＜Easy Opensource＞

**软件需求说明书**

作 者： 张灵箫

完成日期：

签 收 人：

签收日期：

修改情况记录：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修改批准人 | 修改人 | 安装日期 | 签收人 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 引言

## 编写目的

该需求说明书旨在明确Easy Opensource项目的功能和性能需求，并随时记录需求变化，给项目开发、使用、维护人员以参考。该说明书的读者为项目的开发、维护、审核人员。

## 定义

**开源项目资源：**某开源项目本身所包含的代码、文档以及其他与该项目相关的官方以及非官方的数据，包括但不限于项目的版本库、缺陷库、邮件列表，以及相关的博客、论坛等。

# 项目概述

## 产品概述

开源项目在软件工程学术研究领域发挥着非常重要的作用，由于开源项目拥有免费、开放源代码、项目领域覆盖面广等特点，很容易在开源项目社区积累出大量的**开源项目资源**。这些资源是许多软件工程学术研究课题的重要研究对象和数据来源。然而开源项目资源数据种类繁多，数据量庞大，且分布在互联网当中，对这些数据的抓取和整理工作往往需要耗费研究人员大量的精力。

Easy Opensource系统（一下简称EOS系统）旨在简化研究人员从互联网抓取、整理**开源软件资源**这一步骤，从而提高研究人员利用**开源项目资源**进行学术研究的效率。具体来说，本项目希望能提供一套完整的，用户可配置的，自动的解决方法，根据研究人员的需求，自动从互联网搜集，抓取，并以合适的机制存储**开源项目资源**，为研究人员研究提供可靠、全面、容易获取的资料，为软件工程的研究工作，特别是知识驱动的软件开发研究提供良好的数据基础。

## 产品功能

我们通过下面的用况图定义出EOS系统边界和外部环境的交互：



**与系统交互的外部元素包括：**

1. **用户**

本系统的用户为具有专业知识的软件工程研究人员

1. **外部应用程序**

利用开源软件项目资源为数据源的程序可以使用本系统的借口得到数据

1. **Internet**

本系统从互联网上得到开源项目的相关资源

1. **本地数据**

本系统将得到的数据存放在本地的磁盘系统中并且对其进行相应的解析

我们将系统分为三个子系统，每个子系统包含的用况有

* **资源获取子系统：**
  + **配置资源信息**
  + **抓取资源**
* **资源解析子系统：**
  + **获取资源数据**
  + **解析资源数据**
* **资源存储子系统：**
  + **存储管理本地资源**

## 数据范围

本节我们将给出EOS系统所关心的开源项目数据的范围和定义

首先，对于每一个开源项目，系统需要一个数据模型来描述其基本信息，描述一个开源项目基本信息的数据包括：

* 项目名称
* 项目主页
* 编程语言
* 项目描述
* 项目类型

接下来，EOS系统需要抓取和分析的开源项目资源分为以下类别：

* 文档
* 版本库
* Issue Tracker
* 邮件列表
* 其他

对每一类资源，一下分别给出EOS项目关心的资源数据

* 文档
  + 文档类型（包括Manuel，faq，wiki等）
  + 文档标题
  + 文档内容
  + 文档版本
  + 更新时间
* 版本库
  + 版本库类型（包括SVN/CVS/git等）
  + 版本库地址
  + 代码文件内容
  + 更新时间
* Issue Tracker
  + 类型（bugzilla/jira等）
  + Issue tracker主页地址
  + 每条Issue的具体信息
  + 更新时间
* 邮件列表
  + 邮件列表类型
  + 抓取地址
  + 邮件内容
  + 更新时间
* 其他
  + 包括论坛，博客等互联网上散步的相关内容，具体数据依据内容而定。

# 具体需求

本节我们将给出EOS项目每个子系统的每个用况的具体描述，从而明确项目的具体需求。

## 资源获取子系统

资源获取子系统和用户以及互联网的交互过程如下：



用户

Internet

接下来给出两个用况的需求细节

### 配置资源信息

开源软件项目包罗万象，开源项目资源更是包含了多种数据类型，对不同类型的资源，系统需要以不同的方式进行抓

取和存储。我们希望用户能通过我们的系统配置不同资源的属性和管理方式，方便用户对不同的资源进行统一的管理。之后，我们寻求实现对资源配置的自动化。下面我们详细叙述资源信息配置时用户和系统的交互过程

|  |  |  |
| --- | --- | --- |
| **时间线** | **用户行为** | **系统行为** |
|  | 输入开源项目名称 |  |
|  |  | 查询后台数据确定系统中是否已有该项目 |
| **If系统中不含开源项目** |  |  |
|  |  | 用户提示系统给出项目主页（可自动化） |
|  | 用户编辑，确认项目主页URL |  |
|  |  | 系统给出项目基本信息（可自动化） |
|  | 用户编辑，确认项目基本信息 |  |
|  |  | 系统给出需要爬取的资源的类型的缺省项（包括文档，版本库，缺陷库，邮件列表）及其爬取地址（可自动化） |
|  | 用户编辑，确认爬取地址 |  |
|  |  | 系统从互联网抓取资源，存于本地存储系统 |
|  |  | 系统给出本地存储的开源项目基本信息以及资源获取接口 |
|  | 用户通过接口查看存储在本地的资源 |  |

### 抓取资源

抓取数据是本项目的重要功能。由于这些数据将用于以后的进一步研究工作，所以抓取数据功能需要满足以下需求：

#### 文档

1. 输入

包含项目官方文档入口地址的一个或多个URL

1. 输出

尽可能多的官方文档，包括但不限于教程、用户手册、开发者手册、FAQ、wiki等。这些文档的链接在官方网站域名范围之内，抓取的数据能够覆盖90%以上的官方文档。

用户能够浏览所有文档的标题，并且按照标题获取不含其他信息的纯文本的文档内容。

#### 代码库

1. 输入

项目代码库（版本库）地址

1. 输出

项目代码库文件，包含各个历史版本的完整代码信息

用户能够浏览某个历史版本中某个具体文件包含的代码

#### 缺陷库

1. 输入

项目缺陷库（Issue tracker）地址

1. 输出

项目的完整缺陷记录

用户能够浏览某条缺陷记录的具体信息，包括缺陷描述和所有的相关评论等

#### 邮件列表

1. 输入

包含项目邮件列表入口地址的一个或多个URL

1. 输出

项目的全部邮件列表

用户能够浏览邮件的主题和内容

#### 其他

开源项目资源处官方资源外还包括大量非官方资源，如（非官方）论坛、博客、问答网站等。具体的输入输出要求视资源类型而定。

## 资源解析子系统

### 获取资源数据

对每种资源类型的资源，首先在资源未被进一步解析的情况下（raw data），提供较简单的数据获取接口。

#### 文档

可获得所有文档的标题列表，并且根据标题获得文档原始内容

#### 代码库

可获得某个版本的所有代码

#### 缺陷库

可以获得每条缺陷的具体信息

#### 邮件列表

可以获得邮件标题的列表和邮件的具体内容

### 解析资源数据

## 资源存储子系统

### 数据类型

从互联网分布的开源项目资源种类繁多，从数据类型上看，开源项目资源的原始数据有以下两种：

* **文件**

该类型资源本身以一个或多个文件的形式存在与远程数据端，以该类型存在的资源主要包括文档（html、pdf等格式），版本库（版本库由多个大文件构成），邮件列表等。该类型数据可以直接通过文件链接以http请求的方式获得整个文件，并保存在本地的文件系统中。

对于大部分文档，用户可以直接获取这些文件进行查看（html，pdf等）。而版本库或者邮件列表等文件则需要相应的解析程序才能查看其文件内部信息。

* **结构化数据**

该类型资源由多条结构化数据组成（主要代表有issue tracker）。在远程数据端往往以数据库或其他结构化存储方式存在。用户不能直接获取，需要通过数据存储方提供的API结构方能得到数据。

在本地需要用数据库或其他的结构化方法来存储类似的数据

### 资源存储与整理

## 属性

### 可扩展性

开源软件项目资源，特别是非官方的数据种类繁多，并且在不断增加。为了保证系统的长期可用性，要保证系统具有良好的可扩展性，使开发者能够为其他类型的数据编写抓取模块并且集成到系统中。具体包括：

* **良好的数据模型**

必须对不同类型的开源项目资源数据抽象出统一的数据模型，模块接口和后台数据存储在该数据模型上实现，以保证在需要抓取全新类型的数据时能够按该数据模型对其进行抽象，供编程人员使用。

* **统一的模块接口**

必须为全新类型的数据抓取模块提供统一的接口，保证编程人员在实现该接口后能够将新模块集成到系统中，而不影响已有的系统功能。

### 支持分布式存储与计算

开源软件项目资源数据量庞大，单服务器存储方式会让数据难以管理，我们希望系统能够支持以分布式文件系统（如HDFS）在为底层设施，使得到的数据分布在多台服务器上。

另一方面，本项目抓取得到的数据具在数据挖掘等方向上具有很大的研究价值，而相关研究往往需要在海量数据上运行较为复杂的算法。我们希望抓取到的数据能够储存在一个支持分布式计算的平台上，用以支持后续的研究工作。

综上所述，系统抓取到的数据需要存储在一个支持分布式存储和计算的平台上，以便在数据量不断增加时，可以相应的扩展系统的存储和计算能力。

### 可增量存储

各种类型的开源项目资源都是在原有资源的基础上不断增长和进化的，当系统需要再次抓取已经存在于本地的开源项目资源时，需要区分未发生变化的数据和增加或者改变的数据。对未变化的数据，系统不应该再次抓取导致数据重复。同时系统需要识别出增加的数据和发生改变的数据，并且抓取这些数据并相应的改变或增加已经存储在本地的开源项目资源。