Persisting Data: NativeScript Storage

Jogesh K. Muppala

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

NativeScript Data Storage

- Several methods for on-device data storage available
  - Application Settings
  - File System
  - SQLite
  - Couchbase Lite

Application Settings

- Allows you to save and restore any kind of information related to your application
  - Application Settings module
- Key-value based storage:
  - Data Types supported: Boolean, Number, String
- Data access through set*() and get*() methods
  - e.g. setString(“key”, “value”)
  - e.g., getString(“key”, “default value”)
  - has(“key”), remove(“key”), clear()

File System

- High-level abstractions for file system entities such as files, folders, known folders, paths, separators, etc.
  - import File System Module
- Create, read, delete and update operations supported

Databases

- Ideal storage for repeated or structured data
- Allows for adding, reading, updating and deleting data:
  - Typical CRUD operations
    - Create
    - Read
    - Update
    - Delete
- Best to define a Service to manage the database and export a convenient API to use within your components

SQLite Database

- SQLite is a light weight database
  - Atomic
  - Stable
  - Independent
  - Enduring
  - Only several kilobytes
  - Only partly support some SQL commands such as ALTER, TABLE.
- SQLite is included as part of both Android and iOS software stack
- More info about SQLite at http://www.sqlite.org
- NativeScript usage of SQLite through the plugin

  tns plugin add nativescript-sqlite

Couchbase Lite

- NoSQL document database
  - Embedded JSON database that can work standalone, in a P2P network, or as a remote endpoint for Sync Gateway
- NativeScript support for Couchbase Lite

  tns plugin add nativescript-couchbase
- Supported methods: getDocument(), createDocument(), updateDocument(), deleteDocument()
  - Querying with MapReduce View: createView(), executeQuery()