**BackgroundFetchManager**

**属性**

* backgroundFetch: 返回一个Promise，当此对象完全初始化时解析为true。

**方法**

* get(id): 返回一个Promise，解析为一个BackgroundFetchRegistration对象，该对象对应于具有给定id值的后台下载或上传任务注册。
* getIds(): 返回一个Promise，解析为一个字符串数组，其中包含所有已注册后台任务的id。
* fetch(id, urls[, options]): 启动新的后台下载或上传任务。返回一个 Promise，如果成功则解析为BackgroundFetchRegistration对象。
* delete(id): 从浏览器中删除指定的后台下载或上传任务，并在本地存储中清除所有与此任务关联的信息。返回一个Promise，解析为一个布尔值表示任务是否成功删除。

**BackgroundFetchRegistration**

**属性**

* id: 此任务的唯一标识符。只读。
* downloadTotal: 任务所涉及的所有文件的元数据列表。此元数据包括文件名、大小、MIME类型等信息。只读。
* uploadTotal: 任务涉及的所有上传文件的元数据列表。只读。
* uploaded: 要上传的文件数。只读。
* downloaded: 已下载的文件数。只读。
* totalDownloadSize: 所有要下载的文件的总大小（以字节为单位）。
* totalUploadedSize: 所有要上传的文件的总大小（以字节为单位）。
* paused: 表示当前任务是否已被暂停的布尔值。只读。
* completed: 表示当前任务是否已完成的布尔值。只读。
* failureReason: 枚举型值，表示后台任务失败原因。只读。

**方法**

* getResponse(index): 返回一个Promise, 解析为Blob对象，该对象包含与指定文件对应的数据。
* pause(): 将当前任务暂停，并返回一个Promise，解析为新的BackgroundFetchRecord对象数组。
* resume(): 恢复当前任务，并返回一个Promise，解析为新的BackgroundFetchRecord对象数组。
* abort(): 取消当前任务，并返回一个Promise，解析为新的BackgroundFetchRecord对象数组。
* complete(): 完成当前后台下载或上传任务。 在这一步，您可以清理必要的文件，删除不再需要的Registration对象等。

**BackgroundFetchRecord**

**属性**

* type: 文件的MIME类型。
* url: 文件的URL地址。
* status: HTTP请求状态码，200为成功。
* result: 要写入磁盘的文件数据的Promise。
* lastModified: 文件最近修改时间戳。

**方法**

* waitUntil(promise): 使SW在promise结束后才能关闭。