HTML5 - INDEXEDDB

The indexeddb is a new HTML5 concept to store the data inside user's browser. indexeddb is more power than local storage and useful for applications that requires to store large amount of the data. These applications can run more efficiency and load faster.

Why to use indexeddb?

The W3C has announced that the Web SQL database is a deprecated local storage specification so web developer should not use this technology any more. indexeddb is an alternative for web SQL data base and more effective than older technologies.

Features

- it stores key-pair values
- it is not a relational database
- IndexedDB API is mostly asynchronous
- it is not a structured query language
- it has supported to access the data from same domain

IndexedDB

Before enter into an indexeddb, we need to add some prefixes of implementation as shown below

```
window.indexedDB = window.indexedDB || window.mozIndexedDB || window.webkitIndexedDB ||
window.msIndexedDB;

window.IDBTransaction = window.IDBTransaction || window.webkitIDBTransaction ||
window.msIDBTransaction;
window.IDBKeyRange = window.IDBKeyRange || window.webkitIDBKeyRange ||
window.msIDBKeyRange

if (!window.indexedDB) {
   window.alert("Your browser doesn't support a stable version of IndexedDB.")
}
```

Open an IndexedDB database

Before creating a database, we have to prepare some data for the data base.let's start with company employee details.

```
const employeeData = [
   { id: "01", name: "Gopal K Varma", age: 35, email: "contact@tutorialspoint.com" },
   { id: "02", name: "Prasad", age: 24, email: "prasad@tutorialspoint.com" }
];
```

Adding the data

Here adding some data manually into the data as shown below −

```
function add() {
   var request = db.transaction(["employee"], "readwrite")
   .objectStore("employee")
   .add({ id: "01", name: "prasad", age: 24, email: "prasad@tutorialspoint.com" });

   request.onsuccess = function(event) {
      alert("Prasad has been added to your database.");
   };
```

```
   request.onerror = function(event) {
      alert("Unable to add data\r\nPrasad is already exist in your database! ");
   }
}
```

Retrieving Data

We can retrieve the data from the data base using with get

```
function read() {
   var transaction = db.transaction(["employee"]);
   var objectStore = transaction.objectStore("employee");
   var request = objectStore.get("00-03");

   request.onerror = function(event) {
      alert("Unable to retrieve daa from database!");
   };

   request.onsuccess = function(event) {
      if(request.result) {
         alert("Name: " + request.result.name + ", Age: " + request.result.age + ",
Email: " + request.result.email);
      }

      else {
         alert("Kenny couldn't be found in your database!");
      }
   };
}
```

Using with get, we can store the data in object instead of that we can store the data in cursor and we can retrieve the data from cursor

```
function readAll() {
   var objectStore = db.transaction("employee").objectStore("employee");

   objectStore.openCursor().onsuccess = function(event) {
      var cursor = event.target.result;

      if (cursor) {
         alert("Name for id " + cursor.key + " is " + cursor.value.name + ", Age: " +
cursor.value.age + ", Email: " + cursor.value.email);
         cursor.continue();
      }

      else {
         alert("No more entries!");
      }
   };
}
```

Removing the data

We can remove the data from IndexedDB with remove.Here is how the code looks like

```
function remove() {
   var request = db.transaction(["employee"], "readwrite")
   .objectStore("employee")
   .delete("02");

   request.onsuccess = function(event) {
      alert("prasad entry has been removed from your database.");
   };
}
```

HTML Code

To show all the data we need to use onClick event as shown below code −

```
<!DOCTYPE html>
<html>
   <head>

      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
      <title>IndexedDb Demo | onlyWebPro.com</title>

   </head>
   <body>

      <button onclick="read()">Read </button>
      <button onclick="readAll()"></button>
      <button onclick="add()"></button>
      <button onclick="remove()">Delete </button>

   </body>
</html>
```

Final code should be as

```
<!DOCTYPE html>
<html>
   <head>

      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
      <script type="text/javascript">
         //prefixes of implementation that we want to test
         window.indexedDB = window.indexedDB || window.mozIndexedDB ||
window.webkitIndexedDB || window.msIndexedDB;

         //prefixes of window.IDB objects
         window.IDBTransaction = window.IDBTransaction || window.webkitIDBTransaction ||
window.msIDBTransaction;
         window.IDBKeyRange = window.IDBKeyRange || window.webkitIDBKeyRange ||
window.msIDBKeyRange

         if (!window.indexedDB) {
            window.alert("Your browser doesn't support a stable version of IndexedDB.")
         }

         const employeeData = [
            { id: "00-01", name: "gopal", age: 35, email: "gopal@tutorialspoint.com" },
            { id: "00-02", name: "prasad", age: 32, email: "prasad@tutorialspoint.com" }
         ];
         var db;
         var request = window.indexedDB.open("newDatabase", 1);

         request.onerror = function(event) {
            console.log("error: ");
         };

         request.onsuccess = function(event) {
            db = request.result;
            console.log("success: "+ db);
         };

         request.onupgradeneeded = function(event) {
            var db = event.target.result;
            var objectStore = db.createObjectStore("employee", {keyPath: "id"});

            for (var i in employeeData) {
               objectStore.add(employeeData[i]);
            }
         }

         function read() {
```

```
            var transaction = db.transaction(["employee"]);
            var objectStore = transaction.objectStore("employee");
            var request = objectStore.get("00-03");

            request.onerror = function(event) {
               alert("Unable to retrieve daa from database!");
            };

            request.onsuccess = function(event) {
               // Do something with the request.result!
               if(request.result) {
                  alert("Name: " + request.result.name + ", Age: " + request.result.age +
", Email: " + request.result.email);
               }

               else {
                  alert("Kenny couldn't be found in your database!");
               }
            };
         }

         function readAll() {
            var objectStore = db.transaction("employee").objectStore("employee");

            objectStore.openCursor().onsuccess = function(event) {
               var cursor = event.target.result;

               if (cursor) {
                  alert("Name for id " + cursor.key + " is " + cursor.value.name + ",
Age: " + cursor.value.age + ", Email: " + cursor.value.email);
                  cursor.continue();
               }

               else {
                  alert("No more entries!");
               }
            };
         }

         function add() {
            var request = db.transaction(["employee"], "readwrite")
            .objectStore("employee")
            .add({ id: "00-03", name: "Kenny", age: 19, email: "kenny@planet.org" });

            request.onsuccess = function(event) {
               alert("Kenny has been added to your database.");
            };

            request.onerror = function(event) {
               alert("Unable to add data\r\nKenny is aready exist in your database! ");
            }
         }

         function remove() {
            var request = db.transaction(["employee"], "readwrite")
            .objectStore("employee")
            .delete("00-03");

            request.onsuccess = function(event) {
               alert("Kenny's entry has been removed from your database.");
            };
         }
      </script>

   </head>
   <body>

      <button onclick="read()">Read </button>
      <button onclick="readAll()">Read all </button>
```

```
      <button onclick="add()">Add data </button>
      <button onclick="remove()">Delete data </button>

   </body>
</html>
```