

# Dropbox Datastores

Jari-Pekka Voutilainen

Backend as a Service -seminar

12.11.2013

# Dropbox

- Founded June 2007
  - service launched September 2008
- Originally launched as file hosting service with a desktop client to sync the files.
- First developer apis released during 2010.

# Developer APIs

- More secure v1 of the api was released october 2011
  - the old v0 api was retired 1.12.2012
- Datastore api announced july 2013
  - Officially launched 25.9.2013
- Documentantion available at <https://www.dropbox.com/developers>

# Datastore SDKs

Official SDKs are available for:

- IOS
- Android
- JavaScript
- Python

Third party libraries also available for:

- AngularJS, Backbone.js, Xamarin(C#), Core Data

# Using the SDK

Before using the SDK, app must be created in App Console: <https://www.dropbox.com/developers/apps>

```
var client = new Dropbox.Client({key: APP_KEY});

// Try to finish OAuth authorization.
client.authenticate({interactive: false}, function (error) {
  if (error) {
    alert('Authentication error: ' + error);
  }
});

if (client.isAuthenticated()) {
  // Client is authenticated. Display UI.
}
```

# Datastore API

Each application can have multiple datastores

```
dataStoreManager.openDefaultDatastore(function(err, datastore){});  
dataStoreManager.openDatastore(datastoreId, function(err, datastore){});
```

Each datastore can have multiple tables

```
datastore.getTable(tableId);
```

Each table can have multiple records

```
table.query(queryCriteria);
```

# Limitations

- Maximum record size 100 KB
  - Maximum number of record per datastore 100000
  - Maximum datastore size 10 MB
  - Size limit of single sync call 2 MB
- 
- Single app quota is 5 MB, everything exceeding this is taken from user quota

<http://goo.gl/vSHqTW>

# DataStore Manager

API for handling datastores of the application

<https://www.dropbox.com/developers/datastore/docs/js#Dropbox.Datastore.DatastoreManager>

- Get
- Create
- Delete
- List
- ListChanged event



# Datastore

## API for handling single datastore

<https://www.dropbox.com/developers/datastore/docs/js#Dropbox.Datastore>

- Get table
- List tables
- Sync status

# Table & Record

## API for single table

<https://www.dropbox.com/developers/datastore/docs/js#Dropbox.Datastore.Table>

- Get record by id
- Get records with query
- Insert record

## API for single record

<https://www.dropbox.com/developers/datastore/docs/js#Dropbox.Datastore.Record>

- Get/set field name
- Update / delete record

# Using datastores in server-side with node.js

```
var dropbox = require('./dropbox-datastores-1.0.0');  
var client = new dropbox.Client({  
  key: '<key from console>',  
  secret: '<secret from console>'  
});
```

```
var authDriver = new dropbox.AuthDriver.NodeServer(8192);  
client.authDriver(authDriver);
```

Datastore API functions don't have callbacks, see [demo](#).

Only works on localhost, since official library opens a new page to default browser. True server-side requires implementing OAuth process.

# Pros / cons

## Pros

- Easy to use
- Don't have to care user management
- Well established API which also can be used to handle files
- Good documentation

## Cons

- No sharing data between users
- No callbacks might create synchronicity problems in node.js environments