**[HTML5 indexedDB](http://www.cnblogs.com/Johnny_Z/archive/2012/11/04/2753331.html)**

**[简 介](http://www.cnblogs.com/Johnny_Z/archive/2012/11/04/2753331.html)**

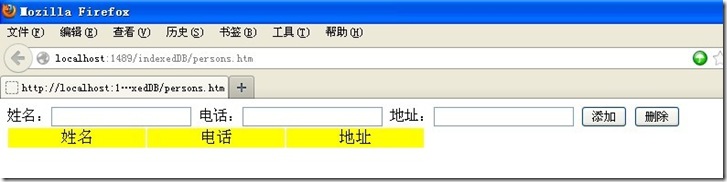
本篇文章主要讲解HTML5的本地存储，使用到的技术是indexedDB。

对于本地存储这个概念想必大家都已经有所了解了，就是把一些数据存储到浏览器中，当与网络断开时，可以从浏览器中读取数据，用来做一些离线应用。

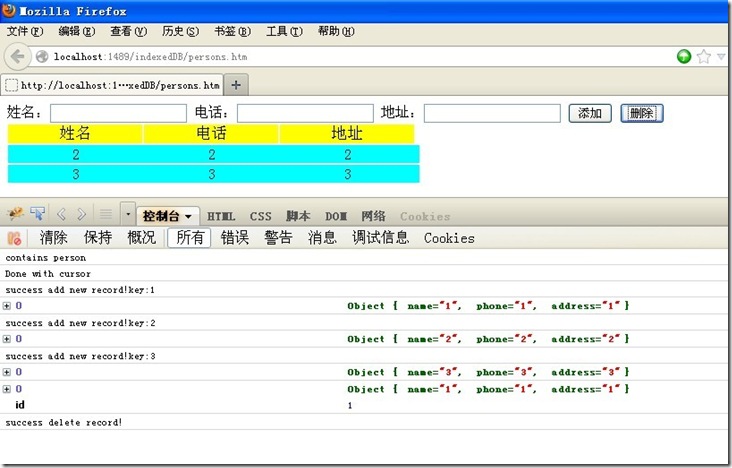
IndexedDB的技术特点是，不需要你去写特定的sql语句来对数据进行操作，它是nosql的，数据形式使用的是json。

下面讲解一个例子，以这个例子来说明一些关键的代码。

这个简单的例子是能增加一个人的信息，并且能删除之。先附上一个效果图：

[](http://images.cnblogs.com/cnblogs_com/Johnny_Z/201211/201211041024358943.jpg)

添加删除信息时配合firebug信息的效果：

[](http://images.cnblogs.com/cnblogs_com/Johnny_Z/201211/201211041024372125.jpg)

示例还是蛮简单的，下面我们来分析一下使用indexedDB的一般步骤，这里只着重讲解一些回调函数和对象，剩下的工作还需要读者自行分析，最后会给出完整的源代码示例。

我们首先需要根据不同浏览器的内核，创建indexedDB对象，如下：

[?](http://www.cnblogs.com/Johnny_Z/archive/2012/11/04/2753331.html)

|  |  |
| --- | --- |
| 1 | var indexedDB = window.indexedDB || window.webkitIndexedDB || window.mozIndexedDB || window.msIndexedDB; |

之后，我们需要打开一个数据库，我这里起名叫做persons：

[?](http://www.cnblogs.com/Johnny_Z/archive/2012/11/04/2753331.html)

|  |  |
| --- | --- |
| 1 | openRequest = indexedDB.open("persons"); |

这里还需要说明一下，open方法是异步方法，我们在这里需要处理它的回调函数即可。例如：

[?](http://www.cnblogs.com/Johnny_Z/archive/2012/11/04/2753331.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | openRequest.onsuccess = function(e)  {  ……  }  openRequest.onerror = function (event)  {  ……  }  openRequest.onupgradeneeded = function(e)  {  ……  } |

我们可以看到，我们需要对成功和错误的回调函数进行处理。onupgradeneeded这个回调函数又是做什么的呢？当数据库发生改变时，我们需要处理这个回调函数。比如说，当某个数据库不存在需要创建，或者数据库更改时。我们一般需要在该回调函数内创建objectStore，它就相当于数据库中的某个表一样。

当我们需要对表中的数据进行遍历时，我们可能需要用到下述代码：

[?](http://www.cnblogs.com/Johnny_Z/archive/2012/11/04/2753331.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | var transaction = db.transaction(["person"],"readwrite");  var objectStore = transaction.objectStore("person");  objectStore.openCursor().onsuccess = function(event)  {  var cursor = event.target.result;  if (cursor)  {  ……  }  }  objectStore.openCursor().onerror = function(event)  {  ……  } |

凡是对数据的读写操作，一般都会用到transacation，它也是异步的。例如，当我们需要读取数据时，有个游标对象，我们操作它就可以很方便的读取数据了。

其他的一些接口就不在这里一一介绍了，感兴趣的朋友可以查看以下网址：

<https://developer.mozilla.org/en/IndexedDB/Using_IndexedDB>

这个是mozilla官方提供的API。

下面还要说重要的一点，就是**运行环境。**

1.页面必须依托于服务器才能正常显示，单独打开页面是无法正常显示的，请把页面发布到服务器中，如IIS、Tomcat、或者调试环境自带的轻量级服务器等。

2.笔者在firefox**14**及以上版本运行代码无误；IE9及更早版本并不支持，据说IE10正式版支持，但是这点笔者并没有实际测试过；opera(当前版本12)目前尚不支持；chrome(当前版本24)支持indexedDB部分功能，但是本例中代码并不能正常运行，据说因为不支持onupgradeneeded这个回调函数，也许chrome有自己的实现策略，这就需要读者自己查找资料了。