

NodeJS需求评审

SY1606117 李岳樵

SY1606118 温元祯

SY1606413 谭伟良

PT1600283 王春柳



CONTENT



1 | 业务需求

2 | 工作重点

3 | 功能需求

4 | 非功能需求



1 | 业务需求

构建高性能Web服务器

在Web领域，大多数的编程语言需要专门的Web服务器作为容器，如ASP需要IIS作为服务器，PHP需要搭载Apache，JSP需要tomcat服务器。Node需要做的是构建一个高性能的，事件驱动，非阻塞I/O的Web服务器。

2009年,Ryan Dahl最初的目标，就是写一个基于事件驱动的、非阻塞I/O的web服务器，提供Apache等服务器之外的选择。

使用一门门槛低，符合事件驱动的语言实现

Javascript开发门槛低，在后端部分一直没有市场，可以说历史包袱为零，为其导入非阻塞I/O库没有额外阻力，另外，Javascript在浏览器中有广泛的事件驱动方面的应用，暗合Node基于事件驱动的需求。



2 | 工作重点

异步并发控制

异步编程解决的问题无外乎是保持异步的性能优势，提升编程体验，但仍然会有别的问题，那就是并发量太大导致下层服务器崩溃，所以需要过载保护。

解决方案初步

- 1.通过一个队列来控制并发量
- 2.如果当前的异步调用不超过规定的最大值，就从队列中取出执行，
- 3．如果达到最大值，就讲异步调用暂时放在队列中。
- 4.一旦有异步调用结束，就从队列中取一个新的异步调用。



3 | 功能需求

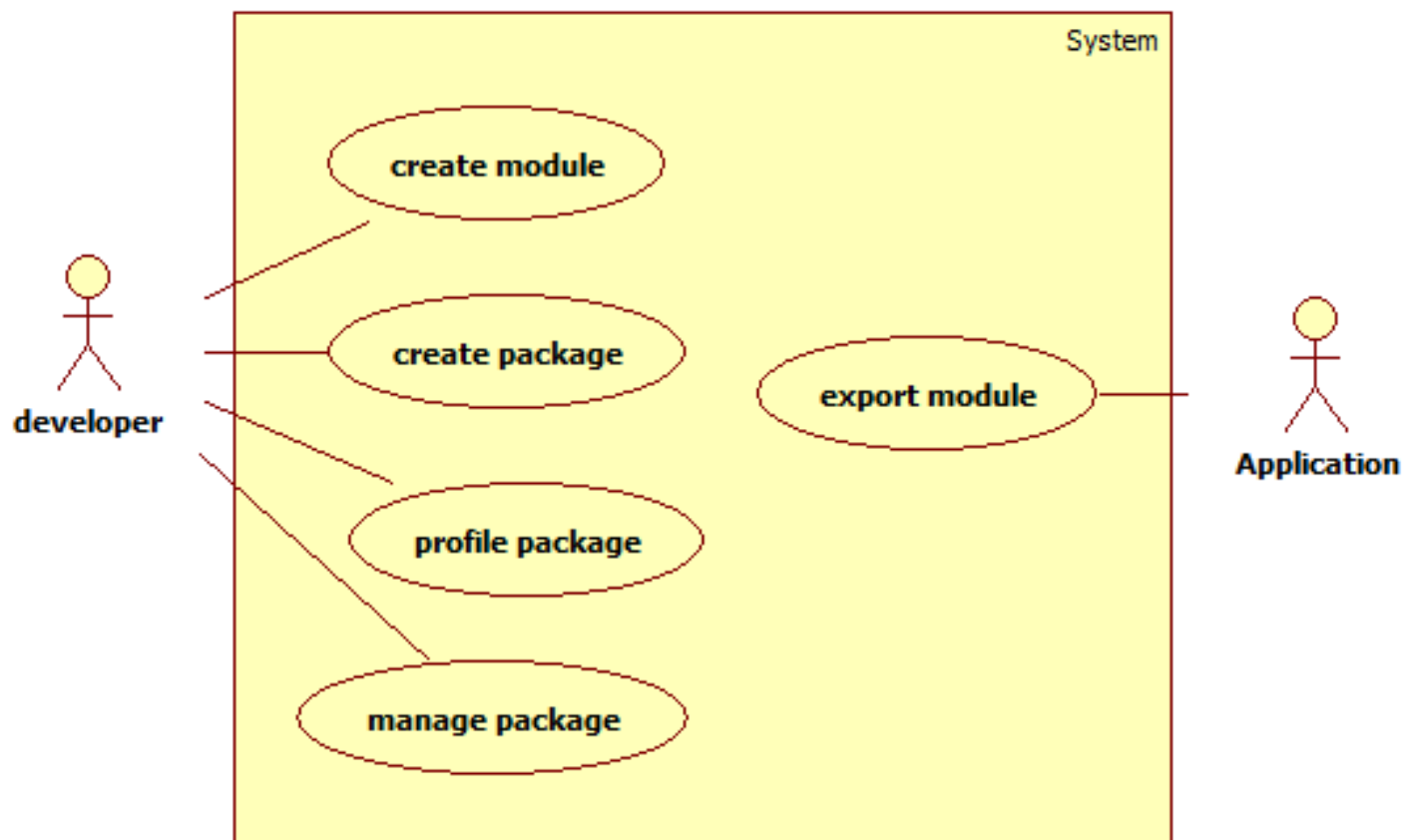
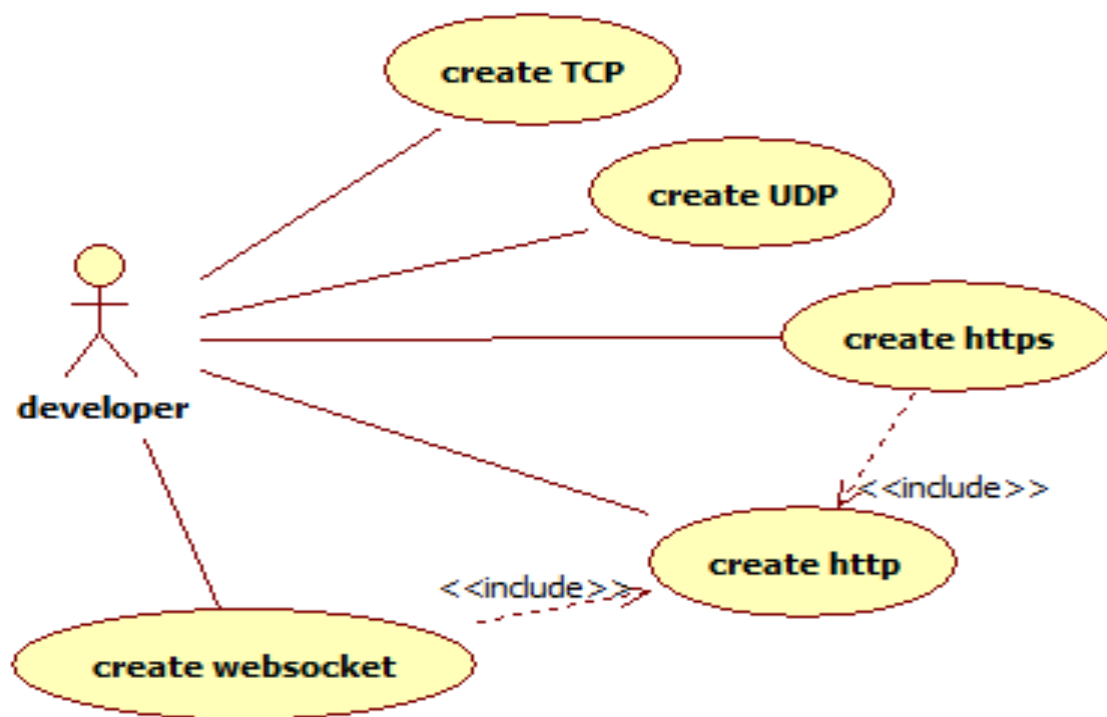
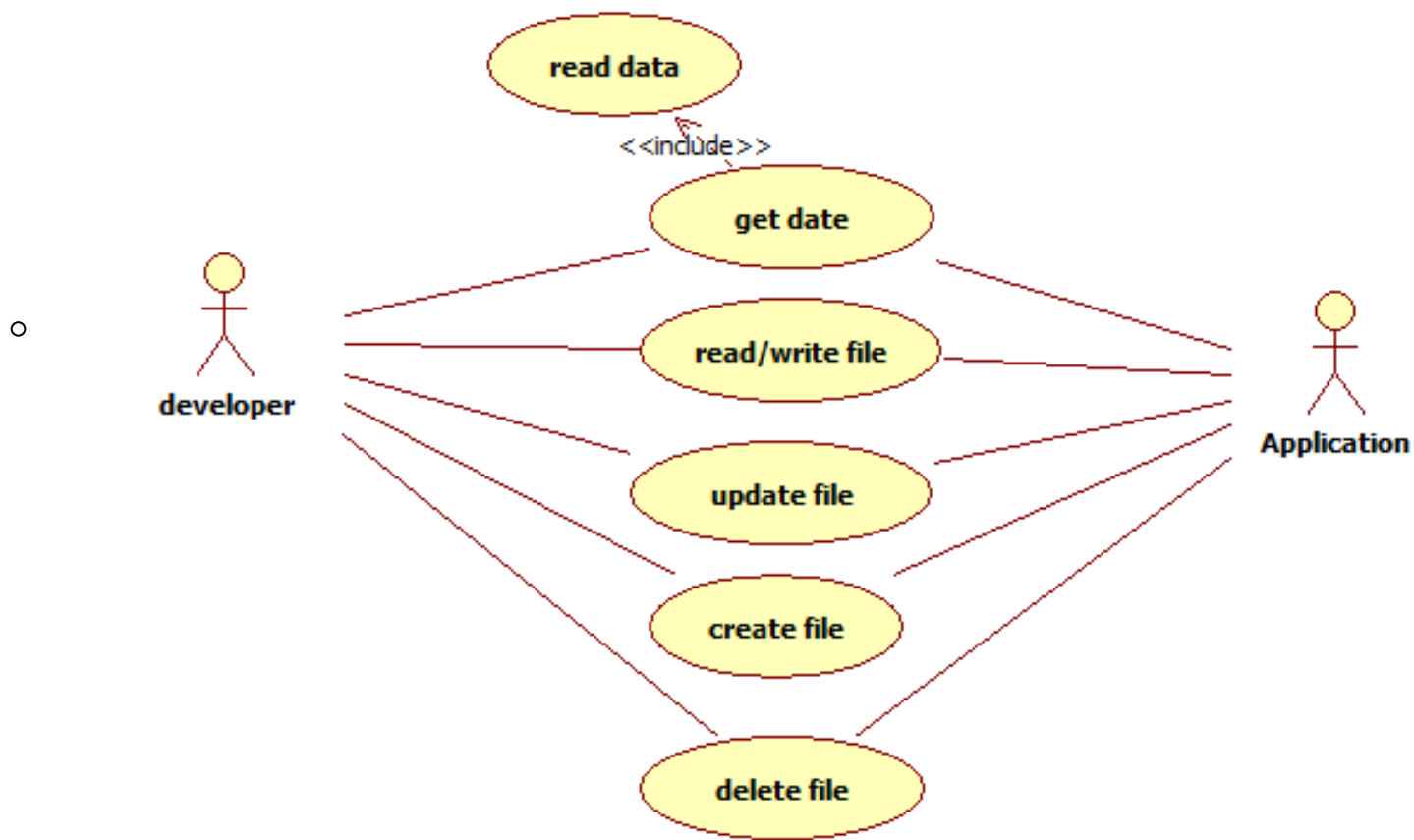


图1 模块和包机制用例图

○



网络通信用例图



文件系统用例图



非功能需求

兼容性

开发者在使用Node时，不需要考虑操作系统的差别，Node在windows,linux上可以实现底层的支持。

高效性

开发者在使用Node时,可以从语言层面很自然的进行并行I/O操作，每个调用无需等待之前的I/O调用结束，从而在编程模型上极大提升效率。

容错性

当程序出现错误时，开发者会被通知到发生了什么错误，而不是直接将这个错误的上下文丢掉或者伴随着错误而退出程序。

可扩展性

开发者可根据自己的需求编写相应的功能模块放入核心模块中。



谢谢

T H A N K S