```
import { OfflineLog } from "./log.js";
import { makeHttpRequest } from "./helper.js";

export class LogCache {

    constructor() {
        this.logs = this.getLogs();
    }

    getLogs() {
        let swiftLogsData = localStorage.getItem('swiftLogs');

        if (swiftLogsData) {
            swiftLogsData = JSON.parse(swiftLogsData);
            return swiftLogsData.map(logData => OfflineLog.fromData(logData))
        }

        return [];
    }

    addLog(log) {
        this.logs.push(log);
        this.saveLogs();
    }

    saveLogs() {
        localStorage.setItem('swiftLogs', JSON.stringify(this.logs));
    }

    addToDb() {

        return new Promise((resolve,reject) => {
            // Build an array of the logs in the format the API expects
            let logData = this.logs.map(log => log.generateDBInsertObject())

            // Insert the log data into the DB
            makeHttpRequest('api/logs','POST',JSON.stringify(logData),'application/json')
            .then(res => {
                res = JSON.parse(res);
                if (res.logIds) {
                    this.logs = [];
                    this.saveLogs();
                    resolve();
                }
                else {
                    reject();
                }
            })
            .catch(res => {
                console.log("ERROR");
                reject();
            })

        });
    }

    hasLogs() {
        return this.logs.length > 0;
    }

    clearCache() {
        this.logs = [];
        this.saveLogs();
    }
}
```