LU3IN017 - Technologies du Web 2021

LOI Alessia, PIRONIN Julie - Groupe 2

**Projet Birdy**

Tables

**Tables SQLite**

**const** createUserTable = `CREATE TABLE IF NOT EXISTS "users" (

      "id"  INTEGER,

      "login" VARCHAR(50) NOT NULL,

      "password"  VARCHAR(50) NOT NULL,

      "lastname"  VARCHAR(50) NOT NULL,

      "firstname" VARCHAR(50) NOT NULL,

      PRIMARY KEY("id" AUTOINCREMENT)

    )`;

    db.exec(createUserTable, **function**(err){

      if(err){

        throw err;

      }

      console.log('User table ready');

    })

**const** createAuthorsTable = `CREATE TABLE IF NOT EXISTS "authors" (

      "id"  INTEGER,

      "lastname"  VARCHAR(50),

      "firstname" VARCHAR(50),

      "alias" VARCHAR(50),

      PRIMARY KEY("id" AUTOINCREMENT)

    )`;

    db.exec(createAuthorsTable, **function**(err){

      if(err){

        throw err;

      }

      console.log('Authors table ready');

    })

**const** createBooksTable = `CREATE TABLE IF NOT EXISTS "books" (

      "id"  INTEGER,

      "id\_author" INTEGER NOT NULL,

      "title" VARCHAR(50) NOT NULL,

      PRIMARY KEY("id" AUTOINCREMENT)

    )`;

    db.exec(createBooksTable, **function**(err){

      if(err){

        throw err;

      }

      console.log('Books table ready');

    })

**const** createFriendsTable = `CREATE TABLE IF NOT EXISTS "friends" (

      "id1" INTEGER NOT NULL,

      "id2" INTEGER NOT NULL,

      "demanding" INTEGER NOT NULL,

      "accepting" INTEGER NOT NULL,

      PRIMARY KEY("id1" , "id2")

    )`;

    db.exec(createFriendsTable, **function**(err){

      if(err){

        throw err;

      }

      console.log('Friends table ready');

    })

**const** createFollowersTable = `CREATE TABLE IF NOT EXISTS "followers" (

      "id1" INTEGER NOT NULL,

      "id2" INTEGER NOT NULL,

      "entity" VARCHAR(50) NOT NULL,

      PRIMARY KEY ("id1" , "id2", "entity")

      FOREIGN KEY ("id1") REFERENCES "users" ("id") ON DELETE CASCADE

    )`;

    db.exec(createFollowersTable, **function**(err){

      if(err){

        throw err;

      }

      console.log('Followers table ready');

    })

**Tables MongoDB**

**const** plumesSchema = db.Schema({

*// id genere automatiquement par mongoose*

      userId : {type : String, required : true},

      typeText : {type : String, required : true},

      text : {type : String, required : true},

      image : {type : String},

      date : {type : Date, default: Date.now},

      entity\_id : {type : String, required : true}, *// id livre ou author*

      typeEntity : {type : String, required : true}, *// livre ou author*

      comments : [],

      spoiler : {type : Boolean, default: false}

    });

*// Compiler le schema en modèle*

    this.Plume = db.model('Plume', plumesSchema);

    console.log("Plumes table ready");

**const** rdvLecturesSchema = db.Schema({

*// id genere automatiquement par mongoose*

      speaker : {type : String, required : true}, *//nom, prenom ou alias*

      title : {type : String, required : true},

      text : {type : String, default : "Rendez-vous lecture en programme!"},

      bookId : {type : String, required : true},

      authorId : {type : String, required : true},

      image : {type : String},

      dateStart : {type : Date, required : true}, *// new Date("2016-05-18T16:00:00Z");*

      dateStop : {type : Date, required : true},

      createdBy : {type : String, required : true}, *//userId*

      createdOn : {type : Date, default: Date.now},

      link :  {type : String, required : true}

    });

*// Compilation du schéma en modèle*

    this.rdvLecture = db.model('rdvLecture', rdvLecturesSchema);

    console.log("rdvLecture table ready");