# LINVODB

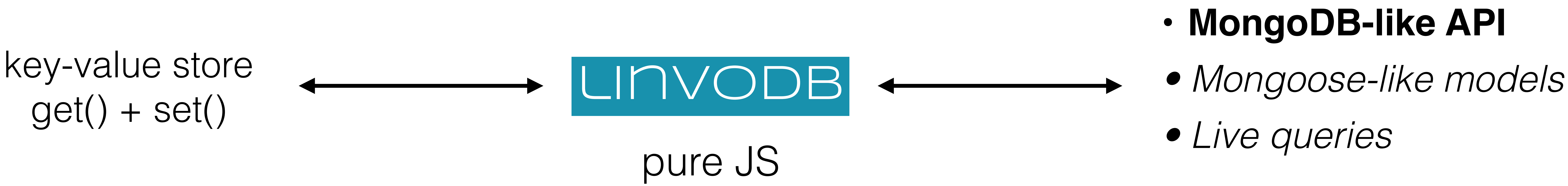
embeddable MongoDB alternative

http://github.com/Stremio/linvodb3

http://github.com/Stremio

# what is linvoDB?

Not really a DBMS:



key-value store
get() + set()

LINVODB

pure JS

- **MongoDB-like API**
- *Mongoose-like models*
- *Live queries*

# Use case

- < 1 million objects
- HTML5 / Electron / NW.js
- AngularJS / React / Meteor / …
- NativeScript, React Native
- Node.js - but not for servers

# you can build…

## Mobile apps

SQLite / LevelDB back-end

## Desktop apps

LevelDB / Medea back-end

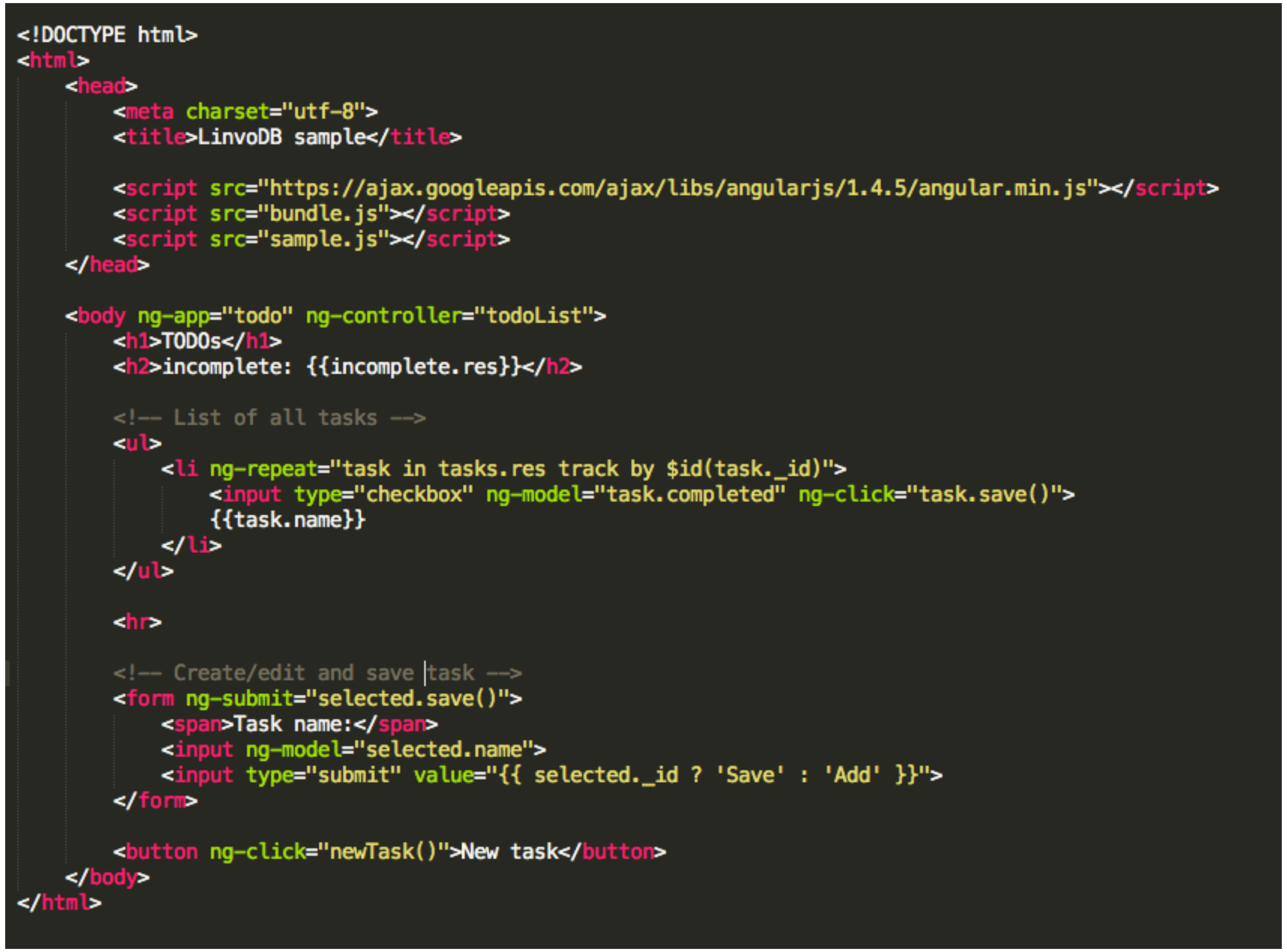
## HTML5 apps

localStorage / IndexedDB back-end

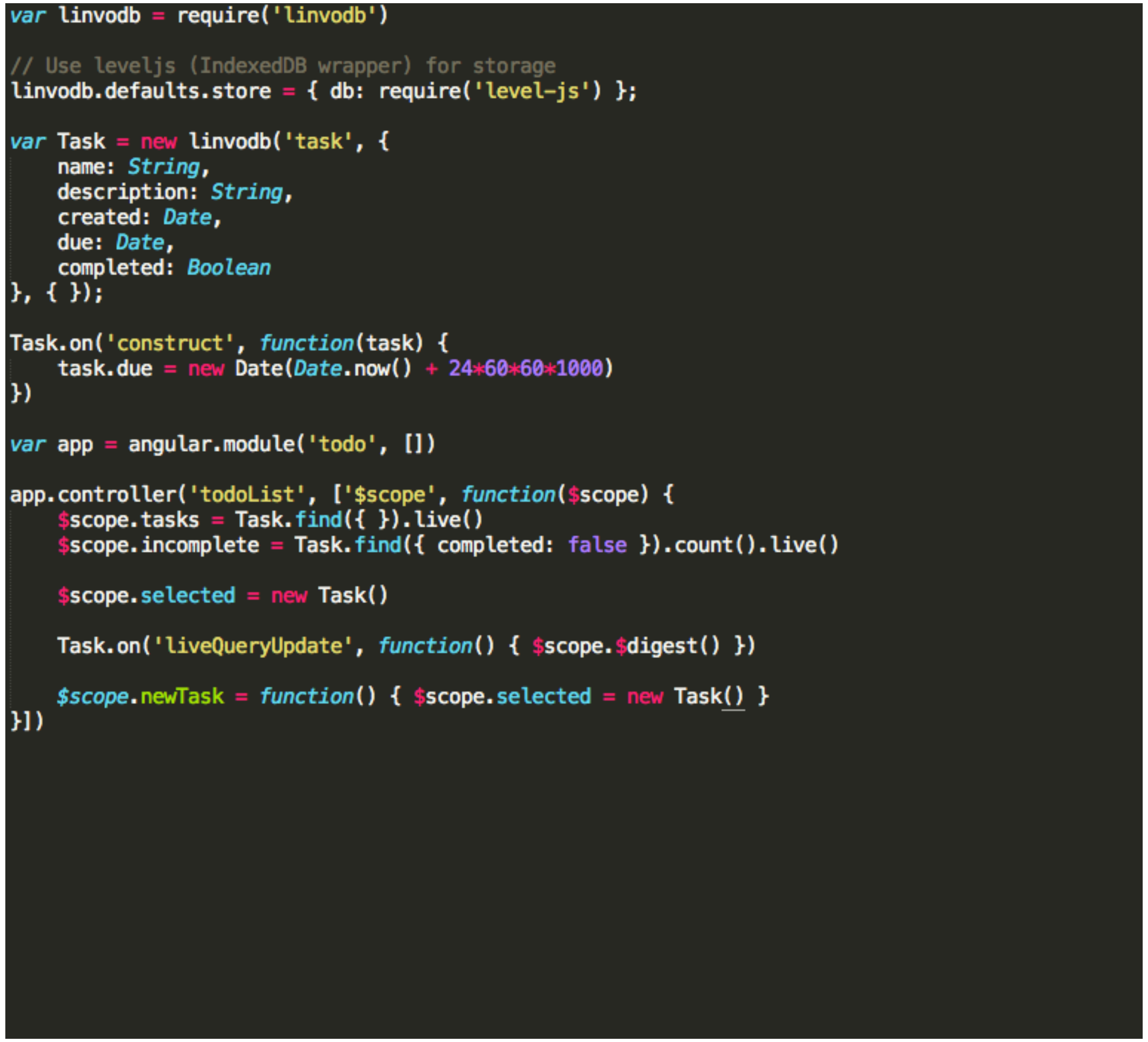
# Features

- Use any back-end - e.g. LevelDB, IndexedDB, SQLite, Medea, LocalStorage, redis…
- NW.js / Electron / NativeScript - friendly
- MongoDB-like query language
- Auto indexes
- Live queries
- Models / schemas
- Map / Reduce / Filter / Aggregate

# Example



```html
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>LinvoDB sample</title>

        <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.5/angular.min.js"></script>
        <script src="bundle.js"></script>
        <script src="sample.js"></script>
    </head>

    <body ng-app="todo" ng-controller="todoList">
        <h1>TODOs</h1>
        <h2>incomplete: {{incomplete.res}}</h2>

        <!-- List of all tasks -->
        <ul>
            <li ng-repeat="task in tasks.res track by $id(task._id)">
                <input type="checkbox" ng-model="task.completed" ng-click="task.save()">
                {{task.name}}
            </li>
        </ul>

        <hr>

        <!-- Create/edit and save task -->
        <form ng-submit="selected.save()">
            <span>Task name:</span>
            <input ng-model="selected.name">
            <input type="submit" value="{{ selected._id ? 'Save' : 'Add' }}">
        </form>

        <button ng-click="newTask()">New task</button>
    </body>
</html>
```



```javascript
var linvodb = require('linvodb')

// Use leveljs (IndexedDB wrapper) for storage
linvodb.defaults.store = { db: require('level-js') };

var Task = new linvodb('task', {
    name: String,
    description: String,
    created: Date,
    due: Date,
    completed: Boolean
}, { });

Task.on('construct', function(task) {
    task.due = new Date(Date.now() + 24*60*60*1000)
})

var app = angular.module('todo', [])

app.controller('todoList', ['$scope', function($scope) {
    $scope.tasks = Task.find({ }).live()
    $scope.incomplete = Task.find({ completed: false }).count().live()

    $scope.selected = new Task()

    Task.on('liveQueryUpdate', function() { $scope.$digest() })

    $scope.newTask = function() { $scope.selected = new Task() }
}])
```

browser demo

# comparison

## NeDB

+ simple storage format

- node.js specific

-- keeps everything in memory

## SQLite

+ widespread

+ performance

- SQL

## linvoDB

+ live queries, map/reduce, schemas

+ / - use it with any back-end

- ???

# Under the hood

**cursor.js** - all of the query cursor/execution logic
**document.js** - document comparison, query language
**indexes.js** - indexes with binary-search-tree
**model.js** - glue, operations like update/remove/bulk save
**schemas.js** - schema enforcing

## deps

**async, bagpipe** - flow control
**binary-search-tree** - fast in-memory indexes
**levelup** - key/value store abstraction
**lodash** - utils

# Going crazy

- full text search in memory - linvodb-fts - trie/metaphone
- p2p replication - mafintosh/multi-master-merge
- persistent indexes - allows for server-side uses
- compound indexes
- ES6 - c58/marsdb

# In the wild…

stremio

http://www.strem.io/

SLIDEWINDER

http://slidewinder.io/

https://www.npmjs.com/package/clipboard-manager

# Thank you!

# Questions?

Ivo Georgiev | ivo@strem.io