Project Setup

Patrick Schroeder
SOFTWARE ENGINEER

Overview

Sequelize Setup

- Node.js and Sqlite

Models

- Synch and view

Data Types

Primary Keys

SQLite Installation

Database Options

MySQL

SQLite

PostgreSQL

MSSQL

SQLite

Lightweight

Command line access

Pre-Installed

Sequelize maps Javascript objects to SQL tables using Models.

Model

```
User = {
  name: Sequelize.STRING,

  bio: Sequelize.STRING
};
```

- Model named User
- Attributes name and bio

Table

Users	
name	**bio**
matt	matt's bio
john	john's bio

- Table named Users
- Key's match Column names
- Value placed in Table Row

Model

```
const User =
  connection.define('User', {
    name: Sequelize.STRING,
    bio: Sequelize.TEXT
})
```

- Model name of User
- Define accepts 2 arguments

Sync Database

```
connection.define('User', {
    name: Sequelize.STRING,
    bio: Sequelize.TEXT
}).sync();


connection.sync();


connection.sync({
   logging: console.log
})
```

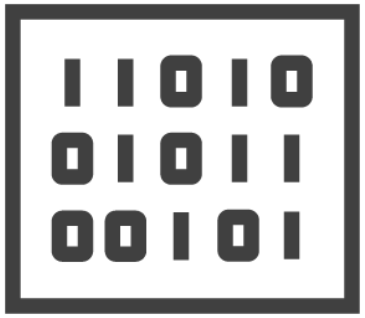
◂ Invoke sync() on model

◂ Invoke sync() on connection

◂ Sync with logging

Data Types

Data Types



- String
- Integer
- Date
- Array
- Boolean

Primary Keys

Primary Keys

- Unique identifiers
- Form associations
- Automatically generated

Summary

- Project setup
- Sqlite connection
- User model
- Created data
- Data types
- Primary keys