

一种 web 应用包处理方法、装置及设备

技术领域

本说明书涉及计算机技术领域，尤其涉及一种 web 应用包处理方法、装置及设备。

5

背景技术

随着网络技术发展，越来越多的网页中使用到 web 应用。

通常，业务人员将 web 应用运行时涉及到的资源文件整合在一起，以压缩应用包的形式提供给 web 容器，web 容器加载所述压缩应用包，以供用户访问。

10 压缩应用包中包含有该应用运行所需的资源，例如 jar 包、class 文件、css 文件、图片等等。

基于此，我们需要一种更节省空间的 web 应用包生成方法。

发明内容

15 本说明书实施例提供一种 web 应用包处理方法、装置及设备，用于解决如下问题：以提供一种更节省空间的 web 应用包生成方法。

基于此，本说明书实施例提供一种 web 应用包处理方法，所述方法包括：

获取 web 应用所需使用的资源；

存储所述资源，并确定所述资源对应的资源地址；

20 生成包含所述资源地址的 web 应用包，其中，所述 web 应用包中不包含资源地址对应的资源。

同时，本说明书实施例还提供一种 web 应用包处理方法，所述方法包括：

确定 web 应用包中所包含的资源地址，其中，所述资源地址在生成应用包之前已确定；

25 根据所述资源地址获取对应的资源；

根据所述资源加载所述 web 应用包。

同时,本说明书的实施例还提供一种 web 应用包处理装置,所述装置包括:

确定模块,获取 web 应用所需使用的资源;

存储模块,存储所述资源,并确定所述资源对应的资源地址;

- 5 生成模块,生成包含所述资源地址的 web 应用包,其中,所述 web 应用包中不包含资源地址对应的资源。

同时,本说明书的实施例还提供一种 web 应用包处理装置,所述装置包括:

确定模块,确定 web 应用包中所包含的资源地址,其中,所述资源地址在生成应用包之前已确定;

- 10 获取模块,根据所述资源地址获取对应的资源;

加载模块,根据所述资源加载所述 web 应用包。

本说明书实施例还提供一种 web 应用包处理设备,所述设备包括:

存储器,存储 web 应用包生成程序;

- 15 处理器,接收 web 应用包生成请求后,调用存储器中的 web 应用包生成程序,并执行:

获取 web 应用所需使用的资源;

存储所述资源,并确定所述资源对应的资源地址;

生成包含所述资源地址的 web 应用包,其中,所述 web 应用包中不包含资源地址对应的资源。

- 20 本说明书实施例还提供一种 web 应用包处理设备,所述设备包括:

存储器,存储 web 应用包加载程序;

处理器,接收 web 应用包加载请求后,调用存储器中的 web 应用包加载程序,并执行:

- 25 确定 web 应用包中所包含的资源地址,其中,所述资源地址在生成应用包之前已确定;

根据所述资源地址获取对应的资源;

根据所述资源加载所述 web 应用包。

对应的，本说明书的实施例还提供一种非易失性计算机存储介质，存储有计算机可执行指令，所述计算机可执行指令设置为：

监听 web 应用包生成请求；

5 获取 web 应用所需使用的资源；

存储所述资源，并确定所述资源对应的资源地址；

生成包含所述资源地址的 web 应用包，其中，所述 web 应用包中不包含资源地址对应的资源。

对应的，本说明书的实施例还提供一种非易失性计算机存储介质，存储有计算机可执行指令，所述计算机可执行指令设置为：

监听 web 应用包加载请求；

确定 web 应用包中所包含的资源地址，其中，所述资源地址在生成应用包之前已确定；

根据所述资源地址获取对应的资源；

15 根据所述资源加载所述 web 应用包。

本说明书实施例采用的上述至少一个技术方案能够达到以下有益效果：

对于具体的 web 应用而言，确定该应用所需使用的各种资源，将其中的部分资源上传至存储设备。生成的应用包不包括所述已存储的部分资源，同时，将所述部分资源在存储设备的资源地址，保留在 web 应用包内。

20 本说明书的实施例中，通过将资源上传至存储设备，在生成 web 应用压缩包时，不包括已存储的资源，同时把对应的资源地址留在包内，以便加载时通过地址获取对应资源，使得 web 应用包内减少了大量静态资源，节省了 web 应用包占用的空间。

25 附图说明

图 1 为本说明书实施例提供的架构示意图；

图 2 为本说明书实施例提供的方法流程示意图；

图 3 为本说明书实施例提供的部分方法流程示意图；

图 4 为本说明书实施例提供的方法流程示意图；

图 5 为本说明书实施例提供的一种装置结构示意图；

5 图 6 为本说明书实施例提供的另一种装置结构示意图。

具体实施方式

为使本申请的目的、技术方案和优点更加清楚，下面将结合本申请具体实
10 施例及相应的附图对本申请技术方案进行清楚、完整地描述。显然，所描述的
实施例仅是本申请一部分实施例，而不是全部的实施例。基于本说明书中的实
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，都属于本申请保护的范围。

基于前述内容，所述 web 应用包的处理可包括生成 web 应用包以及加载
15 所述生成的 web 应用包，所述 web 应用可包括通过 web 访问的应用程序，是
web 网页用来完成某种任务的资源的集合。所述 jar 包指的采用 java 语言已写
好的一些类，然后将这些类进行打包所获得的压缩包。所述页面渲染控制文件
可包括 HTML（HyperText Markup Language，超级文本标记语言）文件、JS
（JavaScript）文件、CSS（Cascading Style Sheets）文件等，所述视图资源可
20 包括在显示页面时所采用的声音文件、图片、视频等。

在本说明书的实施例中，所述的 web 应用包处理方法可采用如图 1 所示的
架构，在所述架构中，web 服务器提供各种页面以供用户浏览；存储设备可以
有多种形式，例如硬盘、数据库等，用于存储 web 应用所需采用的一些静态资
源；处理平台可包括各种集群式、分布式的处理设备，在实际应用中，处理设
25 备可包括各种 CPU、服务器等等。处理平台可提供给用户其它相关的业务功能，
例如提供 web 应用开发、打包、部署一站式服务。存储设备 and 处理设备均可以

是分布式、集群式或者是区块链式的组成形式。

下面将基于如图 1 所示的架构，详细说明本说明书的实施例提供的数据处理过程，对于生成 web 应用包的过程，具体包括以下步骤，如图 2 所示：

步骤 S201，获取 web 应用所需使用的资源。

- 5 基于前述内容，在 web 应用中，可包括多个 Servlet、JSP 页面、HTML 文件、页面渲染控制文件、视图文件、jar 包、权限控制文件、流程控制文件等资源。前述资源是根据具体的 web 应用需求而已开发好的。

具体的说，可包括确定 web 应用所需的 jar 包、页面渲染控制文件和/或视图文件等资源。

- 10 步骤 S203，存储所述资源，并确定所述资源对应的资源地址。

在确定了 web 应用所采用的资源之后，将页面渲染控制文件和/或视图文件等静态资源上传至存储设备，获取并记录各资源在存储设备的存储地址（例如资源在存储设备上的 URL 地址），将所述存储地址确定为资源地址。

- 15 步骤 S205，生成包含所述资源地址的 web 应用包，其中，所述 web 应用包中不包含资源地址对应的资源。

具体的说，该步骤包括两个方面：

- 其一，生成的 web 应用包应保留有已存储的资源地址。可通过以下方式来实现：生成包含所述资源地址的资源引用文件，生成包含所述资源引用文件的 web 应用包。即，将获取得到的资源地址写入资源引用文件，在 web 应用包打包的时候将该资源引用文件打包在内，以便加载应用包的时候根据资源地址获取对应资源。
- 20

其二，生成的 web 应用包中不包含资源地址对应的资源。即，对于那些已经上传至存储设备并记录地址的资源，在打包的时候将它们排除在外。在排除前述已经上传的资源时，可以通过人工手动排除，也可以采用如下方式实现：

- 25 根据预定义的规则对指定资源进行打包，生成包含所述资源地址的 web 应用包。所述指定资源包括带有资源地址信息的资源，同时还包括除已存储的

资源之外，web 应用运行时所需要的其它资源。

换言之，在生成 web 应用包时包括：一，将已存储的资源对应的地址包括在内；二，将已存储的资源排除在外。所述两个方面的执行不分先后。

例如，已经确定 web 应用所采用的 jar 包、页面渲染控制文件和/或视图文件等资源，并将页面渲染控制文件和/或视图文件上传至存储设备，同时通过资源引用文件记录了对应的资源地址。在预定义的打包规则中，指定资源包括资源引用文件、jar 包和其他 web 应用所需的资源，而不包括页面渲染控制文件和/或视图文件，从而实现前述技术方案。

本说明书的实施例中，通过将资源上传至存储设备，在生成 web 应用压缩包时，不包括已存储的资源，同时把对应的资源地址留在包内，以便加载时通过地址获取对应资源，使得 web 应用包内减少了大量静态资源，节省了 web 应用包占用的空间。

值得说明的是，在使用 java 开发应用时，源码需要编译成类文件以供计算机识别，在打包时只保留类文件，源码需要手动排除。当采用特定语言（例如 JS 语言）进行 web 应用开发时，由于 JS 文件计算机可以直接识别，在开发过程中无需编译，不会产生类文件。若此时采用前述的预定义的打包模式，还可指定将开发产生的源码文件（即 JS 文件）和其他指定资源一起上传至存储设备，并记录其存储地址，在生成的 web 应用包中将不包含源码文件。

换言之，若采用 JS 编码时，通过前述预定义的打包模式生成 web 应用包，在打包之前没有类文件，生成的包更为节省空间。打包过程中，无需手动排除掉源码文件，更为方便。

作为本说明书的另一种实施例，在确定了 web 应用中所需要使用的 jar 包后，如图 3 所示，所述的 web 应用包生成方法还包括：

步骤 S301，确定 jar 包中的类信息；

步骤 S303，生成所述类信息的配置文件；

步骤 S305，生成包含所述配置文件的 web 应用包。

即类信息的配置文件显示的记录了该 web 应用包中所设计到的类的信息，生成的 web 应用包包含该配置文件，以便在平台上加载该应用包时，平台可以根据配置文件针对该应用包建立专门的类加载器，避免加载多个 web 应用包时发生冲突。

- 5 当生成 web 应用包之后，将其部署在页面服务器上，即此时的 web 服务器需要加载该 web 应用包。作为本说明书的另一种实施例，提供一种 web 应用包处理方法，如图 4 所示，包括：

步骤 S401，确定 web 应用包中所包含的资源地址，其中，所述资源地址在生成 web 应用包之前已确定；

- 10 步骤 S403，根据所述资源地址获取对应的资源；

步骤 S405，根据所述资源加载所述 web 应用包。

即，在加载的过程中页面服务器根据 web 应用包中的资源地址，动态的获取相应的资源（例如获取已存储的页面渲染控制文件和视图文件等），进行页面渲染。

- 15 作为本说明书中的另一种实施例，对于前述的步骤 S405，根据所述资源加载所述 web 应用包，包括：

确定 web 应用包中的类信息的配置文件，其中，所述类信息的配置文件包含 web 应用包中 jar 包的类信息，且，在生成应用包之前已确定；

根据所述类信息的配置文件建立类加载器；

- 20 采用所述类加载器，根据所述资源加载所述 web 应用包。

即，web 服务器上加载该应用包时，可以根据配置文件建立专门的类加载器，避免加载多个 web 应用包时发生冲突。

- 25 基于同样的思路，本说明书实施例还提供一种 web 应用包处理装置，如图 5 所示，包括：

确定模块 501，获取 web 应用所需使用的资源；

存储模块 503, 存储所述资源, 并确定所述资源对应的资源地址;

生成模块 505, 生成包含所述资源地址的 web 应用包, 其中, 所述 web 应用包中不包含资源地址对应的资源。

进一步地, 所述确定模块 501, 确定 web 应用所采用的 jar 包、页面渲染控制文件和/或视图资源。

进一步地, 所述生成模块 505, 确定 jar 包中的类信息, 生成所述类信息的配置文件, 生成包含所述配置文件的 web 应用包。

进一步地, 所述存储模块 503, 存储所述页面渲染控制文件和/或视图资源, 获取所述页面渲染控制文件和/或视图资源的存储地址, 将所述存储地址确定为资源地址。

进一步地, 所述生成模块 505, 根据预定义的规则对指定资源进行打包, 生成包含所述资源地址的 web 应用包。

进一步地, 所述生成模块 505, 生成包含所述资源地址的资源引用文件, 生成包含所述资源引用文件的 web 应用包。

本说明书实施例还提供一种 web 应用包处理装置, 所述装置包括:

确定模块 601, 确定 web 应用包中所包含的资源地址, 其中, 所述资源地址在生成应用包之前已确定;

获取模块 603, 根据所述资源地址获取对应的资源;

加载模块 605, 根据所述资源加载所述 web 应用包。

进一步地, 所述获取模块 603, 根据资源地址获取其对应的页面渲染控制文件和/或视图资源。

进一步地, 所述加载模块 605, 确定 web 应用包中的类信息的配置文件, 其中, 所述类信息的配置文件包含 web 应用包中 jar 包的类信息, 且, 在生成应用包之前已确定; 根据所述类信息的配置文件建立类加载器; 采用所述类加载器加载所述 web 应用包。

对应的，本说明书还提供一种 web 应用包处理设备，所述设备包括：

存储器，存储 web 应用包生成程序；

处理器，接收 web 应用包生成请求后，调用存储器中的 web 应用包生成程序，并执行：

5 获取 web 应用所需使用的资源；

存储所述资源，并确定所述资源对应的资源地址；

生成包含所述资源地址的 web 应用包，其中，所述 web 应用包中不包含资源地址对应的资源。

对应的，本说明书还提供另一种 web 应用包处理设备，所述设备包括：

10 存储器，存储 web 应用包加载程序；

处理器，接收 web 应用包加载请求后，调用存储器中的 web 应用包加载程序，并执行：

确定 web 应用包中所包含的资源地址，其中，所述资源地址在生成应用包之前已确定；

15 根据所述资源地址获取对应的资源；

根据所述资源加载所述 web 应用包。

基于同样的发明思路，本申请实施例还提供了对应的一种非易失性计算机存储介质，存储有计算机可执行指令，所述计算机可执行指令设置为：

20 监听 web 应用包生成请求；

获取 web 应用所需使用的资源；

存储所述资源，并确定所述资源对应的资源地址；

生成包含所述资源地址的 web 应用包，其中，所述 web 应用包中不包含资源地址对应的资源。

25 对应的，本说明书的实施例还提供另一种非易失性计算机存储介质，存储有计算机可执行指令，所述计算机可执行指令设置为：

监听 web 应用包加载请求；

确定 web 应用包中所包含的资源地址，其中，所述资源地址在生成应用包之前已确定；

根据所述资源地址获取对应的资源；

5 根据所述资源加载所述 web 应用包。

本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于装置、设备和介质类实施例而言，由于其基本相似于方法实施
10 例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可，这里就不再一一赘述。

上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下，在权利要求书中记载的动作或步骤或模块可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外，在附图中描
15 绘的过程不一定要要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中，多任务处理和并行处理也是可以的或者可能是有利的。

在 20 世纪 90 年代，对于一个技术的改进可以很明显地区分是硬件上的改进（例如，对二极管、晶体管、开关等电路结构的改进）还是软件上的改进（对于方法流程的改进）。然而，随着技术的发展，当今的很多方法流程的改进已经可以视为硬件电路结构的直接改进。设计人员几乎都通过将改进的方法流程
20 编程到硬件电路中来得到相应的硬件电路结构。因此，不能说一个方法流程的改进就不能用硬件实体模块来实现。例如，可编程逻辑器件（Programmable Logic Device, PLD）（例如现场可编程门阵列（Field Programmable Gate Array, FPGA））就是这样一种集成电路，其逻辑功能由用户对器件编程来确定。由设计
25 人员自行编程来把一个数字系统“集成”在一片 PLD 上，而不需要请芯片制造厂商来设计和制作专用的集成电路芯片。而且，如今，取代手工地制作集成电

路芯片，这种编程也多半改用“逻辑编译器（logic compiler）”软件来实现，它与程序开发撰写时所用的软件编译器相类似，而要编译之前的原始代码也得用特定的编程语言来撰写，此称之为硬件描述语言（Hardware Description Language, HDL），而 HDL 也并非仅有一种，而是有许多种，如 ABEL（Advanced Boolean Expression Language）、AHDL（Altera Hardware Description Language）、Confluence、CUPL（Cornell University Programming Language）、HDCal、JHDL（Java Hardware Description Language）、Lava、Lola、MyHDL、PALASM、RHD（Ruby Hardware Description Language）等，目前最普遍使用的是 VHDL（Very-High-Speed Integrated Circuit Hardware Description Language）与 Verilog。

10 本领域技术人员也应该清楚，只需要将方法流程用上述几种硬件描述语言稍作逻辑编程并编程到集成电路中，就可以很容易得到实现该逻辑方法流程的硬件电路。

控制器可以按任何适当的方式实现，例如，控制器可以采取例如微处理器或处理器以及存储可由该（微）处理器执行的计算机可读程序代码（例如软件或固件）的计算机可读介质、逻辑门、开关、专用集成电路（Application Specific Integrated Circuit, ASIC）、可编程逻辑控制器和嵌入微控制器的形式，控制器的例子包括但不限于以下微控制器：ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20 以及 Silicone Labs C8051F320，存储器控制器还可以被实现为存储器的控制逻辑的一部分。本领域技术人员也知道，除了以纯计算机可读程序代码方式实现控制器以外，完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件，而对其内包括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至，可以将用于实现各种功能的装置视为既可以是实现方法的软件模块又可以是硬件

20 部件内的结构。

25

上述实施例阐明的系统、装置、模块或单元，具体可以由计算机芯片或实

体实现，或者由具有某种功能的产品来实现。一种典型的实现设备为计算机。具体的，计算机例如可以为个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任何设备的组合。

- 5 为了描述的方便，描述以上装置时以功能分为各种单元分别描述。当然，在实施本说明书的实施例时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

本领域内的技术人员应明白，本发明的实施例可提供为方法、系统、或计算机程序产品。因此，本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

10

本发明是参照根据本发明实施例的方法、设备（系统）、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

15

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

20

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个

25

流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的步骤。

在一个典型的配置中，计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器
5 (RAM)和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存
10 (PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他
15 磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体(transitory media)，如调制的数据信号和载波。

还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、
20 方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

本领域技术人员应明白，本说明书中一个或多个的实施例可提供为方法、系统或计算机程序产品。因此，本说明书的实施例可采用完全硬件实施例、完
25 全软件实施例或结合软件和硬件方面的形式。而且，本说明书的实施例可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但

不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

本说明书的实施例可以在由计算机执行的计算机可执行指令的一般上下文中描述,例如程序模块。一般地,程序模块包括执行特定事务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本说明书的实施例,在这些分布式计算环境中,由通过通信网络而被连接的远程处理设备来执行事务。在分布式计算环境中,程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

本说明书中的各个实施例均采用递进的方式描述,各个实施例之间相同相似的部分互相参见即可,每个实施例重点说明的都是与其他实施例的不同之处。尤其,对于系统实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

以上所述仅为本说明书的实施例而已,并不用于限制本申请。对于本领域技术人员来说,本说明书的实施例可以有各种更改和变化。凡在本说明书的实施例的精神和原理之内所作的任何修改、等同替换、改进等,均应包含在本申请的权利要求范围之内。