remaining data after the interception, to obtain the second data packet, and return the second data packet to the watch end, so that the watch end can obtain the first task result data without obtaining the second task result data.

[0100] Further, to respond to the watch end in a more timely manner, based on a dynamic task requirement status of different mobile phone ends on the smartphone, a mobile phone end that is currently more suitable for being used as a proxy can be dynamically selected from these mobile phone ends. Based on this, one or more embodiments of this specification further provide a schematic flowchart of a dynamic selection solution for a proxy mobile phone end in an application scenario, as shown in FIG. 6.

[0101] The procedure in FIG. 6 includes the following steps.

[0102] S602: The watch end receives asynchronous communication requirement information sent by the smartphone through short-range wireless communication, where the asynchronous communication requirement information reflects asynchronous communication currently being performed or predicted asynchronous communication to be performed between a corresponding mobile phone end on the smartphone and the serving end.

[0103] In one or more embodiments of this specification, at least one mobile phone end that is in a plurality of different mobile phone ends on the smartphone and that currently has a service task that

can be delayed for execution (which means that the service task can be delayed for execution at a time after a current moment) can be determined as a target mobile phone end. In this case, the service task that can be delayed for execution can be determined as the second service task. This can motivate the target mobile phone end to accept a proxy identity, and accelerate execution of the service task that can be delayed for execution.

[0104] S604: The watch end determines the target mobile phone end from the plurality of corresponding different mobile phone ends on the smartphone based on the received asynchronous communication requirement information.

[0105] In this procedure, an asynchronous communication status is used as an example, and a service task to be executed through asynchronous communication is a service task that can be delayed for execution. It should be noted that in another scenario, there may also be a service task that can be delayed for execution, for example, a timing service task for which an execution moment is not reached, or a polling service task, which can be used as a basis to determine the target mobile phone end in a similar manner.

[0106] S606: The watch end sends the first data packet to the target mobile phone end on the smartphone by performing short-range wireless communication with the smartphone.

[0107] S608: The target mobile phone end determines a service task, of the target mobile phone end, corresponding to the asynchronous communication requirement information as the second service task.

[0108] The target mobile phone end can use a service task that is of the target mobile phone end and that can be delayed for execution as the second service task.

[0109] According to the solutions in FIG. 5 and FIG. 6, cooperation value between the watch end and the mobile phone end is further improved, and from a global perspective, it helps improve task execution efficiency and resource utilization.

[0110] Based on the same idea, one or more embodiments of this specification further provide an apparatus and a device corresponding to the above-mentioned method, as shown in FIG. 7 and FIG. 8. The apparatuses and the devices can correspondingly perform the above-mentioned method and related optional solutions.

[0111] FIG. 7 is a schematic structural diagram of a service processing apparatus for a smartwatch according to one or more embodiments of this specification. The apparatus is applied to a watch end on the smartwatch, and the apparatus includes:

[0112] a trigger data obtaining module 702, configured to obtain Socket communication trigger data;

[0113] a trigger data encapsulation module 704, configured to perform encapsulation by using a

short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet;

[0114] a data packet sending module 706, configured to send the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs encapsulation by using the short-range wireless communication protocol and based on Socket communication response data returned by the serving end, to obtain a second data packet; and

[0115] a response data obtaining module 708, configured to: receive the second data packet returned from the mobile phone end by performing short-range wireless communication with the smartphone, and parse the second data packet to obtain the Socket communication response data.

[0116] Optionally, the trigger data obtaining module 702 is configured to: encapsulate a Socket API parameter based on a predesigned Socket API parameter structure, to obtain a Socket API structure object; and

[0117] perform serialization processing on the Socket API structure object, to obtain the Socket communication trigger data.

[0118] Optionally, the trigger data encapsulation module 704 is configured to encapsulate the Socket communication trigger data into packets to obtain a plurality of first data packets, so that after obtaining the plurality of first data packets, the mobile phone end correspondingly assembles and parses the packets to obtain the Socket communication trigger data.

[0119] Optionally, the response data obtaining module 708 is configured to: after a specified watch proxy connection between the mobile phone end and the serving end is established, receive, by performing short-range wireless communication with the smartphone, information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection, where

[0120] the watch proxy connection is triggered to be established based on short-range wireless communication between the watch end and the mobile phone end.

[0121] Optionally, the watch end and the serving end belong to payment applications or payment channel applications;

[0122] the apparatus further includes:

[0123] a watch end payment module 710, configured to: before the information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection is received, make a watch payment through the watch end, where the watch payment is completed

through two-dimensional code scanning interaction or short-range wireless communication between the smartphone and a target checkout device; and

[0124] the response data obtaining module 708 is configured to receive detailed payment result information that corresponds to the watch payment, that is sent by the mobile phone end, and that is actively pushed by the serving end through the watch proxy connection, where the detailed payment result information includes at least an actual payment amount and an account balance after the payment.

[0125] Optionally, the data packet sending module 706 is configured to send the first data packet indicating a first service task of the watch end to the mobile phone end on the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, and performs Socket communication with the corresponding serving end based on the Socket communication trigger data and data indicating a second service task of the mobile phone end, where

[0126] the second data packet is obtained through encapsulation after the mobile phone end intercepts, from the Socket communication response data, second task result data obtained by executing the second service task by the serving end, so that the watch end obtains remaining first task result data obtained by executing the first service task by the serving end.

[0127] Optionally, the watch end and the mobile phone end respectively belong to different applications, or the first service task and the second service task belong to different services.

[0128] Optionally, the data packet sending module 706 is configured to: determine at least one mobile phone end that is in a plurality of different mobile phone ends on the smartphone and that currently has a service task that can be delayed for execution as a target mobile phone end; and

[0129] send the first data packet to the target mobile phone end on the smartphone by performing short-range wireless communication with the smartphone, so that the target mobile phone end uses the service task that can be delayed for execution as the second service task.

[0130] Optionally, the data packet sending module 706 is configured to: receive asynchronous communication requirement information sent by the smartphone through short-range wireless communication, where the asynchronous communication requirement information reflects asynchronous communication currently being performed or predicted asynchronous communication to be performed between a corresponding mobile phone end on the smartphone and the serving end; and

[0131] determine the target mobile phone end from the plurality of corresponding different mobile phone ends on the smartphone based on the received asynchronous communication requirement information.

[0132] Optionally, the short-range wireless communication protocol is a Bluetooth protocol or a

BLE protocol.

[0133] FIG. 8 is a schematic structural diagram of a service processing device for a smartwatch according to one or more embodiments of this specification. The apparatus is applied to a watch end on the smartwatch, and the device includes:

[0134] at least one processor; and

[0135] a memory communicatively connected to the at least one processor.

[0136] The memory stores instructions that can be executed by the at least one processor, and the instructions are executed by the at least one processor, to enable the at least one processor to perform the following operations:

[0137] obtaining Socket communication trigger data;

[0138] performing encapsulation by using a short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet;

[0139] sending the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs encapsulation by using the short-range wireless communication protocol and based on Socket communication response data returned by the serving end, to obtain a second data packet; and

[0140] receiving the second data packet returned from the mobile phone end by performing short-range wireless communication with the smartphone, and parsing the second data packet to obtain the Socket communication response data.

[0141] Based on the same idea, one or more embodiments of this specification further provide a nonvolatile computer storage medium, applied to a watch end on a smartwatch. The medium stores computer-executable instructions, and the computer-executable instructions are set to perform the following operations:

[0142] obtaining Socket communication trigger data;

[0143] performing encapsulation by using a short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet;

[0144] sending the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs

encapsulation by using the short-range wireless communication protocol and based on Socket communication response data returned by the serving end, to obtain a second data packet; and

[0145] receiving the second data packet returned from the mobile phone end by performing short-range wireless communication with the smartphone, and parsing the second data packet to obtain the Socket communication response data.

[0146] In the 1990s, whether a technical improvement is a hardware improvement (for example, an improvement to a circuit structure such as a diode, a transistor, or a switch) or a software improvement (an improvement to a method procedure) can be clearly distinguished. However, as technologies develop, current improvements to many method procedures can be considered as direct improvements to hardware circuit structures. Almost all designers program an improved method procedure into a hardware circuit, to obtain a corresponding hardware circuit structure. Therefore, a method procedure can be improved by using a hardware entity module. For example, a programmable logic device (PLD) (for example, a field programmable gate array (FPGA)) is such an integrated circuit, and a logical function of the PLD is determined by a user through device programming. The designer independently performs programming to "integrate" a digital system onto a PLD, without requesting a chip manufacturer to design and manufacture a dedicated integrated circuit chip. In addition, currently, instead of manually manufacturing an integrated circuit chip, such programming is mostly implemented by using "logic compiler" software. The "logic compiler" software is similar to a software compiler used to develop and write a program. Original code needs to be written in a particular programming language before being compiled. The language is referred to as a hardware description language (HDL). There are many HDLs, such as the Advanced Boolean Expression Language (ABEL), the Altera Hardware Description Language (AHDL), Confluence, the Cornell University Programming Language (CUPL), HDCal, the Java Hardware Description Language (JHDL), Lava, Lola, MyHDL, PALASM, and the Ruby Hardware Description Language (RHDL). At present, the Very-High-Speed Integrated Circuit Hardware Description Language (VHDL) and Verilog are most commonly used. It should also be clear to a person skilled in the art that a hardware circuit for implementing a logical method procedure can be easily obtained by performing slight logic programming on the method procedure by using the above-mentioned several hardware description languages and programming the method procedure into an integrated circuit.

[0147] A controller can be implemented in any suitable manner. For example, the controller can be in a form such as a microprocessor, a processor, or a computer-readable medium, a logic gate, a switch, an application-specific integrated circuit (ASIC), a programmable logic controller, or an embedded microcontroller storing computer-readable program code (such as software or firmware) that can be executed by the (micro)processor. Examples of the controller include but are not limited

to the following microcontrollers: ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20, and Silicone Labs C8051F320. A memory controller can also be implemented as a part of control logic of the memory. A person skilled in the art also knows that in addition to implementing the controller by using only the computer-readable program code, logic programming can be performed on method steps to enable the controller to implement the same function in a form of a logic gate, a switch, an application specific integrated circuit, a programmable logic controller, or an embedded microcontroller. Therefore, the controller can be considered as a hardware component, and an apparatus configured to implement various functions in the controller can also be considered as a structure in the hardware component. Alternatively, an apparatus configured to implement various functions can even be considered as both a software module implementing the method and a structure in the hardware component.

[0148] The systems, apparatuses, modules, or units described in the above-mentioned embodiments can be specifically implemented by a computer chip or an entity, or can be implemented by a product having a certain function. A typical implementation device is a computer. Specifically, the computer can be, for example, a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media player, a navigation device, an email device, a game console, a tablet computer, a wearable device, or a combination of any of these devices.

[0149] For ease of description, the above-mentioned apparatus is described by dividing functions into various units. Certainly, during implementation of this specification, functions of units can be implemented in the same or more software or hardware.

[0150] A person skilled in the art should understand that the embodiments of this specification can be provided as methods, systems, or computer program products. Therefore, the embodiments of this specification can use a form of hardware only embodiments, software only embodiments, or embodiments with a combination of software and hardware. In addition, the embodiments of this specification can use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a disk storage, a CD-ROM, an optical storage, etc.) that include computer-usable program code.

[0151] This specification is described with reference to the flowcharts and/or block diagrams of the method, the device (system), and the computer program product according to the embodiments of this specification. It should be understood that each procedure and/or block in the flowcharts and/or the block diagrams and a combination of procedures and/or blocks in the flowcharts and/or the block diagrams can be implemented by using computer program instructions. These computer program instructions can be provided for a general-purpose computer, a dedicated computer, an embedded processor, or a processor of another programmable data processing device to generate a machine, so

the instructions executed by the computer or the processor of the another programmable data processing device generate an apparatus for implementing a specific function in one or more procedures in the flowcharts and/or in one or more blocks in the block diagrams.

[0152] Alternatively, these computer program instructions can be stored in a computer-readable memory that can instruct a computer or another programmable data processing device to work in a specific manner, so the instructions stored in the computer-readable memory generate an artifact that includes an instruction apparatus. The instruction apparatus implements a specific function in one or more procedures in the flowcharts and/or in one or more blocks in the block diagrams.

[0153] Alternatively, these computer program instructions can be loaded onto a computer or another programmable data processing device, so that a series of operations and steps are performed on the computer or the another programmable device, to generate computer-implemented processing. Therefore, the instructions executed on the computer or the another programmable device provide steps for implementing a specific function in one or more procedures in the flowcharts and/or in one or more blocks in the block diagrams.

[0154] In a typical configuration, a computing device includes one or more processors (CPUs), one or more input/output interfaces, one or more network interfaces, and one or more memories.

[0155] The memory can include a form such as a non-permanent memory, a random access memory (RAM), or a nonvolatile memory in a computer-readable medium, for example, a read-only memory (ROM) or a flash memory (flash RAM). The memory is an example of the computer-readable medium.

[0156] The computer-readable medium includes persistent, non-persistent, removable and non-removable media that can store information by using any method or technology. The information can be computer-readable instructions, a data structure, a program module, or other data. Examples of the computer storage medium include but are not limited to a phase change random access memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), another type of random access memory (RAM), a read-only memory (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or another memory technology, a compact disc read-only memory (CD-ROM), a digital versatile disc (DVD) or another optical storage, a cassette magnetic tape, a magnetic tape/magnetic disk storage, another magnetic storage device, or any other non-transmission medium. The computer storage medium can be configured to store information that can be accessed by a computing device. Based on the definition in this specification, the computer-readable medium does not include transitory media such as a modulated data signal and carrier.

[0157] It should be further noted that the terms "include", "comprise", or any other variants

thereof are intended to cover a non-exclusive inclusion, so that a process, a method, a product, or a device that includes a list of elements not only includes those elements but also includes other elements which are not expressly listed, or further includes elements inherent to such a process, method, product, or device. Without more constraints, an element preceded by "includes a ..." does not preclude the presence of additional identical elements in the process, method, product, or device that includes the element.

[0158] This specification can be described in a general context of a computer-executable instruction executed by a computer, for example, a program module. Generally, the program module includes a routine, a program, an object, a component, a data structure, etc. for executing a specific task or implementing a specific abstract data type. This specification can alternatively be practiced in distributed computing environments. In the distributed computing environments, tasks are executed by remote processing devices connected through a communication network. In the distributed computing environment, a program module can be located in local and remote computer storage media including a storage device.

[0159] The embodiments of this specification are described in a progressive manner. For same or similar parts in the embodiments, mutual references can be made to the embodiments. Each embodiment focuses on a difference from other embodiments. In particular, the apparatus embodiments, the device embodiments, and the nonvolatile computer storage medium embodiments are basically similar to the method embodiments, and therefore are described briefly. For related parts, references can be made to some descriptions of the method embodiments.

[0160] Specific embodiments of this specification are described above. Other embodiments fall within the scope of the appended claims. In some cases, the actions or steps described in the claims can be performed in a sequence different from that in the embodiments and desired results can still be achieved. In addition, the process depicted in the accompanying drawings does not necessarily need a particular sequence or consecutive sequence to achieve the desired results. In some implementations, multi-tasking and concurrent processing are feasible or may be advantageous.

[0161] The above-mentioned descriptions are merely one or more embodiments of this specification and are not intended to limit this specification. A person skilled in the art can make various changes and variations to one or more embodiments of this specification. Any modification, equivalent replacement, improvement, etc. made within the spirit and principle of one or more embodiments of this specification shall fall within the scope of the claims of this specification.

CLAIMS

1. A service processing method for a smartwatch, applied to a watch end on the smartwatch, wherein the method comprises:

obtaining Socket communication trigger data;

performing encapsulation by using a short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet;

sending the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs encapsulation by using the short-range wireless communication protocol and based on Socket communication response data returned by the serving end, to obtain a second data packet; and

receiving the second data packet returned from the mobile phone end by performing short-range wireless communication with the smartphone, and parsing the second data packet to obtain the Socket communication response data.

2. The method according to claim 1, wherein obtaining the Socket communication trigger data specifically comprises:

encapsulating a Socket API parameter based on a predesigned Socket API parameter structure, to obtain a Socket API structure object; and

performing serialization processing on the Socket API structure object, to obtain the Socket communication trigger data.

3. The method according to claim 1, wherein performing the encapsulation based on the Socket communication trigger data, to obtain a first data packet specifically comprises:

encapsulating the Socket communication trigger data into packets to obtain a plurality of first data packets, so that after obtaining the plurality of first data packets, the mobile phone end correspondingly assembles and parses the packets to obtain the Socket communication trigger data.

4. The method according to claim 1, further comprising:

after a specified watch proxy connection between the mobile phone end and the serving end is established, receiving, by performing short-range wireless communication with the smartphone, information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection, wherein

the watch proxy connection is triggered to be established based on short-range wireless communication between the watch end and the mobile phone end.

5. The method according to claim 4, wherein the watch end and the serving end belong to payment applications or payment channel applications;

before the receiving information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection, the method further comprises:

making a watch payment through the watch end, wherein the watch payment is completed through two-dimensional code scanning interaction or short-range wireless communication between the smartphone and a target checkout device; and

the receiving information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection specifically comprises:

receiving detailed payment result information that corresponds to the watch payment, that is sent by the mobile phone end, and that is actively pushed by the serving end through the watch proxy connection, wherein the detailed payment result information comprises at least an actual payment amount and an account balance after the payment.

6. The method according to claim 1, wherein sending the first data packet to the mobile phone end on the smartphone specifically comprises:

sending the first data packet indicating a first service task of the watch end to the mobile phone end on the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, and performs Socket communication with the corresponding serving end based on the Socket communication trigger data and data indicating a second service task of the mobile phone end, wherein

the second data packet is obtained through encapsulation after the mobile phone end intercepts, from the Socket communication response data, second task result data obtained by executing the second service task by the serving end, so that the watch end obtains remaining first task result data obtained by executing the first service task by the serving end.

7. The method according to claim 6, wherein the watch end and the mobile phone end respectively belong to different applications, or the first service task and the second service task belong to different services.

8. The method according to claim 6, wherein sending the first data packet to the mobile phone end on the smartphone by performing short-range wireless communication with the smartphone specifically comprises:

determining at least one mobile phone end that is in a plurality of different mobile phone ends on the smartphone and that currently has a service task that can be delayed for execution as a target mobile phone end; and

sending the first data packet to the target mobile phone end on the smartphone by performing

short-range wireless communication with the smartphone, so that the target mobile phone end uses the service task that can be delayed for execution as the second service task.

9. The method according to claim 8, wherein determining the at least one mobile phone end that is in the plurality of different mobile phone ends on the smartphone and that currently has the service task that can be delayed for execution as the target mobile phone end specifically comprises:

receiving asynchronous communication requirement information sent by the smartphone through short-range wireless communication, wherein the asynchronous communication requirement information reflects asynchronous communication currently being performed or predicted asynchronous communication to be performed between a corresponding mobile phone end on the smartphone and the serving end; and

determining the target mobile phone end from the plurality of corresponding different mobile phone ends on the smartphone based on the received asynchronous communication requirement information.

10. The method according to claim 1, wherein the short-range wireless communication protocol is a Bluetooth protocol or a Bluetooth low energy protocol.

11. A service processing device for a smartwatch, applied to a watch end on the smartwatch, wherein the device comprises:

at least one processor; and

a memory communicatively connected to the at least one processor, wherein

the memory stores instructions that can be executed by the at least one processor, and when the instructions are executed by the at least one processor, the service processing device is caused to:

obtain Socket communication trigger data;

perform encapsulation by using a short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet;

send the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs encapsulation by using the short-range wireless communication protocol and based on Socket communication response data returned by the serving end, to obtain a second data packet; and

receive the second data packet returned from the mobile phone end by performing short-range wireless communication with the smartphone, and parse the second data packet to obtain the Socket communication response data.

12. The service processing device according to claim 11, wherein the service processing device

being caused to obtain the Socket communication trigger data specifically comprises being caused to:
encapsulate a Socket API parameter based on a predesigned Socket API parameter structure, to obtain a Socket API structure object; and

perform serialization processing on the Socket API structure object, to obtain the Socket communication trigger data.

13. The service processing device according to claim 11, wherein the service processing device being caused to perform the encapsulation based on the Socket communication trigger data, to obtain a first data packet specifically comprises being caused to:

encapsulate the Socket communication trigger data into packets to obtain a plurality of first data packets, so that after obtaining the plurality of first data packets, the mobile phone end correspondingly assembles and parses the packets to obtain the Socket communication trigger data.

14. The service processing device according to claim 11, wherein the service processing device further comprises being caused to:

after a specified watch proxy connection between the mobile phone end and the serving end is established, receive, by performing short-range wireless communication with the smartphone, information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection, wherein

the watch proxy connection is triggered to be established based on short-range wireless communication between the watch end and the mobile phone end.

15. The service processing device according to claim 14, wherein the watch end and the serving end belong to payment applications or payment channel applications;

before the receiving information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection, the service processing device further comprises being caused to:

make a watch payment through the watch end, wherein the watch payment is completed through two-dimensional code scanning interaction or short-range wireless communication between the smartphone and a target checkout device; and

the receiving information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection specifically comprises being caused to:

receive detailed payment result information that corresponds to the watch payment, that is sent by the mobile phone end, and that is actively pushed by the serving end through the watch proxy connection, wherein the detailed payment result information comprises at least an actual payment amount and an account balance after the payment.

16. A non-transitory computer-readable storage medium, wherein the non-transitory computer-

readable storage medium stores a computer program, which when executed by a processor causes the processor to:

- obtain Socket communication trigger data;

- perform encapsulation by using a short-range wireless communication protocol supported by the smartwatch and based on the Socket communication trigger data, to obtain a first data packet;

- send the first data packet to a mobile phone end on a smartphone by performing short-range wireless communication with the smartphone, so that the mobile phone end parses the first data packet to obtain the Socket communication trigger data, performs Socket communication with a corresponding serving end based on the Socket communication trigger data, and performs encapsulation by using the short-range wireless communication protocol and based on Socket communication response data returned by the serving end, to obtain a second data packet; and

- receive the second data packet returned from the mobile phone end by performing short-range wireless communication with the smartphone, and parse the second data packet to obtain the Socket communication response data.

17. The non-transitory computer-readable storage medium according to claim 16, wherein the processor being caused to obtain the Socket communication trigger data specifically comprises being caused to:

- encapsulate a Socket API parameter based on a predesigned Socket API parameter structure, to obtain a Socket API structure object; and

- perform serialization processing on the Socket API structure object, to obtain the Socket communication trigger data.

18. The non-transitory computer-readable storage medium according to claim 16, wherein the processor being caused to perform the encapsulation based on the Socket communication trigger data, to obtain a first data packet specifically comprises being caused to:

- encapsulate the Socket communication trigger data into packets to obtain a plurality of first data packets, so that after obtaining the plurality of first data packets, the mobile phone end correspondingly assembles and parses the packets to obtain the Socket communication trigger data.

19. The non-transitory computer-readable storage medium according to claim 16, wherein the processor further comprises being caused to:

- after a specified watch proxy connection between the mobile phone end and the serving end is established, receive, by performing short-range wireless communication with the smartphone, information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection, wherein

- the watch proxy connection is triggered to be established based on short-range wireless

communication between the watch end and the mobile phone end.

20. The non-transitory computer-readable storage medium according to claim 19, wherein the watch end and the serving end belong to payment applications or payment channel applications;

before the receiving information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection, the processor further comprises being caused to:

make a watch payment through the watch end, wherein the watch payment is completed through two-dimensional code scanning interaction or short-range wireless communication between the smartphone and a target checkout device; and

the receiving information that is sent by the mobile phone end and that is actively pushed by the serving end through the watch proxy connection specifically comprises being caused to:

receive detailed payment result information that corresponds to the watch payment, that is sent by the mobile phone end, and that is actively pushed by the serving end through the watch proxy connection, wherein the detailed payment result information comprises at least an actual payment amount and an account balance after the payment.