CLAIMS

1. A sub-application page processing method, performed by a computer device and comprising:

when an event of requesting to obtain a sub-application page is detected, obtaining page code of the sub-application page, the sub-application page being a page displayed in a sub-application;

parsing the page code to obtain a page parsing result, and traversing a code set in the page parsing result to create a page node tree corresponding to the sub-application page, the page node tree being a tree structure generated based on a page node of the sub-application page;

creating a first instance object for converting the page node tree, and establishing a binding relationship between the first instance object and the sub-application page;

in a case that the binding relationship is established, converting the page node tree into an interface invocation form by using the first instance object, to obtain interface invocation information corresponding to the page node tree; and

performing page drawing based on the interface invocation information to generate the sub-application page.

2. The method according to claim 1, wherein the creating a first instance object for converting the page node tree, and establishing a binding relationship between the first instance object and the sub-application page comprises:

obtaining, based on a protocol mode of a preconfigured interface protocol, a first interface class for converting the page node tree, the first interface class comprising a logical method for converting the page node tree;

creating, based on the first interface class, the first instance object for converting the page node tree; and

creating, by using the first instance object as a creation parameter, a root component instance corresponding to the sub-application page to establish the binding relationship between the first instance object and the sub-application page.

3. The method according to claim 2, wherein the creating, based on the first interface class, the first instance object for converting the page node tree comprises:

creating, by using a first keyword for class-based interface implementation, a first

implementation class corresponding to the first interface class, the first implementation class being configured for implementing the logical method for converting the page node tree in the first interface class; and

instantiating, by using a second keyword for class-based instance object creation, the first implementation class to create a first instance object corresponding to the first implementation class.

4. The method according to any one of claims 1 to 3, wherein the converting the page node tree into an interface invocation form by using the first instance object, to obtain interface invocation information corresponding to the page node tree comprises:

starting from a root node of the page node tree, sequentially traversing page nodes in the page node tree based on first interface configuration information provided by the first instance object, to respectively determine first invocation information corresponding to each page node and obtain second invocation information corresponding to page configuration information of the page node tree; and

obtaining, based on the first invocation information and the second invocation information, the interface invocation information corresponding to the page node tree.

5. The method according to claim 4, wherein the starting from a root node of the page node tree, sequentially traversing page nodes in the page node tree based on first interface configuration information provided by the first instance object, to respectively determine first invocation information corresponding to each page node comprises:

for each page node in the page node tree, determining, from the first interface configuration information provided by the first instance object, first interface information matching a page node type of the page node, and determining, based on the first interface information, first invocation information corresponding to the page node.

6. The method according to claim 5, wherein the determining, based on the first interface information, first invocation information corresponding to the page node comprises:

obtaining a second instance object based on the first interface information, the second instance object providing second interface configuration information for converting the page node tree;

obtaining, by using the second interface configuration information provided by the second instance object, second interface information corresponding to the page node; and

obtaining, based on the first interface information and the second interface information, the

first invocation information corresponding to the page node.

7. The method according to claim 6, wherein the method further comprises:

creating, by using the first instance object, a second interface class for converting a page node, the second interface class comprising a logical method for converting a page node;

creating, by using a first keyword for class-based interface implementation, a second implementation class corresponding to the second interface class, the second implementation class being configured for implementing the logical method for converting a page node in the second interface class; and

instantiating, by using a second keyword for class-based instance object creation, the second implementation class to create a second instance object corresponding to the second implementation class.

8. The method according to claim 6, wherein the obtaining, by using the second interface configuration information provided by the second instance object, second interface information corresponding to the page node comprises:

obtaining, based on node configuration information of the page node, the second interface information corresponding to the page node from the second interface configuration information provided by the second instance object.

9. The method according to claim 6, wherein the obtaining, based on the first interface information and the second interface information, the first invocation information corresponding to the page node comprises:

when the page node is a component node, obtaining a pre-created third instance object based on the second interface information, the third instance object being an instance object for converting the component node;

obtaining, by using the third instance object, third interface information corresponding to the component node; and

obtaining, based on the first interface information, the second interface information, and the third interface information that correspond to the component node, first invocation information corresponding to the component node.

10. The method according to claim 9, wherein the obtaining, by using the third instance object, third interface information corresponding to the component node comprises:

starting from a root node of the component node, sequentially traversing child nodes in the

component node based on third interface configuration information provided by the third instance object, to respectively determine child node invocation information corresponding to each child node; and

obtaining, based on the child node invocation information corresponding to each child node, the third interface information corresponding to the component node.

11. The method according to claim 10, wherein the starting from a root node of the component node, sequentially traversing child nodes in the component node based on third interface configuration information provided by the third instance object, to respectively determine child node invocation information corresponding to each child node comprises:

for each child node in the component node, determining, from the third interface configuration information provided by the third instance object, child node interface information matching a child node type of the child node; and

obtaining a second instance object based on the child node interface information, and obtaining, by using the second instance object, child node invocation information corresponding to the child node.

12. The method according to claim 11, wherein the obtaining, by using the second instance object, child node invocation information corresponding to the child node comprises:

obtaining, based on child node configuration information of the child node, the child node invocation information corresponding to the child node from second interface configuration information provided by the second instance object.

13. The method according to any one of claims 1 to 12, wherein the parsing the page code to obtain a page parsing result comprises:

determining a markup language used to write the page code, and compiling the page code by using a markup language compiler corresponding to the markup language to obtain the page parsing result.

14. A sub-application page processing apparatus, comprising:

a code obtaining module, configured to: when an event of requesting to obtain a sub-application page is detected, obtain page code of the sub-application page, the sub-application page being a page displayed in a sub-application;

a node tree generation module, configured to parse the page code to obtain a page parsing result, and traverse a code set in the page parsing result to create a page node tree corresponding to

the sub-application page, the page node tree being a tree structure generated based on a page node of the sub-application page;

a page initialization module, configured to create a first instance object for converting the page node tree, and establish a binding relationship between the first instance object and the sub-application page;

a node tree conversion module, configured to: in a case that the binding relationship is established, convert the page node tree into an interface invocation form by using the first instance object, to obtain interface invocation information corresponding to the page node tree; and

a page drawing module, configured to perform page drawing based on the interface invocation information to generate the sub-application page.

15. A computer device, comprising a memory and a processor, the memory storing computer-readable instructions, and the operations of the method according to any one of claims 1 to 13 being implemented when the processor executes the computer-readable instructions.

16. A computer-readable storage medium, having computer-readable instructions stored therein, the operations of the method according to any one of claims 1 to 13 being implemented when the computer-readable instructions are executed by a processor.

17. A computer program product, comprising computer-readable instructions, the operations of the method according to any one of claims 1 to 13 being implemented when the computer-readable instructions are executed by a processor.