



(12)发明专利申请

(10)申请公布号 CN 110740048 A

(43)申请公布日 2020.01.31

(21)申请号 201810791616.3

(22)申请日 2018.07.18

(71)申请人 广东亿迅科技有限公司

地址 510000 广东省广州市天河区中山大道109号1号楼13层

(72)发明人 陈锐标 赵善浮 田欢 温斯嘉

(74)专利代理机构 深圳市精英专利事务所  
44242

代理人 葛勤

(51) Int. Cl.

H04L 12/24(2006.01)

H04L 29/08(2006.01)

H04L 29/06(2006.01)

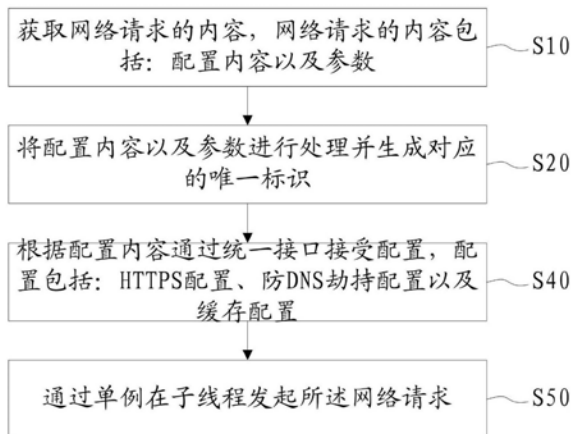
权利要求书2页 说明书5页 附图6页

(54)发明名称

一种基于iOS网络的网络请求处理方法及装置

(57)摘要

本发明涉及一种基于iOS网络的网络请求处理方法,其中所述方法包括:获取网络请求的内容,所述网络请求的内容包括:配置内容以及参数;将所述配置内容以及参数进行处理并生成对应的唯一标识;根据所述配置内容通过统一接口接受配置,所述配置包括:HTTPS配置、防DNS劫持配置以及缓存配置;通过单例在子线程发起所述网络请求。本发明通过配置类的方式快速添加HTTPS配置以及防止DNS劫持配置,免去重复繁琐的操作。并且,本发明根据网络请求阶段封装成不同的模块,并提供统一的API。此外,通过使用单例和子线程的方式进行请求,使得应用在运行期间只会存在一个网络请求对象,有效地避免了多个不同请求对象可能导致的请求混乱。



1. 一种基于iOS网络的网络请求处理方法,其特征在于,所述基于iOS网络的网络请求处理方法包括按顺序如下步骤:

S10、获取网络请求的内容,所述网络请求的内容包括:配置内容以及参数;

S20、将所述配置内容以及参数进行处理并生成对应的唯一标识;

S40、根据所述配置内容通过统一接口接受配置,所述配置包括:HTTPS配置、防DNS劫持配置以及缓存配置;

S50、通过单例在子线程发起所述网络请求。

2. 如权利要求1所述的基于iOS网络的网络请求处理方法,其特征在于,所述步骤S20之后,还包括:

S302、检测请求状态列表;

S304、根据所述的标识判断是否存在同样的请求,若不存在则更新所述请求状态列表,若存在则终止所述网络请求。

3. 如权利要求2所述的基于iOS网络的网络请求处理方法,其特征在于,所述步骤S50之后,还包括:

S60、接收所述网络请求返回的数据,并将所述数据转换成通用的报文格式。

4. 如权利要求3所述的基于iOS网络的网络请求处理方法,其特征在于,所述步骤S60之后还包括:

S70、将所述网络请求返回的数据进行缓存,并在请求时根据缓存配置来读取缓存。

5. 如权利要求1-4任一项所述的基于iOS网络的网络请求处理方法,其特征在于,所述步骤S50之后还包括:

S80、对所述基于iOS网络的网络请求处理方法中出现的差错进行检测,并对检测到的差错进行反馈。

6. 一种基于iOS网络的网络请求处理装置,其特征在于,所述基于iOS网络的网络请求处理装置包括:

获取模块,用于获取网络请求的内容,所述网络请求的内容包括:配置内容以及参数;

生成模块,用于将所述配置内容以及参数进行处理并生成对应的唯一标识;

配置模块,用于根据所述配置内容通过统一接口接受配置,所述配置包括:HTTPS配置、防DNS劫持配置以及缓存配置;

发起模块,用于通过单例在子线程发起所述网络请求。

7. 如权利要求6所述的基于iOS网络的网络请求处理装置,其特征在于,所述基于iOS网络的网络请求处理装置还包括:

状态检测模块,用于检测请求状态列表;

判断模块,用于根据所述的标识判断是否存在同样的请求,若不存在则更新所述请求状态列表,若存在则终止所述网络请求。

8. 如权利要求7所述的基于iOS网络的网络请求处理装置,其特征在于,所述基于iOS网络的网络请求处理装置还包括:

报文模块,用于接收所述网络请求返回的数据,并将所述数据转换成通用的报文格式。

9. 如权利要求8所述的基于iOS网络的网络请求处理装置,其特征在于,所述基于iOS网络的网络请求处理装置还包括:

缓存模块,用于将所述网络请求返回的数据进行缓存,并在请求时根据缓存配置来读取缓存。

10.如权利要求6-9任一项所述的基于iOS网络的网络请求处理装置,其特征在于,所述基于iOS网络的网络请求处理装置还包括:

差错反馈模块,用于对所述基于iOS网络的网络请求处理方法中出现的差错进行检测,并对检测到的差错进行反馈。

## 一种基于iOS网络的网络请求处理方法及装置

### 技术领域

[0001] 本发明涉及计算机网络技术领域,尤其涉及一种基于iOS网络的网络请求处理方法及装置。

### 背景技术

[0002] 移动互联网的世界中,移动应用已经深入到广大群众的方方面面。每天通过移动应用发出的网络请求数以亿计,无数移动开发者为保障这些请求能稳定、高效、安全地传输而劳动着。

[0003] 尽管绝大多数的请求在开发时都得到了适当的处理,但是,使用系统提供的API进行开发时,仍然存在种种问题。首先,系统API封装层次低,使用复杂。开发人员需要比较深入地了解网络相关API才能进行开发,对开发人员的水平要求较高。其次,请求安全方面配置繁琐。如今,用户对数据安全的要求越来越高。但是HTTPS网络请求的配置却非常繁琐,对普通开发人员来说,使用非常不便。此外,图片、小文件等类型的缓存处理。普通的网络请求显然不会对这些内容进行缓存处理,而实际上对这些类型的内容进行缓存处理能大大地提高请求效率和节省流量。

[0004] 综上所述,直接使用系统API进行网络请求配置复杂,系统API使用繁琐、网络安全配置复杂、没有完善的缓存机制、不同项目的网络请求模块难以复用,传承性差等,对开发人员的水平要求高,不利于项目的进行,现有网络请求框架虽然提供了不少便捷的方式,但也存在各自的问题。

[0005] 有鉴于此,有必要提出对目前的基于iOS网络的网络请求处理技术进行进一步的改进。

### 发明内容

[0006] 为解决上述至少一技术问题,本发明的主要目的是提供一种基于iOS网络的网络请求处理方法。

[0007] 为实现上述目的,本发明采用的一个技术方案为:提供一种基于iOS网络的网络请求处理方法,所述基于iOS网络的网络请求处理方法包括按顺序如下步骤:

[0008] S10、获取网络请求的内容,所述网络请求的内容包括:配置内容以及参数;

[0009] S20、将所述配置内容以及参数进行处理并生成对应的唯一标识;

[0010] S40、根据所述配置内容通过统一接口接受配置,所述配置包括:HTTPS配置、防DNS劫持配置以及缓存配置;

[0011] S50、通过单例在子线程发起所述网络请求。

[0012] 其中,所述步骤S20之后,还包括:

[0013] S302、检测请求状态列表;

[0014] S304、根据所述的标识判断是否存在同样的请求,若不存在则更新所述请求状态列表,若存在则终止所述网络请求。

- [0015] 其中,所述步骤S50之后,还包括:
- [0016] S60、接收所述网络请求返回的数据,并将所述数据转换成通用的报文格式。
- [0017] 其中,所述步骤S60之后还包括:
- [0018] S70、将所述网络请求返回的数据进行缓存,并在请求时根据缓存配置来读取缓存。
- [0019] 其中,上述步骤S50之后还包括:
- [0020] S80、对所述基于iOS网络的网络请求处理方法中出现的差错进行检测,并对检测到的差错进行反馈。
- [0021] 为实现上述目的,本发明采用的另一个技术方案为:提供一种基于iOS网络的网络请求处理装置,所述基于iOS网络的网络请求处理装置包括:
- [0022] 获取模块,用于获取网络请求的内容,所述网络请求的内容包括:配置内容以及参数;
- [0023] 生成模块,用于将所述配置内容以及参数进行处理并生成对应的唯一标识;
- [0024] 配置模块,用于根据所述配置内容通过统一接口接受配置,所述配置包括:HTTPS配置、防DNS劫持配置以及缓存配置;
- [0025] 发起模块,用于通过单例在子线程发起所述网络请求。
- [0026] 其中,所述基于iOS网络的网络请求处理装置还包括:
- [0027] 状态检测模块,用于检测请求状态列表;
- [0028] 判断模块,用于根据所述的标识判断是否存在同样的请求,若不存在则更新所述请求状态列表,若存在则终止所述网络请求。
- [0029] 其中,所述基于iOS网络的网络请求处理装置还包括:
- [0030] 报文模块,用于接收所述网络请求返回的数据,并将所述数据转换成通用的报文格式。
- [0031] 其中,所述基于iOS网络的网络请求处理装置还包括:
- [0032] 缓存模块,用于将所述网络请求返回的数据进行缓存,并在请求时根据缓存配置来读取缓存。
- [0033] 其中,上述基于iOS网络的网络请求处理装置还包括:
- [0034] 差错反馈模块,用于对所述基于iOS网络的网络请求处理方法中出现的差错进行检测,并对检测到的差错进行反馈。
- [0035] 本发明提出的一种基于iOS网络的网络请求处理方法,通过配置类的方式快速添加HTTPS配置以及防止DNS劫持配置,免去重复繁琐的操作。并且,本发明根据网络请求阶段封装成不同的模块,并提供统一的API,有利于后续维护和扩展。此外,通过使用单例和子线程的方式进行请求,使得应用在运行期间只会存在一个网络请求对象,有效地避免了多个不同请求对象可能导致的请求混乱。

## 附图说明

[0036] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅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

根据这些附图示出的结构获得其他的附图。

[0037] 图1为本发明一实施例的一种基于iOS网络的网络请求处理方法的方法流程图；

[0038] 图2为本发明另一实施例的一种基于iOS网络的网络请求处理方法的方法流程图；

[0039] 图3为本发明另一实施例的一种基于iOS网络的网络请求处理方法的方法流程图；

[0040] 图4为本发明另一实施例的一种基于iOS网络的网络请求处理方法的方法流程图；

[0041] 图5为本发明一实施例的一种基于iOS网络的网络请求处理装置的模块方框图；

[0042] 图6为本发明另一实施例的一种基于iOS网络的网络请求处理装置的模块方框图。

[0043] 本发明目的的实现、功能特点及优点将结合实施例,参照附图做进一步说明。

## 具体实施方式

[0044] 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,所描述的实施例仅仅是本发明的一部分实施例,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作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。

[0045] 需要说明,本发明中涉及“第一”、“第二”等的描述仅用于描述目的,而不能理解为指示或暗示其相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”的特征可以明示或者隐含地包括至少一个该特征。另外,各个实施例之间的技术方案可以相互结合,但是必须是以本领域普通技术人员能够实现为基础,当技术方案的结合出现相互矛盾或无法实现时应当认为这种技术方案的结合不存在,也不在本发明要求的保护范围之内。

[0046] 请参照图1,在本发明实施例中,提供一种基于iOS网络的网络请求处理方法,所述基于iOS网络的网络请求处理方法包括按顺序如下步骤:

[0047] S10、获取网络请求的内容,所述网络请求的内容包括:配置内容以及参数;

[0048] S20、将所述配置内容以及参数进行处理并生成对应的唯一标识;

[0049] S40、根据所述配置内容通过统一接口接受配置,所述配置包括:HTTPS配置、防DNS劫持配置以及缓存配置;

[0050] S50、通过单例在子线程发起所述网络请求。

[0051] 在本实施例中,所述的基于iOS网络的网络请求处理方法运用于App中。具体地,首先获取网络请求的内容,所述网络请求的内容包括:配置内容以及参数。然后,将所述配置内容以及参数进行处理并生成对应的唯一标识,用来标识当前请求,防止同样输入的多次请求。根据所述配置内容通过统一接口接受配置,所述配置包括:HTTPS配置、防DNS劫持配置以及缓存配置,方便管理。最后,通过使用单例和子线程运行的模式达到高效、一致的网络请求效果,不影响应用其他部分的运行。

[0052] 综上,本实施例中提出提出的一种基于iOS网络的网络请求处理方法,通过配置类的方式快速添加HTTPS配置以及防止DNS劫持配置,免去重复繁琐的操作。并且,本发明根据网络请求阶段封装成不同的模块,并提供统一的API,有利于后续维护和扩展。此外,通过使用单例和子线程的方式进行请求,使得应用在运行期间只会存在一个网络请求对象,有效地避免了多个不同请求对象可能导致的请求混乱。

[0053] 参考图2,在一具体实施例中,所述步骤S20之后,还包括:

[0054] S302、检测请求状态列表；

[0055] S304、根据所述的标识判断是否存在同样的请求，若不存在则更新所述请求状态列表，若存在则终止所述网络请求。

[0056] 具体地，首先，检测请求状态列表，然后，根据所述的标识判断是否存在同样的请求，若不存在则更新所述请求状态列表，若存在则终止所述网络请求。

[0057] 本实施例实现了通过检测以及更新请求状态列表，根据唯一的标识判断是否存在同样的网络请求，让用户可以随时检查每个请求的当前状态，有效地防止同一网络请求多次发起和段时间内重复发起的尴尬情况。

[0058] 参考图3，在一具体实施例中，所述步骤S50之后，还包括：

[0059] S60、接收所述网络请求返回的数据，并将所述数据转换成通用的报文格式。

[0060] 具体地，在本实施例中，接受网络请求返回的数据，并尝试转换成通用的报文，内置XML格式和JSON格式，并提供开放接口，用户可方便地扩展使用新的报文格式。

[0061] 本实施例实现了为用户提供了开放式的报文解析方案。内置的报文解析模块支持JSON、XML两种报文的解析，满足目前大部分的需求。

[0062] 参考图4，在一具体实施例中，所述步骤S60之后还包括：

[0063] S70、将所述网络请求返回的数据进行缓存，并在请求时根据缓存配置来读取缓存。

[0064] 具体地，对请求返回数据进行缓存，并且在请求时根据缓存配置来读取缓存，对于内容变化的内容有着明显的效率提升，降低流量使用。

[0065] 本实施例实现了对图片、小文件等类型的缓存处理，通过对这些类型的内容进行缓存处理能大大地提高请求效率和节省流量。

[0066] 在一具体实施例中，上述步骤S50之后还包括：

[0067] S80、对所述基于iOS网络的网络请求处理方法中出现的差错进行检测，并对检测到的差错进行反馈。

[0068] 具体地，本实施例中定义了一套完整的差错反馈机制，网络请求过程中出现的错误都会流转到差错处理模块，对检测到的差错进行反馈，使得用户可根据提示进行操作。

[0069] 参考图5，本实施例提供一种基于iOS网络的网络请求处理装置，所述基于iOS网络的网络请求处理装置包括：

[0070] 获取模块10，用于获取网络请求的内容，所述网络请求的内容包括：配置内容以及参数；

[0071] 生成模块20，用于将所述配置内容以及参数进行处理并生成对应的唯一标识；

[0072] 配置模块40，用于根据所述配置内容通过统一接口接受配置，所述配置包括：HTTPS配置、防DNS劫持配置以及缓存配置；

[0073] 发起模块50，用于通过单例在子线程发起所述网络请求。

[0074] 其中，在一具体实施例中，所述基于iOS网络的网络请求处理装置还包括：

[0075] 状态检测模块302，用于检测请求状态列表；

[0076] 判断模块304，用于根据所述的标识判断是否存在同样的请求，若不存在则更新所述请求状态列表，若存在则终止所述网络请求。

[0077] 本实施例中，通过配置类的方式快速添加HTTPS配置以及防止DNS劫持配置，免去

重复繁琐的操作。并且,本发明根据网络请求阶段封装成不同的模块,并提供统一的API,有利于后续维护和扩展。此外,通过使用单例和子线程的方式进行请求,使得应用在运行期间只会存在一个网络请求对象,有效地避免了多个不同请求对象可能导致的请求混乱。

[0078] 在一具体实施例中,所述基于iOS网络的网络请求处理装置还包括:

[0079] 报文模块60,用于接收所述网络请求返回的数据,并将所述数据转换成通用的报文格式。

[0080] 参考图6,在一具体实施例中,所述基于iOS网络的网络请求处理装置还包括:

[0081] 缓存模块70,用于将所述网络请求返回的数据进行缓存,并在请求时根据缓存配置来读取缓存。

[0082] 在一具体实施例中,上述基于iOS网络的网络请求处理装置还包括:

[0083] 差错反馈模块80,用于对所述基于iOS网络的网络请求处理方法中出现的差错进行检测,并对检测到的差错进行反馈。

[0084] 以上所述仅为本发明的优选实施例,并非因此限制本发明的专利范围,凡是在本发明的发明构思下,利用本发明说明书及附图内容所作的等效结构变换,或直接/间接运用在其他相关的技术领域均包括在本发明的专利保护范围内。



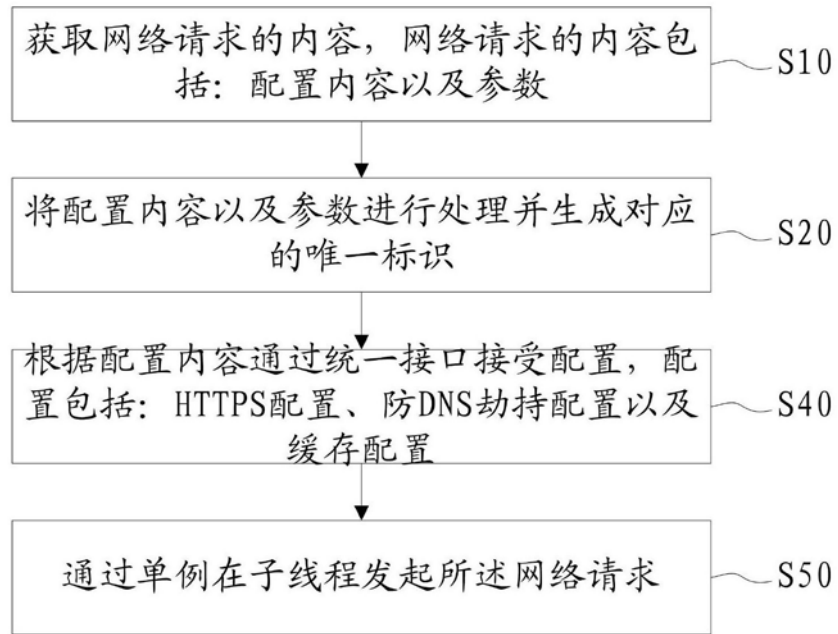


图1



图2

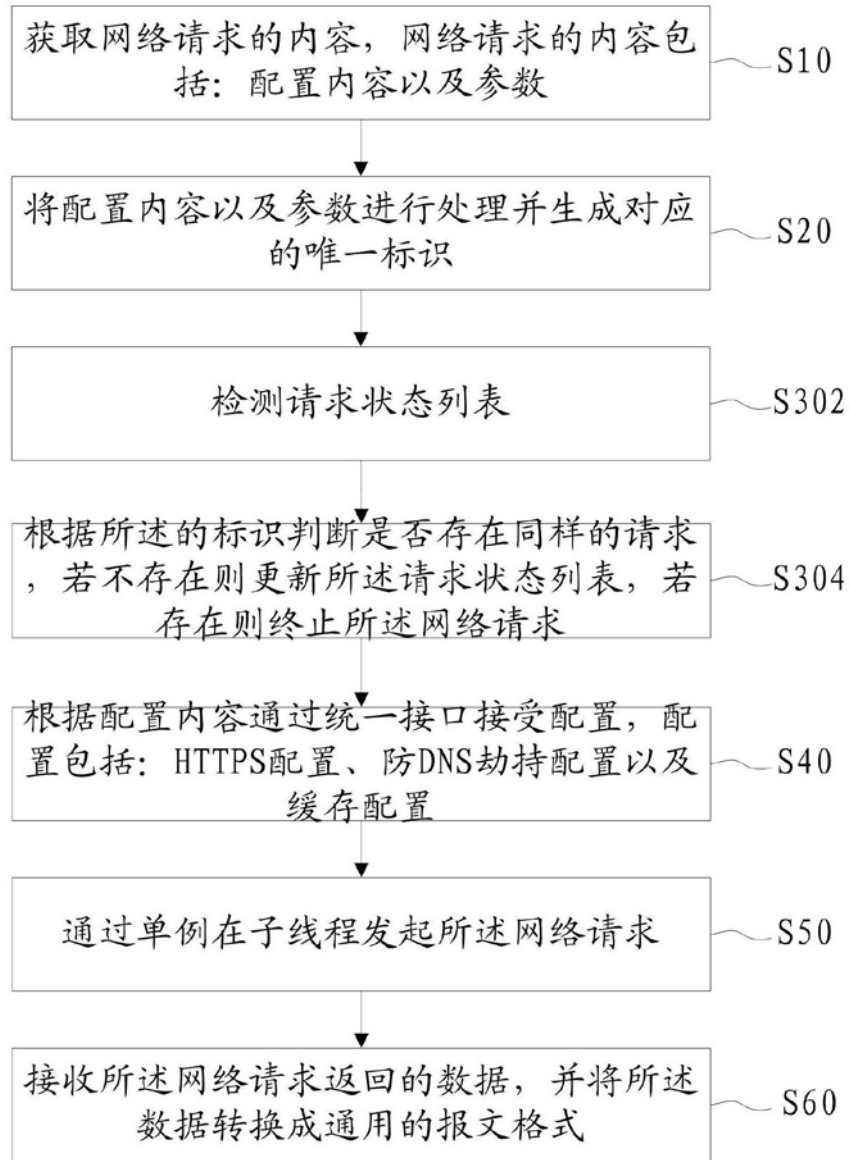


图3



图4

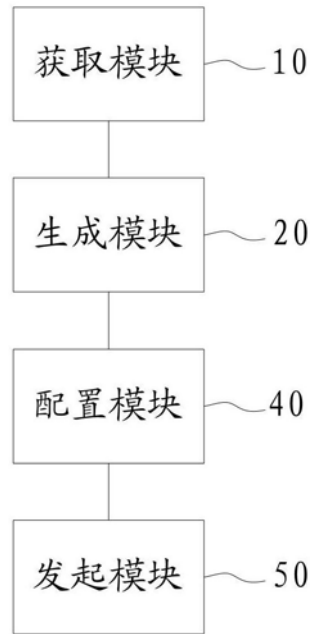


图5

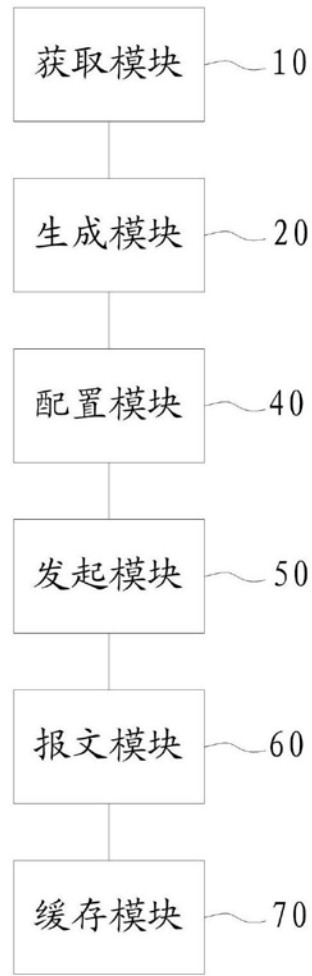


图6