

FICHE : PoC & CHOIX TECHNOLOGIQUES

=> Après avoir rempli cette fiche : convertir ce document en PDF et le téléverser dans votre dépôt Github dans un répertoire nommé 'doc'.

COMPTE RENDU DE LA VEILLE TECHNOLOGIQUE PoC

QUESTIONS DE RECHERCHE SUR LA PoC

Comment récupérer les données ?

Comment traiter les données pour obtenir un ensemble cohérent ?

Comment renvoyer de manière optimale les données ?

Comment éviter les surcharges liées au trafic ?

Comment faire communiquer 2 processus NodeJs ?

POC = PREUVE DE CONCEPT

Réaliser un prototype récupérant les données et les traitant avant de les renvoyer.

Tenter d'implémenter un système de "rate limit"

Échanger un message entre 2 processus locaux

LES MARQUE-PAGES IDENTIFIÉS LORS DE VOS RECHERCHES

Lien vers une page publique contenant vos marque-pages collaboratifs ou lister les marque-pages directement ici. Pour chaque lien : URL, nom de la page et description sommaire.

[Module express-rate-limit](#) : lib NPM permettant de créer des limites pour chaque point d'accès de l'API dat

[Ensemble de données ouvertes des CROUS](#) : liste de 5 fichiers contenant des données liées aux CROUS

LES PREUVES DE CONCEPT

Pour chaque preuve de concept réalisée : identifier le but de la preuve de concept (ce qu'elle vérifie), le lien vers le sous-répertoire de votre dépôt GitHub qui contient le code de la preuve de concept ainsi que les résultats de votre expérimentation, puis, finalement, vos conclusions.

Au moins une preuve de concept doit être documentée et réalisée.

PREMIÈRE POC RÉALISÉE

Preuve : Communication entre 2 processus NodeJS locaux

URL Github : [POC/sockets/](https://github.com/your-repo/POC/sockets/)

EXPLIQUEZ VOTRE POC

Décrivez la Poc en détails.

La Poc comprend 2 processus NodeJS dont un écoutant sur le port 3015.

Le processus sans port enverra un événement "PING" et l'autre répondra avec un événement "PONG"

Que PROUVE la Poc ?

la Poc prouve qu'il est possible à 2 processus de communiquer

Que reste-t-il à prouver ?

Il reste à prouver qu'il est possible d'envoyer une charge utile au format JSON ainsi que la possibilité de diffuser à une grande échelle

Quels sont vos résultats de la Poc ?

Résultats Valide : les 2 processus communiquent via le socket et échangent les informations souhaités via les événements définis dans le code.

PREMIÈRE TECHNOLOGIE SÉLECTIONNÉE (LA NOUVELLE)

Technologie : Socket.io

URL : <https://www.npmjs.com/package/socket.io>

JUSTIFIER VOTRE CHOIX TECHNOLOGIQUE POUR CETTE TECHNOLOGIE

Socket.IO permet une communication basée sur la programmation événementielle en temps réel et bidirectionnelle. Cette communication consiste en un serveur Node.js et un client Javascript/Node.js

DEUXIÈME TECHNOLOGIE SÉLECTIONNÉE (LA CONNUE)

Technologie : Requête XMLHttpRequest

URL : <https://developer.mozilla.org/fr/docs/Web/API/XMLHttpRequest>

JUSTIFIER VOTRE CHOIX TECHNOLOGIQUE POUR CETTE TECHNOLOGIE

Cette technologie permet de réaliser des requêtes web sur un serveur et de demander une réponse

DEUXIÈME POC RÉALISÉE

Preuve : Récupération de données liées aux restaurants universitaires

URL Github : [POC/crousResto/](#)

EXPLIQUEZ VOTRE POC

Décrivez la Poc en détails.

La Poc comprend 1 processus NodeJS réalisant une requête afin de récupérer les données liées aux restaurants universitaires français et de les stocker de manière à ce que les données soient accessible facilement

Que PROUVE la Poc ?

la Poc prouve qu'il est possible de récupérer les informations souhaitées sur les CROUS

Que reste-t-il à prouver ?

Il reste à prouver qu'il est possible de traiter les données afin d'établir des liens solides entre 2 renseignements liés au même restaurant/objet mais de 2 sources d'informations distinctes

Quels sont vos résultats de la Poc ?

à définir

PREMIÈRE TECHNOLOGIE SÉLECTIONNÉE (LA NOUVELLE)

Technologie : Axios

URL : <https://www.npmjs.com/package/axios>

JUSTIFIER VOTRE CHOIX TECHNOLOGIQUE POUR CETTE TECHNOLOGIE

Réalisation de requêtes HTTP depuis Node.JS

Supporte les Promesses

Intercepte les requêtes et les réponses pour réaliser des transformations si souhaitées

Possibilité d'annuler une requête en cours

Transformation automatique des données en JSON

Support client pour la protection contre les attaques XSRF

DEUXIÈME TECHNOLOGIE SÉLECTIONNÉE (LA CONNUE)

Technologie : node-fetch

URL : <https://www.npmjs.com/package/node-fetch>

JUSTIFIER VOTRE CHOIX TECHNOLOGIQUE POUR CETTE TECHNOLOGIE

Au lieu d'implémenter XMLHttpRequest pour lancer une requête spécifique aux navigateurs node-fetch utilise directement l'API "http to fetch" native.