

# Digital Solutions Challenge

## Description

### Web Browser Log Catcher (in Javascript)

#### Common Issue :

Frontend applications usually keep errors and exceptions for themselves. (in console) . Console logs are returned to the user only. This is an issue for web applications based on client-server structure, users usually raise some errors but it's not possible or difficult for technical teams to understand what happens on client side.

The challenge is to implement a JS client-side log catcher/appender/sender to send logs (occurring on client side only) to an external server.

Objective is to facilitate Maintenance of applications and anticipate fixing of some errors that some users may face and not visible for maintenance technical team.

Implementations, working on any web Javascript application (Angular, React, Native JS), for different, browsers, (Chrome & Firefox), consist in creation of a library collecting as much user log events as possible in order to facilitate understanding, by maintenance technical teams, of errors happening on client side .

*Any questions?*



**Loic Andraud**  
Digital Solutions

# Digital Solutions Challenge

## Requirements

- Personal Computer
- Javascript IDE
- Web Browser (Chrome, Firefox)
- Web Server (nginx, apache, ...)

## Deliverables

Send the written code (within allowed time) with a brief explanation on how to use it to:

- Loic on Slack
- By email to: [loic.andraud@saint-gobain.com](mailto:loic.andraud@saint-gobain.com)

# Digital Solutions Challenge

## Digital Solutions Challenge Scoring

*The challenge consists, in Javascript :*

*(20% of the score)*

*1/ Generate 3 type of errors by the code the most various as possible*

*it should generate some errors in your browser console*

*(60% of the score)*

*2/ Catch in the code as much logs events as possible (if possible) :*

- logs raised in the browser's console*
- errors for JS Frameworks (React, Angular, Native JS, ....)*
- errors for external requests to APIs*
- CORS Exceptions*

*(20% of the score)*

*3/ Format log events following below Pattern:*

*[FormattedDate] [LogLevel] [FileName] [Class] [Method] [Message]*

*( + 5% <=100%)*

*4/ OPTIONAL (BONUS) : Queue Lines*

*( + 5% <=100%)*

*5/ OPTIONAL (BONUS) : Automatically and regularly sends logs in queue to rest API (ex: Elasticsearch, Kibana, ...), and clear queue*