(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 110083352 A
(43)申请公布日 2019.08.02

(21)申请号 201910214279.6

(22)申请日 2019.03.20

(71)申请人 平安普惠企业管理有限公司
地址 518000 广东省深圳市前海深港合作区前湾一路1号A栋201室(入驻深圳市前海商务秘书有限公司)

(72)发明人 陈翔

(74)专利代理机构 深圳市精英专利事务所 44242
代理人 林燕云

(51)Int.Cl.
G06F 8/38(2018.01)

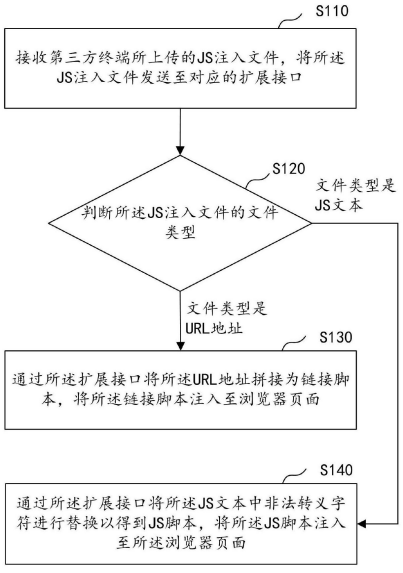
权利要求书2页 说明书10页 附图7页

(54)发明名称

JS代码注入方法、装置、计算机设备及存储介质

(57)摘要

本发明公开了JS代码注入方法、装置、计算机设备及存储介质。该方法包括:接收第三方终端所上传的JS注入文件,将所述JS注入文件发送至对应的扩展接口;判断所述JS注入文件的文件类型;若所述JS注入文件的文件类型是URL地址,通过所述扩展接口将所述URL地址拼接为链接脚本,将所述链接脚本注入至浏览器页面;以及若所述JS注入文件的文件类型是JS文本,通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本,将所述JS脚本注入至所述浏览器页面。该方法通过提供第三方JS注入能力接口,采用数据传输优化技术实现了由第三方提供和管理自己的JS代码,无需针对第三方的需求提供适配,利于后期维护。



CN 110083352 A

1.一种JS代码注入方法,其特征在于,包括:

接收第三方终端所上传的JS注入文件,将所述JS注入文件发送至对应的扩展接口;

判断所述JS注入文件的文件类型;

若所述JS注入文件的文件类型是URL地址,通过所述扩展接口将所述URL地址拼接为链接脚本,将所述链接脚本注入至浏览器页面;以及

若所述JS注入文件的文件类型是JS文本,通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本,将所述JS脚本注入至所述浏览器页面。

2.根据权利要求1所述的JS代码注入方法,其特征在于,所述接收第三方终端所上传的JS注入文件,将所述JS注入文件发送至对应的扩展接口之前,还包括:

通过用于对已存在的类增加方法的Category语言特性对浏览器页面进行接口扩展,得到扩展接口。

3.根据权利要求1所述的JS代码注入方法,其特征在于,所述通过所述扩展接口将所述URL地址拼接为链接脚本,包括:

构建超文本标记语言文件的头部,得到初始处理文件;

将所述URL地址注入至所述初始处理文件中位于头部之后的主体部,得到链接脚本。

4.根据权利要求3所述的JS代码注入方法,其特征在于,所述将所述链接脚本注入至浏览器页面之后,还包括:

根据所述链接脚本获取与所述URL地址对应的网页内容;

将所述网页内容注入至所述浏览器页面。

5.根据权利要求1所述的JS代码注入方法,其特征在于,所述通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本,将所述JS脚本注入至所述浏览器页面,包括:

构建超文本标记语言文件的头部,得到初始处理文件;

判断所述JS文本中是否包括预设的非法转义字符;

若所述JS文本中包括至少一个所述非法转义字符,将所述非法转义字符进行屏蔽,得到处理后JS文本;

将所述处理后JS文本注入至初始处理文件中位于头部之后的主体部,得到所述JS脚本;

若所述JS文本中不包括所述非法转义字符,将所述JS文本注入至初始处理文件中位于头部之后的主体部,得到所述JS脚本。

6.一种JS代码注入装置,其特征在于,包括:

注入文件获取单元,用于接收第三方终端所上传的JS注入文件,将所述JS注入文件发送至对应的扩展接口;

文件类型判断单元,用于判断所述JS注入文件的文件类型;

第一注入单元,用于若所述JS注入文件的文件类型是URL地址,通过所述扩展接口将所述URL地址拼接为链接脚本,将所述链接脚本注入至浏览器页面;以及

第二注入单元,用于若所述JS注入文件的文件类型是JS文本,通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本,将所述JS脚本注入至所述浏览器页面。

7.根据权利要求6所述的JS代码注入装置,其特征在于,还包括:

扩展接口构建单元，用于通过用于对已存在的类增加方法的Category语言特性对浏览器页面进行接口扩展，得到扩展接口。

8.根据权利要求7所述的JS代码注入装置，其特征在于，所述第一注入单元，包括：

第一文件构建单元，用于构建超文本标记语言文件的头部，得到初始处理文件；

URL地址注入单元，用于将所述URL地址注入至所述初始处理文件中位于头部之后的主体部，得到链接脚本。

9.一种计算机设备，包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序，其特征在于，所述处理器执行所述计算机程序时实现如权利要求1至5中任一项所述的JS代码注入方法。

10.一种计算机可读存储介质，其特征在于，所述计算机可读存储介质存储有计算机程序，所述计算机程序当被处理器执行时使所述处理器执行如权利要求1至5任一项所述的JS代码注入方法。

JS代码注入方法、装置、计算机设备及存储介质

技术领域

[0001] 本发明涉及数据传输优化领域，尤其涉及一种JS代码注入方法、装置、计算机设备及存储介质。

背景技术

[0002] 目前，应用程序一般都会引用很多第三方的SDK(全称是Software Development Kit，表示软件开发工具包)，而第三方SDK往往是使用宿主的Webview(网络视图)展示一些页面，这些页面往往需要通过js(全称是javascript，它是一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型)实现Webview网页和SDK进行交互。如果应用程序的开发运营方对针对第三方的页面业务需求做具体的处理适配，那么第三方SDK业务就需和该应用程序耦合，耦合的工作量巨大，不利于后期维护。

发明内容

[0003] 本发明实施例提供了一种JS代码注入方法、装置、计算机设备及存储介质，旨在解决现有技术中针对第三方的页面业务需求做具体的处理适配，第三方SDK业务就需和该应用程序耦合，耦合的工作量巨大，不利于后期维护的问题。

[0004] 第一方面，本发明实施例提供了一种JS代码注入方法，其包括：

[0005] 接收第三方终端所上传的JS注入文件，将所述JS注入文件发送至对应的扩展接口；

[0006] 判断所述JS注入文件的文件类型；

[0007] 若所述JS注入文件的文件类型是URL地址，通过所述扩展接口将所述URL地址拼接为链接脚本，将所述链接脚本注入至浏览器页面；以及

[0008] 若所述JS注入文件的文件类型是JS文本，通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面。

[0009] 第二方面，本发明实施例提供了一种JS代码注入装置，其包括：

[0010] 注入文件获取单元，用于接收第三方终端所上传的JS注入文件，将所述JS注入文件发送至对应的扩展接口；

[0011] 文件类型判断单元，用于判断所述JS注入文件的文件类型；

[0012] 第一注入单元，用于若所述JS注入文件的文件类型是URL地址，通过所述扩展接口将所述URL地址拼接为链接脚本，将所述链接脚本注入至浏览器页面；以及

[0013] 第二注入单元，用于若所述JS注入文件的文件类型是JS文本，通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面。

[0014] 第三方面，本发明实施例又提供了一种计算机设备，其包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序，所述处理器执行所述计算机程序时实现上述第一方面所述的JS代码注入方法。

[0015] 第四方面，本发明实施例还提供了一种计算机可读存储介质，其中所述计算机可读存储介质存储有计算机程序，所述计算机程序当被处理器执行时使所述处理器执行上述第一方面所述的JS代码注入方法。

[0016] 本发明实施例提供了一种JS代码注入方法、装置、计算机设备及存储介质。该方法包括接收第三方终端所上传的JS注入文件，将所述JS注入文件发送至对应的扩展接口；判断所述JS注入文件的文件类型；若所述JS注入文件的文件类型是URL地址，通过所述扩展接口将所述URL地址拼接为链接脚本，将所述链接脚本注入至浏览器页面；以及若所述JS注入文件的文件类型是JS文本，通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面。该方法通过提供第三方JS注入能力接口，实现了由第三方提供和管理自己的JS代码，无需针对第三方的需求提供适配，利于后期维护。

附图说明

[0017] 为了更清楚地说明本发明实施例技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0018] 图1为本发明实施例提供的JS代码注入方法的应用场景示意图；

[0019] 图2为本发明实施例提供的JS代码注入方法的流程示意图；

[0020] 图3为本发明实施例提供的JS代码注入方法的另一流程示意图；

[0021] 图4为本发明实施例提供的JS代码注入方法的子流程示意图；

[0022] 图5为本发明实施例提供的JS代码注入方法的另一子流程示意图；

[0023] 图6为本发明实施例提供的JS代码注入方法的另一子流程示意图；

[0024] 图7为本发明实施例提供的JS代码注入装置的示意性框图；

[0025] 图8为本发明实施例提供的JS代码注入装置的另一示意性框图；

[0026] 图9为本发明实施例提供的JS代码注入装置的子单元示意性框图；

[0027] 图10为本发明实施例提供的JS代码注入装置的另一子单元示意性框图；

[0028] 图11为本发明实施例提供的JS代码注入装置的另一子单元示意性框图；

[0029] 图12为本发明实施例提供的计算机设备的示意性框图。

具体实施方式

[0030] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0031] 应当理解，当在本说明书和所附权利要求书中使用时，术语“包括”和“包含”指示所描述特征、整体、步骤、操作、元素和/或组件的存在，但并不排除一个或多个其它特征、整体、步骤、操作、元素、组件和/或其集合的存在或添加。

[0032] 还应当理解，在此本发明说明书中所使用的术语仅仅是出于描述特定实施例的目的而并不意在限制本发明。如在本发明说明书和所附权利要求书中所使用的那样，除非上

下文清楚地指明其它情况，否则单数形式的“一”、“一个”及“该”意在包括复数形式。

[0033] 还应当进一步理解，在本发明说明书和所附权利要求书中使用的术语“和/或”是指相关联列出的项中的一个或多个的任何组合以及所有可能组合，并且包括这些组合。

[0034] 请参阅图1和图2，图1为本发明实施例提供的JS代码注入方法的应用场景示意图，图2为本发明实施例提供的JS代码注入方法的流程示意图，该JS代码注入方法应用于终端中，该终端可以为手机、平板电脑、手提电脑、台式电脑等电子设备。该终端中安装有应用程序，其中，该应用程序可以为原生应用程序、Web应用程序或混合应用程序。该方法通过安装于终端中的应用软件进行执行。

[0035] 如图2所示，该方法包括步骤S110～S140。

[0036] S110、接收第三方终端所上传的JS注入文件，将所述JS注入文件发送至对应的扩展接口。

[0037] 在本实施例中，终端本地的APP应用程序引入了很多第三方的软件开发工具包(即SDK)，而第三方SDK往往是使用宿主App(即终端本地的APP应用程序)的Webview(网络视图)展示一些页面，这些页面往往需要通过JS注入实现Webview和第三方SDK进行交互。

[0038] 当终端本地的APP应用程序的接口对第三方SDK开放时，可接收第三方SDK的JS注入文件，其中JS注入文件是在浏览器地址栏中输入一段JavaScript代码，用来改变页面JS变量、页面标签的内容。

[0039] 为了使终端本地的APP应用程序中的浏览器页面(即UIWebview，UIWebView类是用来显示网络内容，使用它可以简单的创造一个UIWebView对象，放置到窗口上，并且发送一个指向网络内容的请求。通过这个类，可以控制网页历史的前进后退，也可以通过程序去控制网络内容的属性)具备接收第三方终端所上传的JS注入文件，需要通过Category特性(Category是指对一个类的扩展)对UIWebview扩展一个接口，专用于接收和处理JS注入文件。

[0040] 在一实施例中，如图3所示，步骤S110之前还包括：

[0041] S101、通过用于对已存在的类增加方法的Category语言特性对浏览器页面进行接口扩展，得到扩展接口。

[0042] 在本实施例中，Category是Objective-C 2.0之后添加的语言特性，Category的主要作用是为已经存在的类添加方法，例如对UIWebView类进行接口的扩展从而得到扩展接口。通过对UIWebView类进行接口扩展得到扩展接口，从而具备了接收第三方SDK注入的能力，便于与第三发SDK进行交互。

[0043] S120、判断所述JS注入文件的文件类型。

[0044] 在本实施例中，所述文件类型主要是URL地址和JS文本；其中，URL地址中URL的全称是Uniform Resource Locator，表示统一资源定位符，对可以从互联网上得到的资源的位置和访问方法的一种简洁的表示，是互联网上标准资源的地址；JS文本中JS全称是JavaScript，是网页开发设计语言的一种。判断JS注入文件是为URL地址或是JS文本，是为了有针对性的对JS注入文件进行下一步处理。

[0045] S130、若所述JS注入文件的文件类型是URL地址，通过所述扩展接口将所述URL地址拼接为链接脚本，将所述链接脚本注入至浏览器页面。

[0046] 在本实施例中，若所述JS注入文件的文件类型是URL地址，只需将所述URL地址与

超文本标记语言文件拼接为链接脚本，然后注入至浏览器页面即可。若第三方终端注入的JS注入文件是URL地址，在解析时是获取URL地址对应网页中获取的最新网页内容并注入到浏览器页面，使得第三方拥有在线发布修复JS代码的能力。

[0047] 在一实施例中，如图4所示，步骤S130包括：

[0048] S131、构建超文本标记语言文件的头部，得到初始处理文件；

[0049] S132、将所述URL地址注入至所述初始处理文件中位于头部之后的主体部，得到链接脚本。

[0050] 在本实施例中，超文本标记语言文件即HTML文件(HTML的全称是HyperText Markup Language)，HTML文件的头部即head部分，head标签用于定义文档的头部，它是所有头部元素的容器。<head>中的元素可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等等。文档的头部描述了文档的各种属性和信息，包括文档的标题、在Web中的位置以及和其他文档的关系等。

[0051] 主体部即HTML文件的body部分，其包含文档的所有内容，比如文本、超链接、图像、表格、列表等等；

[0052] HTML文件的头部之前是<html>开头，主体部之后是</html>结尾，用于作为一个完整的HTML文件的开头和结尾；

[0053] 以超文本标记语言文件为载体接收所述URL地址，然后注入到浏览器页面，满足了第三方SDK与终端本地的APP应用程序的JS交互需求。

[0054] 在一实施例中，如图5所示，步骤S132之后还包括：

[0055] S133、根据所述链接脚本获取与所述URL地址对应的网页内容；

[0056] S134、将所述网页内容注入至所述浏览器页面。

[0057] 本实施例中，将URL地址注入到主体部，表示该链接脚本中所获取的JS注入内容是由对应的URL地址获取，若该URL地址下的内容发生更新时，此处将所述链接脚本注入至所述浏览器页面时，是将URL地址下的最新内容注入到所述浏览器页面，使得第三方拥有在线发布修复JS的能力。

[0058] 通过上述方式，终端本地的APP应用程序提供了第三方JS注入的能力，和终端本地的APP应用程序的本地JS注入区分开，防止不安全以及业务耦合，保证了注入的独立性，而且JS注入过程中，不用针对第三方的需求提供适配。

[0059] S140、若所述JS注入文件的文件类型是JS文本，通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面。

[0060] 在一实施例中，如图6所示，步骤S140包括：

[0061] S141、构建超文本标记语言文件的头部，得到初始处理文件；

[0062] S142、判断所述JS文本中是否包括预设的非法转义字符；

[0063] S143、若所述JS文本中包括至少一个所述非法转义字符，将所述非法转义字符进行屏蔽，得到处理后JS文本；

[0064] S144、将所述处理后JS文本注入至初始处理文件中位于头部之后的主体部，得到所述JS脚本；

[0065] S145、若所述JS文本中不包括所述非法转义字符，将所述JS文本注入至初始处理文件中位于头部之后的主体部，得到所述JS脚本。

[0066] 即先构建HTML文件的头部,得到初始处理文件;判断所述JS文本中是否包括预设的非法转义字符;其中,预设的非法转义字符包括\\、\”、\’、\n、\r、\f;若所述JS文本中包括所述非法转义字符,将所述非法转义字符进行屏蔽,得到处理后JS文本,其中,屏蔽\\、\”、\’是均是在前面加上\\,屏蔽\n、\r、\f时是在前面加上\;将所述JS文本注入至初始处理文件中位于头部之后的主体部,得到JS脚本。

[0067] 在本实施例中将JS文本屏蔽非法转义字符后转化成处理后JS文本并注入到主体部,表示该JS脚本中所获取的JS注入内容是直接由所述JS脚本获取,当第三方SDK需要更新JS注入内容时,再次将最新的JS文本由扩展接口注入即可。

[0068] 该方法通过提供第三方JS注入能力接口,实现了由第三方提供和管理自己的JS代码,无需针对第三方的需求提供适配,利于后期维护。

[0069] 本发明实施例还提供一种JS代码注入装置,该JS代码注入装置用于执行前述JS代码注入方法的任一实施例。具体地,请参阅图7,图7是本发明实施例提供的JS代码注入装置的示意性框图。该JS代码注入装置100可以配置于终端中。

[0070] 如图7所示,JS代码注入装置100包括注入文件获取单元110、文件类型判断单元120、第一注入单元130、第二注入单元140。

[0071] 注入文件获取单元110,用于接收第三方终端所上传的JS注入文件,将所述JS注入文件发送至对应的扩展接口。

[0072] 在本实施例中,终端本地的APP应用程序引入了很多第三方的软件开发工具包(即SDK),而第三方SDK往往是使用宿主App(即终端本地的APP应用程序)的Webview(网络视图)展示一些页面,这些页面往往需要通过JS注入实现Webview和第三方SDK进行交互。

[0073] 当终端本地的APP应用程序的接口对第三方SDK开放时,可接收第三方SDK的JS注入文件,其中JS注入文件是在浏览器地址栏中输入一段JavaScript代码,用来改变页面JS变量、页面标签的内容。

[0074] 为了使终端本地的APP应用程序中的浏览器页面(即UIWebview,UIWebView类是用来显示网络内容,使用它可以简单的创造一个UIWebView对象,放置到窗口上,并且发送一个指向网络内容的请求。通过这个类,可以控制网页历史的前进后退,也可以通过程序去控制网络内容的属性)具备接收第三方终端所上传的JS注入文件,需要通过Category特性(Category是指对一个类的扩展)对UIWebview扩展一个接口,专用于接收和处理JS注入文件。

[0075] 在一实施例中,如图8所示,JS代码注入装置100还包括:

[0076] 扩展接口构建单元101,用于通过用于对已存在的类增加方法的Category语言特性对浏览器页面进行接口扩展,得到扩展接口。

[0077] 在本实施例中,Category是Objective-C 2.0之后添加的语言特性,Category的主要作用是为已经存在的类添加方法,例如对UIWebView类进行接口的扩展从而得到扩展接口。通过对UIWebView类进行接口扩展得到扩展接口,从而具备了接收第三方SDK注入的能力,便于与第三发SDK进行交互。

[0078] 文件类型判断单元120,用于判断所述JS注入文件的文件类型。

[0079] 在本实施例中,所述文件类型主要是URL地址和JS文本;其中,URL地址中URL的全称是Uniform Resource Locator,表示统一资源定位符,对可以从互联网上得到的资源的

位置和访问方法的一种简洁的表示，是互联网上标准资源的地址；JS文本中JS全称是JavaScript，是网页开发设计语言的一种。判断JS注入文件是为URL地址或是JS文本，是为了有针对性的对JS注入文件进行下一步处理。

[0080] 第一注入单元130，用于若所述JS注入文件的文件类型是URL地址，通过所述扩展接口将所述URL地址拼接为链接脚本，将所述链接脚本注入至浏览器页面。

[0081] 在本实施例中，若所述JS注入文件的文件类型是URL地址，只需将所述URL地址与超文本标记语言文件拼接为链接脚本，然后注入至浏览器页面即可。若第三方终端注入的JS注入文件是URL地址，在解析时是获取URL地址对应网页中获取的最新网页内容并注入到浏览器页面，使得第三方拥有在线发布修复JS代码的能力。

[0082] 在一实施例中，如图9所示，第一注入单元130包括：

[0083] 第一文件构建单元131，用于构建超文本标记语言文件的头部，得到初始处理文件；

[0084] URL地址注入单元132，用于将所述URL地址注入至所述初始处理文件中位于头部之后的主体部，得到链接脚本。

[0085] 在本实施例中，超文本标记语言文件即HTML文件(HTML的全称是HyperText Markup Language)，HTML文件的头部即head部分，head标签用于定义文档的头部，它是所有头部元素的容器。<head>中的元素可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等等。文档的头部描述了文档的各种属性和信息，包括文档的标题、在Web中的位置以及和其他文档的关系等。

[0086] 主体部即HTML文件的body部分，其包含文档的所有内容，比如文本、超链接、图像、表格、列表等等；

[0087] HTML文件的头部之前是<html>开头，主体部之后是</html>结尾，用于作为一个完整的HTML文件的开头和结尾；

[0088] 以超文本标记语言文件为载体接收所述URL地址，然后注入到浏览器页面，满足了第三方SDK与终端本地的APP应用程序的JS交互需求。

[0089] 在一实施例中，如图10所示，第一注入单元130还包括：

[0090] 网页内容获取单元133，用于根据所述链接脚本获取与所述URL地址对应的网页内容；

[0091] 内容注入单元134，用于将所述网页内容注入至所述浏览器页面。

[0092] 本实施例中，将URL地址注入到主体部，表示该链接脚本中所获取的JS注入内容是由对应的URL地址获取，若该URL地址下的内容发生更新时，此处将所述链接脚本注入至所述浏览器页面时，是将URL地址下的最新内容注入到所述浏览器页面，使得第三方拥有在线发布修复JS的能力。

[0093] 通过上述方式，终端本地的APP应用程序提供了第三方JS注入的能力，和终端本地的APP应用程序的本地JS注入区分开，防止不安全以及业务耦合，保证了注入的独立性，而且JS注入过程中，不用针对第三方的需求提供适配。

[0094] 第二注入单元140，用于若所述JS注入文件的文件类型是JS文本，通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面。

[0095] 在一实施例中，如图11所示，第二注入单元140包括：

[0096] 第二文件构建单元141，用于构建超文本标记语言文件的头部，得到初始处理文件；

[0097] 非法转义字符判断单元142，用于判断所述JS文本中是否包括预设的非法转义字符；

[0098] 字符屏蔽转化单元143，用于若所述JS文本中包括至少一个所述非法转义字符，将所述非法转义字符进行屏蔽，得到处理后JS文本；

[0099] 第一JS脚本获取单元144，用于将所述处理后JS文本注入至初始处理文件中位于头部之后的主体部，得到所述JS脚本；

[0100] 第二JS脚本获取单元145，用于若所述JS文本中不包括所述非法转义字符，将所述JS文本注入至初始处理文件中位于头部之后的主体部，得到所述JS脚本。

[0101] 即先构建HTML文件的头部，得到初始处理文件；判断所述JS文本中是否包括预设的非法转义字符；其中，预设的非法转义字符包括\\、\”、\’、\n、\r、\f；若所述JS文本中包括所述非法转义字符，将所述非法转义字符进行屏蔽，得到处理后JS文本，其中，屏蔽\\、\”、\’是均是在前面加上\\，屏蔽\n、\r、\f时是在前面加上\；将所述JS文本注入至初始处理文件中位于头部之后的主体部，得到JS脚本。

[0102] 在本实施例中将JS文本屏蔽非法转义字符后转化成处理后JS文本并注入到主体部，表示该JS脚本中所获取的JS注入内容是直接由所述JS脚本获取，当第三方SDK需要更新JS注入内容时，再次将最新的JS文本由扩展接口注入即可。

[0103] 该装置通过提供第三方JS注入能力接口，实现了由第三方提供和管理自己的JS代码，无需针对第三方的需求提供适配，利于后期维护。

[0104] 上述JS代码注入装置可以实现为计算机程序的形式，该计算机程序可以在如图12所示的计算机设备上运行。

[0105] 请参阅图12，图12是本发明实施例提供的计算机设备的示意性框图。该计算机设备500是终端，该终端可以为手机、平板电脑、手提电脑、台式电脑等电子设备。

[0106] 参阅图12，该计算机设备500包括通过系统总线501连接的处理器502、存储器和网络接口505，其中，存储器可以包括非易失性存储介质503和内存储器504。

[0107] 该非易失性存储介质503可存储操作系统5031和计算机程序5032。该计算机程序5032被执行时，可使得处理器502执行JS代码注入方法。

[0108] 该处理器502用于提供计算和控制能力，支撑整个计算机设备500的运行。

[0109] 该内存储器504为非易失性存储介质503中的计算机程序5032的运行提供环境，该计算机程序5032被处理器502执行时，可使得处理器502执行JS代码注入方法。

[0110] 该网络接口505用于进行网络通信，如提供数据信息的传输等。本领域技术人员可以理解，图12中示出的结构，仅仅是与本发明方案相关的部分结构的框图，并不构成对本发明方案所应用于其上的计算机设备500的限定，具体的计算机设备500可以包括比图中所示更多或更少的部件，或者组合某些部件，或者具有不同的部件布置。

[0111] 其中，所述处理器502用于运行存储在存储器中的计算机程序5032，以实现如下功能：接收第三方终端所上传的JS注入文件，将所述JS注入文件发送至对应的扩展接口；判断所述JS注入文件的文件类型；若所述JS注入文件的文件类型是URL地址，通过所述扩展接口

将所述URL地址拼接为链接脚本，将所述链接脚本注入至浏览器页面；以及若所述JS注入文件的文件类型是JS文本，通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面。

[0112] 在一实施例中，处理器502在执行所述所述接收第三方终端所上传的JS注入文件，将所述JS注入文件发送至对应的扩展接口的步骤之前，还执行如下操作：通过用于对已存在的类增加方法的Category语言特性对浏览器页面进行接口扩展，得到扩展接口。

[0113] 在一实施例中，处理器502在执行所述通过所述扩展接口将所述URL地址拼接为链接脚本的步骤时，执行如下操作：构建超文本标记语言文件的头部，得到初始处理文件；将所述URL地址注入至所述初始处理文件中位于头部之后的主体部，得到链接脚本。

[0114] 在一实施例中，处理器502在执行所述将所述链接脚本注入至浏览器页面的步骤之后，还执行如下操作：根据所述链接脚本获取与所述URL地址对应的网页内容；将所述网页内容注入至所述浏览器页面。

[0115] 在一实施例中，处理器502在执行所述通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面的步骤时，执行如下操作：构建超文本标记语言文件的头部，得到初始处理文件；判断所述JS文本中是否包括预设的非法转义字符；若所述JS文本中包括至少一个所述非法转义字符，将所述非法转义字符进行屏蔽，得到处理后JS文本；将所述处理后JS文本注入至初始处理文件中位于头部之后的主体部，得到所述JS脚本；若所述JS文本中不包括所述非法转义字符，将所述JS文本注入至初始处理文件中位于头部之后的主体部，得到所述JS脚本。

[0116] 本领域技术人员可以理解，图12中示出的计算机设备的实施例并不构成对计算机设备具体构成的限定，在其他实施例中，计算机设备可以包括比图示更多或更少的部件，或者组合某些部件，或者不同的部件布置。例如，在一些实施例中，计算机设备可以仅包括存储器及处理器，在这样的实施例中，存储器及处理器的结构及功能与图12所示实施例一致，在此不再赘述。

[0117] 应当理解，在本发明实施例中，处理器502可以是中央处理单元(Central Processing Unit，CPU)，该处理器502还可以是其他通用处理器、数字信号处理器(Digital Signal Processor，DSP)、专用集成电路(Application Specific Integrated Circuit，ASIC)、现成可编程门阵列(Field-Programmable GateArray，FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。其中，通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0118] 在本发明的另一实施例中提供计算机可读存储介质。该计算机可读存储介质可以为非易失性的计算机可读存储介质。该计算机可读存储介质存储有计算机程序，其中计算机程序被处理器执行时实现以下步骤：接收第三方终端所上传的JS注入文件，将所述JS注入文件发送至对应的扩展接口；判断所述JS注入文件的文件类型；若所述JS注入文件的文件类型是URL地址，通过所述扩展接口将所述URL地址拼接为链接脚本，将所述链接脚本注入至浏览器页面；以及若所述JS注入文件的文件类型是JS文本，通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面。

[0119] 在一实施例中，所述接收第三方终端所上传的JS注入文件，将所述JS注入文件发送至对应的扩展接口之前，还包括：通过用于对已存在的类增加方法的Category语言特性

对浏览器页面进行接口扩展，得到扩展接口。

[0120] 在一实施例中，所述通过所述扩展接口将所述URL地址拼接为链接脚本，包括：构建超文本标记语言文件的头部，得到初始处理文件；将所述URL地址注入至所述初始处理文件中位于头部之后的主体部，得到链接脚本。

[0121] 在一实施例中，所述将所述链接脚本注入至浏览器页面，还包括：根据所述链接脚本获取与所述URL地址对应的网页内容；将所述网页内容注入至所述浏览器页面。

[0122] 在一实施例中，所述通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面，包括：构建超文本标记语言文件的头部，得到初始处理文件；判断所述JS文本中是否包括预设的非法转义字符；若所述JS文本中包括至少一个所述非法转义字符，将所述非法转义字符进行屏蔽，得到处理后JS文本；将所述处理后JS文本注入至初始处理文件中位于头部之后的主体部，得到所述JS脚本；若所述JS文本中不包括所述非法转义字符，将所述JS文本注入至初始处理文件中位于头部之后的主体部，得到所述JS脚本。

[0123] 所属领域的技术人员可以清楚地了解到，为了描述的方便和简洁，上述描述的设备、装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、计算机软件或者二者的结合来实现，为了清楚地说明硬件和软件的可互换性，在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本发明的范围。

[0124] 在本发明所提供的几个实施例中，应该理解到，所揭露的设备、装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为逻辑功能划分，实际实现时可以有另外的划分方式，也可以将具有相同功能的单元集合成一个单元，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另外，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口、装置或单元的间接耦合或通信连接，也可以是电的，机械的或其它的形式连接。

[0125] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本发明实施例方案的目的。

[0126] 另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以是两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

[0127] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个存储介质中。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分，或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备(可以是

个人计算机,终端,或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0128] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到各种等效的修改或替换,这些修改或替换都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求的保护范围为准。

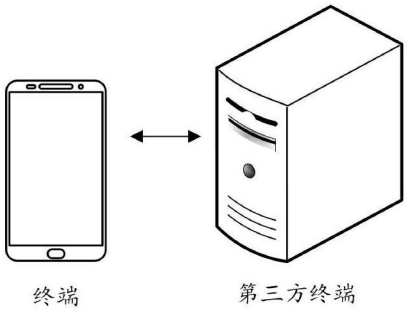

终端
第三方终端

图1

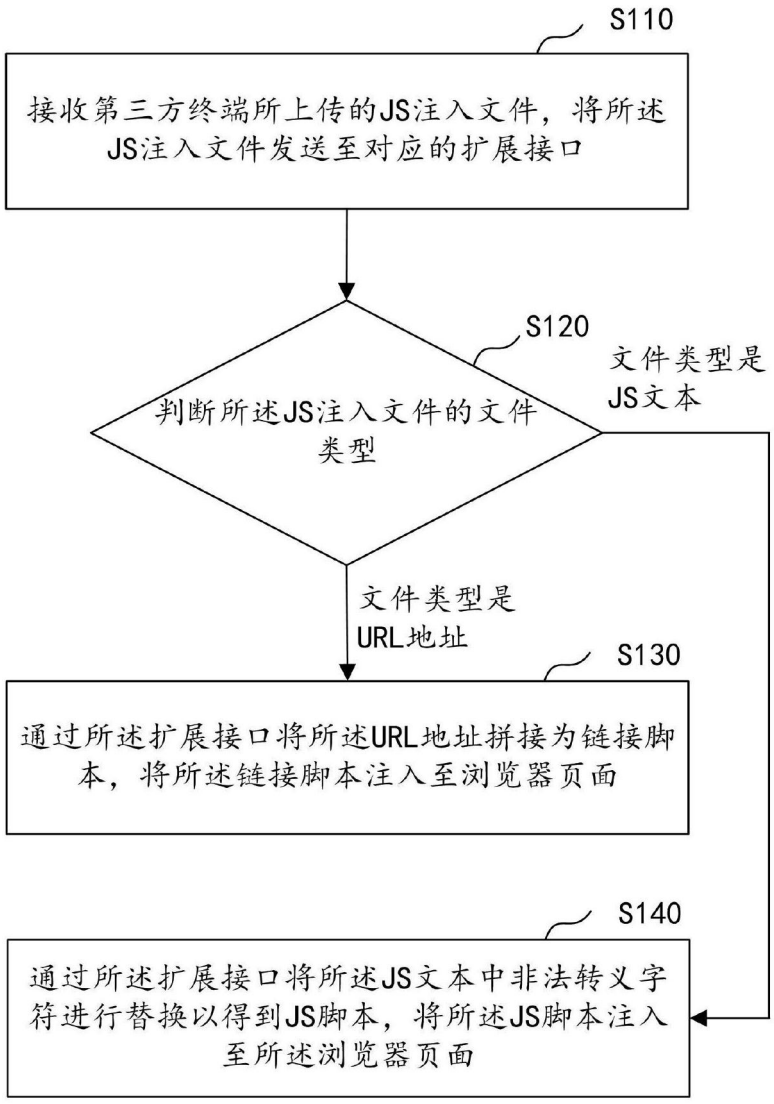

S110
接收第三方终端所上传的JS注入文件，将所述JS注入文件发送至对应的扩展接口
S120
判断所述JS注入文件的文件类型
文件类型是JS文本
文件类型是URL地址
S130
通过所述扩展接口将所述URL地址拼接为链接脚本，将所述链接脚本注入至浏览器页面
S140
通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面

图2

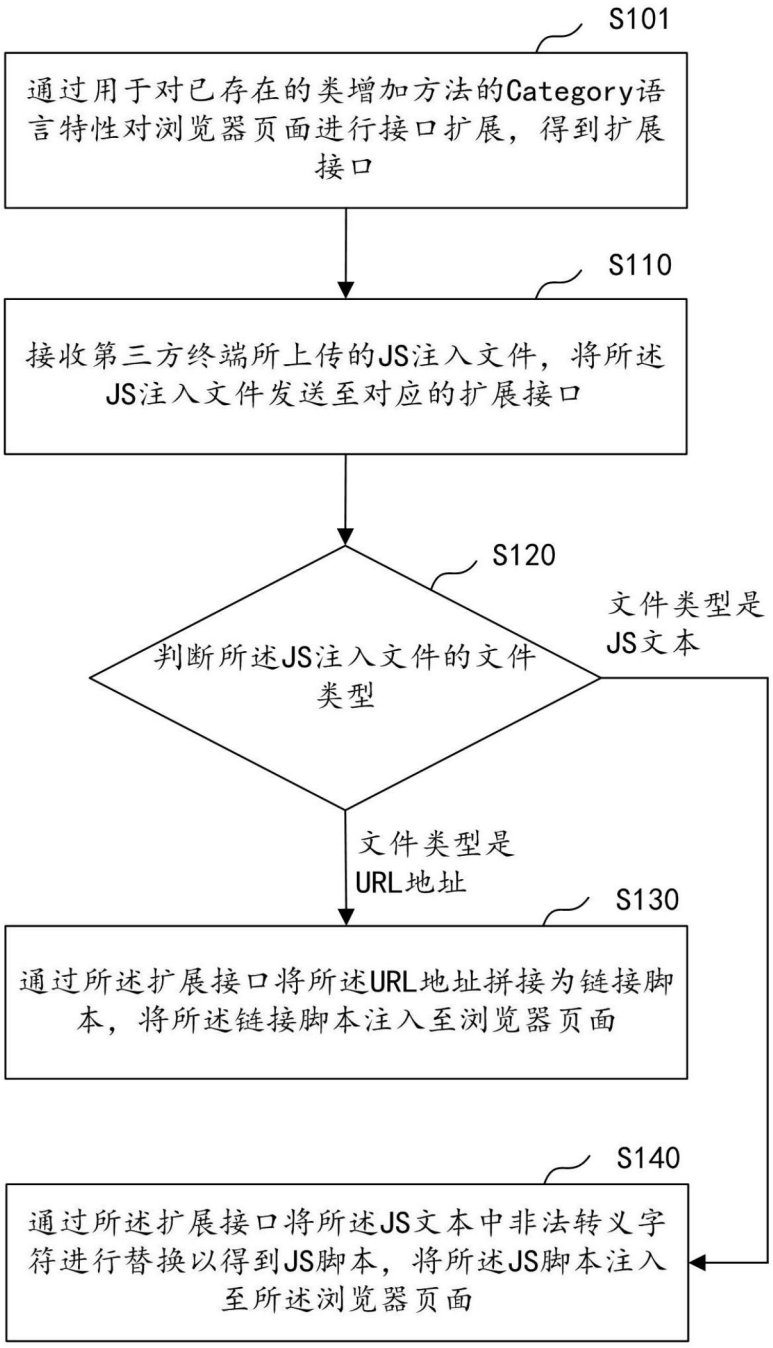

S101
通过用于对已存在的类增加方法的Category语言特性对浏览器页面进行接口扩展，得到扩展接口
S110
接收第三方终端所上传的JS注入文件，将所述JS注入文件发送至对应的扩展接口
S120
判断所述JS注入文件的文件类型
文件类型是JS文本
文件类型是URL地址
S130
通过所述扩展接口将所述URL地址拼接为链接脚本，将所述链接脚本注入至浏览器页面
S140
通过所述扩展接口将所述JS文本中非法转义字符进行替换以得到JS脚本，将所述JS脚本注入至所述浏览器页面

图3

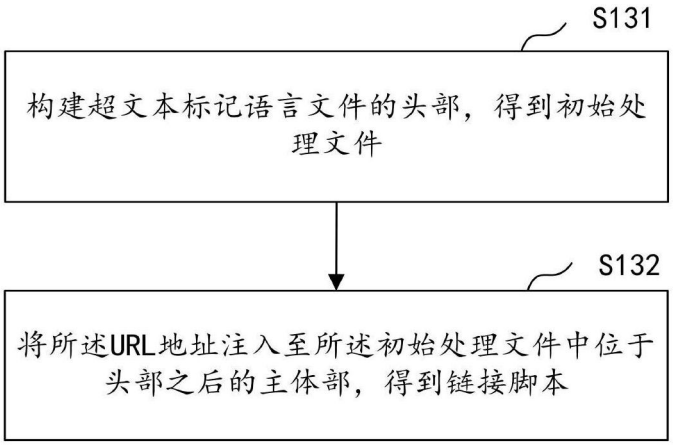

S131
构建超文本标记语言文件的头部，得到初始处理文件
S132
将所述URL地址注入至所述初始处理文件中位于头部之后的主体部，得到链接脚本

图4

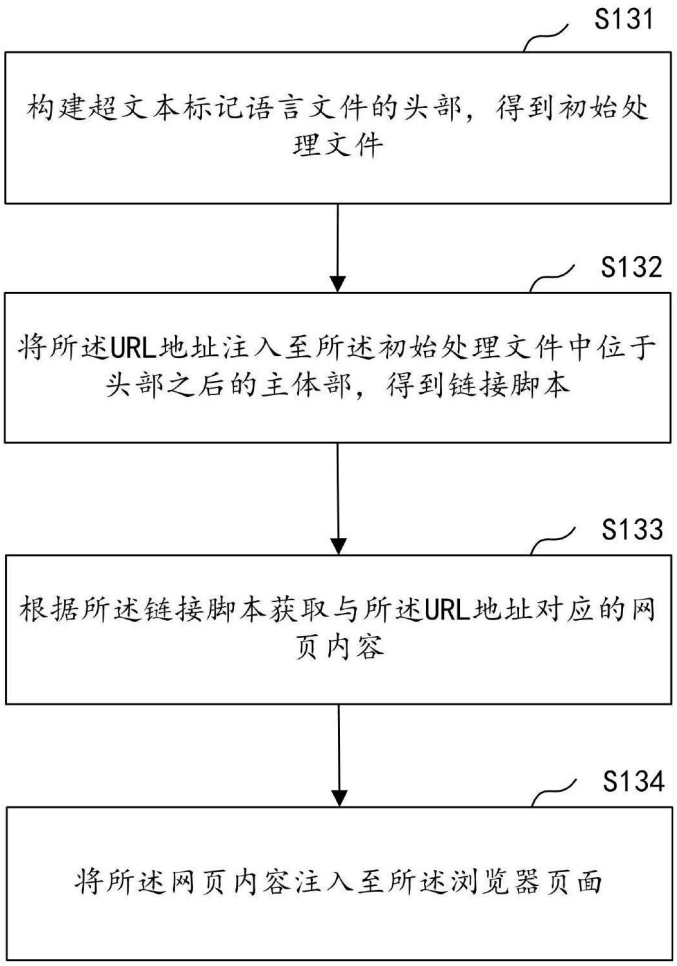

S131
构建超文本标记语言文件的头部，得到初始处理文件
S132
将所述URL地址注入至所述初始处理文件中位于头部之后的主体部，得到链接脚本
S133
根据所述链接脚本获取与所述URL地址对应的网页内容
S134
将所述网页内容注入至所述浏览器页面

图5

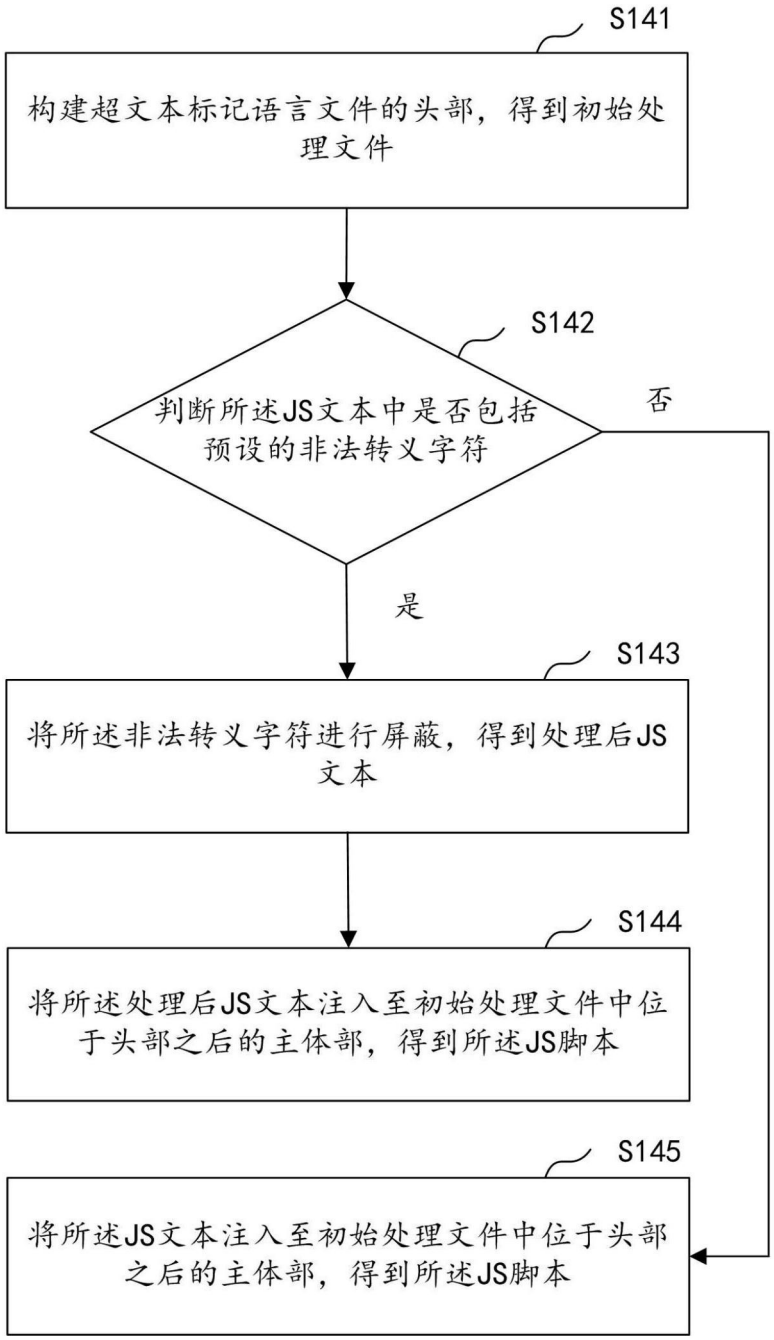

S141
构建超文本标记语言文件的头部，得到初始处理文件
S142
判断所述JS文本中是否包括预设的非法转义字符
否
是
S143
将所述非法转义字符进行屏蔽，得到处理后JS文本
S144
将所述处理后JS文本注入至初始处理文件中位于头部之后的主体部，得到所述JS脚本
S145
将所述JS文本注入至初始处理文件中位于头部之后的主体部，得到所述JS脚本

图6

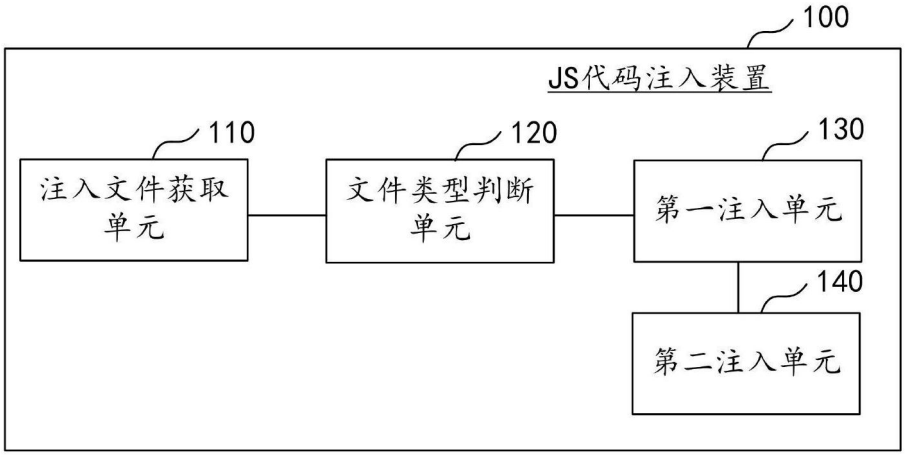

100
JS代码注入装置
110
注入文件获取单元
120
文件类型判断单元
130
第一注入单元
140
第二注入单元

图7

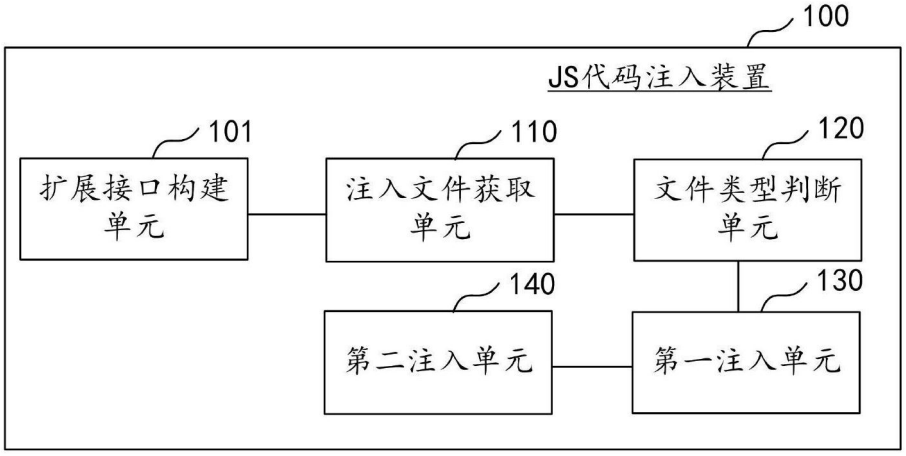

100
JS代码注入装置
101
扩展接口构建单元
110
注入文件获取单元
120
文件类型判断单元
140
第二注入单元
130
第一注入单元

图8

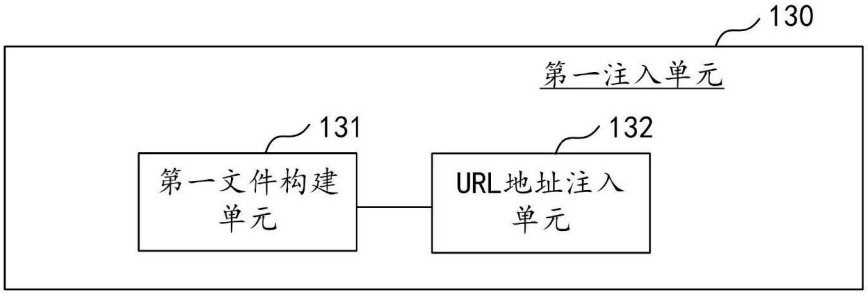

130
第一注入单元
131
第一文件构建单元
132
URL地址注入单元

图9

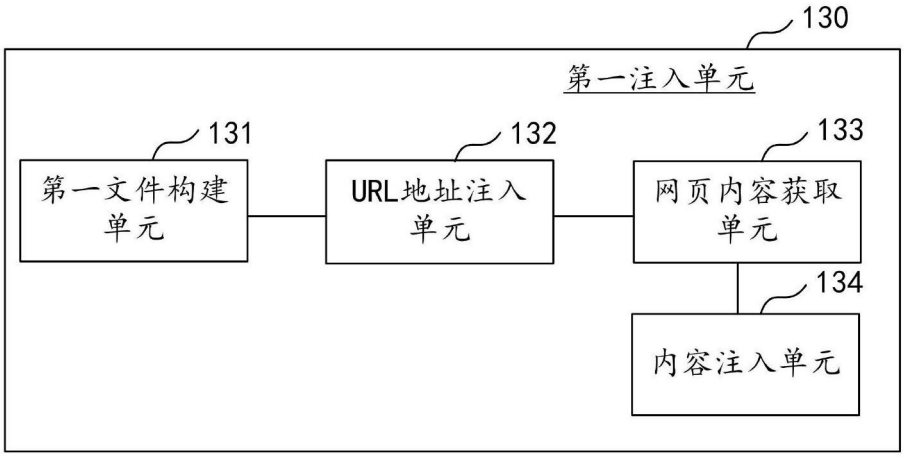

130
第一注入单元
131
第一文件构建单元
132
URL地址注入单元
133
网页内容获取单元
134
内容注入单元

图10

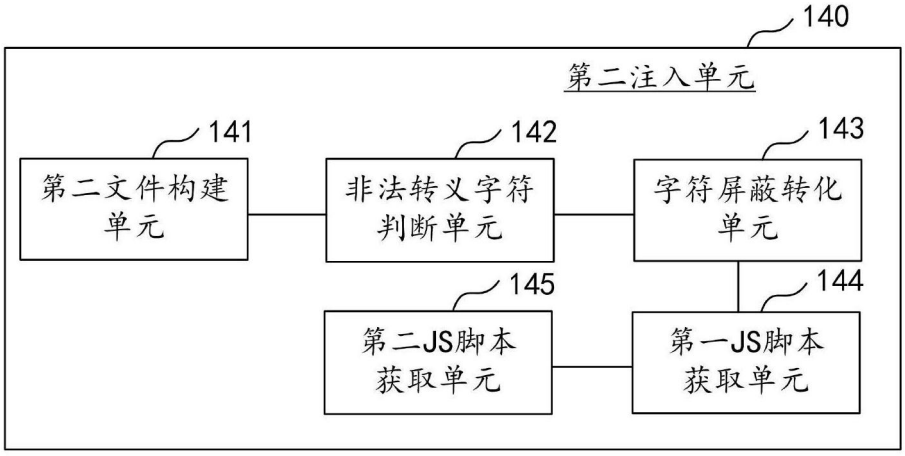

140
第二注入单元
141
第二文件构建单元
142
非法转义字符判断单元
143
字符屏蔽转化单元
145
第二JS脚本获取单元
144
第一JS脚本获取单元

图11

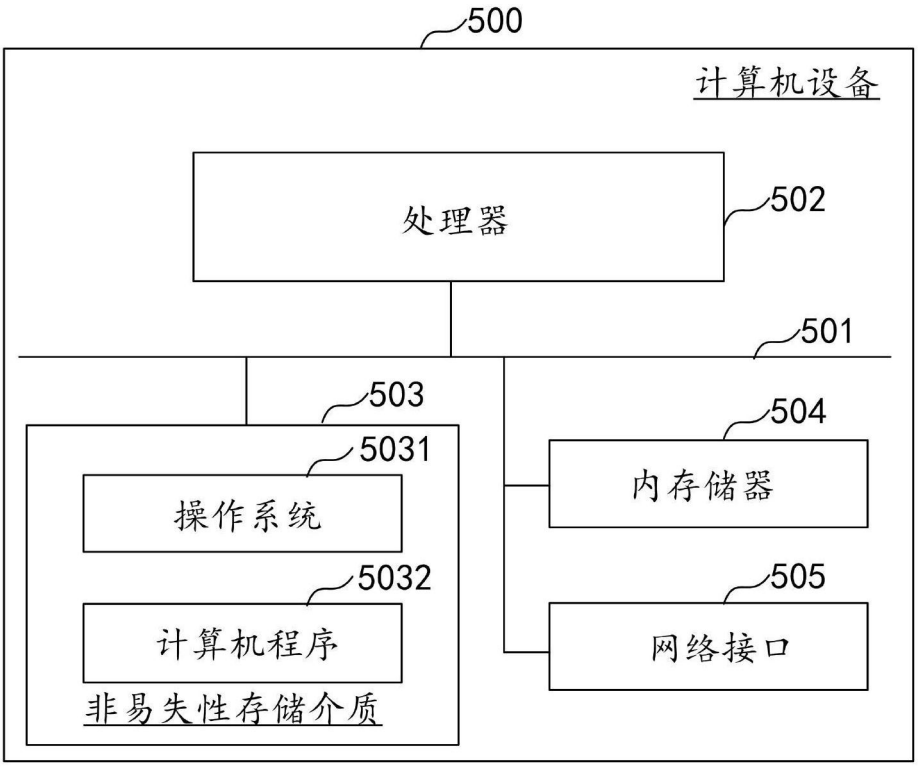

500
计算机设备
处理器
502
501
503
504
5031
操作系统
内存储器
5032
505
计算机程序
网络接口
非易失性存储介质

图12