Développement par l'API

MIX-IT 2013

Éric Daspet - David Larlet

Qui sommes-nous ?

Éric Daspet

- TEA, livre num.
- PHP 5 avancé
- Performances web
- Paris-Web

eric.daspet.name

David Larlet

- scopyleft
- artisan
- geek
- citoyen

larlet.fr/david

no-conférence

nous n'enseignons pas,
nous échangeons nos expériences

Qu'abordons-nous

Par quoi commencer ?

Comment structurer ?

Quels formats ?

Comment ajouter une nouvelle version ?

Que faire des erreurs ?

Comment gérer les spams et les abus ?

Comment paginer ?

Hypermédia et découverte, késaco ?

Quelles autres bonnes pratiques ?

Retrié par popularité (Doodle)

Comment structurer ?

Appel, URL, ressources, hiérarchies

Structure

REST, fuyez SOAP, évitez le "truc perso"
Ne vous limitez pas aux verbes, utilisez HTTP

Une ressource = un nom, en minuscules

hiérarchies à plat, /ressource/ et /ressource/id
profondeur limitée à /ressource/id/ressource

vs. approche liée (hypermedia)

Comment faire une nouvelle version ?

v1.2.4-beta

Versionnement

Restez compatible : même URI, ajout de paramètres optionnels, valeurs additionnelles

Si besoin d'une nouvelle version :

- uniquement des versions majeures (v1, v2)
- /v2/* en racine du service
- rarement dans le chemin de la ressource

vs. approche liée (hypermedia)

Comment sécuriser ?

Sécurité

Utilisez ce que vous maîtrisez

Toujours du SSL, toujours valider les certificats

Auth HTTP basic fonctionne très bien
Auth déléguée : OAuth 1, attention à OAuth 2
Évitez les cookies de session (CRSF)

Demandez une clef d'api par applicatif

Que faire des erreurs ?

Erreurs

Utilisez les erreurs HTTP
au minimum 2xx, 3xx, 4xx, 5xx

Détail dans le corps du message respectant le format attendu

Code + lien vers l'aide en ligne correspondante
message : pour dev ou pour utilisateur final ?
différenciez bien les deux

Hypermédia et découvertes, késaco ?

Hypermédia et découverte

Envisager votre API comme votre site Web : navigable, standard, indexable, évolutif.

Ne plus fournir des ids mais des liens

Utiliser un format qui permette d'utiliser l'hypertext (pas JSON par exemple)

Un point d'entrée unique devrait suffire

Par quoi commencer ?

Mobile first ?

API First

D'abord mener la réflexion sur les données
et sur comment elles seront utilisées

Penser l'API en prenant la place de l'utilisateur

Et si vous deveniez votre propre utilisateur ?

Éventuellement en commençant par le mobile
ou y brancher des applicatifs indépendants

Comment gérer les spams et abus ?

SPAM et abus

Throttling/métriques

- par IP
- par application (clef d'API)
- par utilisateur

Coûteux en ressources, mais indispensable

Quid d'imposer de toujours être authentifié ?

Limiter la taille des résultats

Timeout contre les requêtes longues

Comment paginer ?

Pagination

Faire simple ? pas ici, pièges en vue

Cas simple : sort + order + limit + offset
Hypermedia : liens rel=next, rel=prev

Quid des nouveaux items entre 2 requêtes ?
exemple avec twitter :
paramètre since, upto avec une date ou un id

Quel(s) format(s)

JSON, ATOM, XML, (x)HTML, RDF, www-form-urlencoded, ...

Format

Requêtes : pensez au form-encoded

Utilisez des formats standard !
Utilisez un format extensible pour plus tard

Utilisez une extension de fichier dans l'URL
Déclarez le format dans les entêtes
Imposez le format, et le codage (UTF8 svp)

Quelles bonnes pratiques ?

Bonnes pratiques

Soyez en relation avec vos utilisateurs

Dates au format ISO, *avec* fuseaux horaires

Utilisez des codes, pas des messages texte

Hypermedia : documentez les types de liens

Améliorer les performances a posteriori (préchargement des sous-ressources)

Pas de questions

Normalement c'est déjà fait :-)

Quelques retours sur le format utilisé ?