IndexedDb

Construction

```
var openRequest = window.indexedDB.open('InitDB', 1);
openRequest.onupgradeneeded = function (e) {
    $$result.log('Upgrade Needed');
    var newVersion  = e.target.result;
    if (!newVersion.objectStoreNames.contains('courses')) {
        newVersion.createObjectStore('courses',
                {
                    autoIncrement: true
                });
    }
};
openRequest.onerror = openRequest.onblocked = $$result.log;
openRequest.onsuccess = function (e) {
    $$result.log('Database open');
    // set aside instance of open database
    // declared in earlier scope
    initDB = e.target.result;
};
```

Destruction

```
if (typeof initDB !== 'undefined') {
    $$result.log('Closing the database...');
    initDB.close();
    $$result.log('Attempting to delete the database...');
    var deleteRequest = indexedDB.deleteDatabase('InitDB');
    deleteRequest.onerror = deleteRequest.onblocked = $$result.log;
    deleteRequest.onsuccess = function () {
        $$result.log('Database deleted');
    };
} else {
    $$result.log('You must first create a database before attemping a delete.');
}
```

CRUD Operations

```
var db = {
    name: 'CrudDB',
    version: 1,
    instance: {},
    storeNames: {
        courses: 'courses'
    },
    defaultErrorHandler: function (e) {
        $$result.log(e);
    },
    setDefaultErrorHandler: function (request) {
        if ('onerror' in request) {
            request.onerror = db.defaultErrorHandler;
        }
        if ('onblocked' in request) {
            request.onblocked = db.defaultErrorHandler;
        }
    }
};
```

CRUD Operations - Create

```
var dt = new Date();
var course = {
    title: 'HTML5',
    author: {
        first: 'Abhishek',
        last: 'Prajapati'
    },
    courseID: 'html5-radix',
    insertDate: dt,
    modifiedDate: dt
};
var transaction = db.instance.transaction([db.storeNames.courses], 'readwrite');
var
    store = transaction.objectStore(db.storeNames.courses),
    addRequest = store.add(course);
db.setDefaultErrorHandler(addRequest);
addRequest.onsuccess = function (e) {
    $$result.log('Course added');
    $$result.log('key: ' + e.target.result);
    $('#id-box').val(e.target.result);
};
```

CRUD Operations - Read

```
var transaction = db.instance.transaction([db.storeNames.courses], 'readonly');
var
    store = transaction.objectStore(db.storeNames.courses),
    key = Number($('#id-box').val()),
    getRequest = store.get(key);
db.setDefaultErrorHandler(getRequest);
getRequest.onsuccess = function (e) {
    var course = e.target.result;
    if (course !== undefined) {
        $$result.log(course);
    } else {
        $$result.log('A course with the key of ' + key + ' does not exist');
    }
};
```

CRUD Operations - Update

```
var transaction = db.instance.transaction([db.storeNames.courses], 'readwrite');
var
    store = transaction.objectStore(db.storeNames.courses),
    key = Number($('#id-box').val()),
    getRequest = store.get(key);
db.setDefaultErrorHandler(getRequest);
getRequest.onsuccess = function (e) {
    var course = e.target.result;
    if (course !== undefined) {
        course.modifiedDate = new Date();
        var putRequest = store.put(course, key);
        db.setDefaultErrorHandler(putRequest);
        putRequest.onsuccess = function (e) {
            $$result.log('Course Updated');
        };
    } else {
        $$result.log('A course with the key of ' + key + ' does not exist');
    }
};
```

CRUD Operations - Delete

```
var transaction = db.instance.transaction([db.storeNames.courses], 'readwrite');
var
    store = transaction.objectStore(db.storeNames.courses),
    key = Number($('#id-box').val()),
    deleteRequest = store.delete(key);
db.setDefaultErrorHandler(deleteRequest);
deleteRequest.onsuccess = function (e) {
    $$result.log('Course deleted');
    $('#id-box').val('');
};
```

Keys - AutoKey

```
newVersion.createObjectStore(
    'AutoKey',
    {
        autoIncrement: true
    });
```

Keys - EmailKey

```
newVersion.createObjectStore(
    'EmailKey',
    {
        keyPath: 'email'
    });
```

Keys - AutoKeyWithPath

```
    'AutoKeyWithPath',
    {
        keyPath: 'id',
        autoIncrement: true
    });
```

Keys - GuidKey

```
newVersion.createObjectStore(
    'GuidKey',
    {
        keyPath: 'clientId'
    });
```