

Présentation du format JSON appliqué au Sandre

Thème :

FORMAT D'ECHANGE

Version :

1.0



Version 1	
	07/10/2024 : Création du document

Les conditions d'utilisation de ce document Sandre sont décrites selon la licence *creative commons* ci-dessous. Elles indiquent clairement que vous êtes libre de :

- partager, reproduire, distribuer et communiquer cette œuvre,
- d'utiliser cette œuvre à des fins commerciales.



Chaque document Sandre est décrit par un ensemble de métadonnées issues du Dublin Core (<http://purl.org/dc>).

Titre	Spécifications du format JSON appliqué au Sandre
Créateur	Système d'Information sur l'Eau / SANDRE
Sujet	JSON ; format d'échange
Description	Décrit l'utilisation du format JSON pour les échanges de données selon les prescriptions Sandre.
Editeur	Office Français de la Biodiversité (OFB)
Contributeur	Administrateurs de données SANDRE
Date / Création	07/10/2024
Date / Modification	22/06/2025
Date / Validation	19/06/2025
Type	Text
Format	Adobe Acrobat
Identifiant	http://id.eaufrance.fr/docpres/formatjson/1
Langue	Fr
Couverture	France
Droits	© Sandre
Version	1.0

I. AVANT PROPOS

I.A. Le Système d'Information sur l'Eau et le Sandre

Le domaine de l'eau est caractérisé par le grand nombre d'acteurs qui sont impliqués dans la réglementation, la gestion et l'utilisation des eaux : ministères avec leurs services déconcentrés, établissements publics comme les agences de l'eau, collectivités locales, entreprises publiques et privées, associations... Tous ces acteurs produisent des données pour leurs propres besoins. La mise en commun de ces gisements d'information est une nécessité forte.

Le Système d'Information sur l'Eau (SIE ou SI Eau) est un dispositif créé par l'État pour le partage et la mise à disposition des données sur l'eau, les milieux aquatiques et les services publics d'eau et d'assainissement du secteur public. Il est visible sur le web à travers le portail eaufrance.fr. Eaufrance est un service public qui a pour but de faciliter l'accès à l'information publique dans le domaine de l'eau en France. Il propose un accès aux données du SIE grâce à une quarantaine de sites thématiques qui diffusent des données brutes ou élaborées, des valorisations, des actualités thématiques, etc. Il propose également des informations générales sur l'eau et les milieux aquatiques à destination de tout public : des articles et des synthèses, des publications, des datavisualisations, des cartes et des chiffres-clés... Pour partager les données collectées, le SIE s'est doté d'un référentiel composé de protocoles et de méthodes et d'un référentiel technique Sandre composé d'un langage commun et de "standards" d'échange des données qui, de la production à la diffusion, encadrent les différentes étapes. Ce dernier est élaboré par le Service d'administration nationale des données et référentiels sur l'eau (SANDRE).

Le Schéma National des Données sur l'Eau (SNDE) définit le système des données publiques de l'eau et fonde sur celui-ci le SIE, son service d'information Eaufrance, en organise la gouvernance, décrit son référentiel technique et les modalités de son approbation. Il s'inscrit dans la démarche de modernisation de l'action publique engagée par l'Etat. La traduction réglementaire du SNDE se compose de l'arrêté du 27 mai 2021 modifiant l'arrêté du 19 octobre 2018 approuvant le schéma national des données sur l'eau, les milieux aquatiques et les services publics d'eau et d'assainissement et de l'article R.131-34 du Code de l'environnement. Complété par des documents techniques dont ceux du SANDRE, le SNDE doit être respecté par tous les acteurs du SIE conformément au Décret n° 2019-1580 du 31 décembre 2019.

I.B. Le SANDRE

I.B.1. Les missions du Sandre

Le SANDRE a pour mission :

d'établir le référentiel technique Sandre pour le SIE et contribuer aux autres systèmes qui lui sont liés tels que le Système d'Information sur le Milieu Marin (SIMM), le Système d'Information sur la Biodiversité (SIB) et le Water Information System for Europe (WISE) relatif au rapportage européen,

- de mettre librement accessible, utilisable et réutilisable le référentiel technique Sandre notamment à travers des services en réseau,
- de veiller à la cohérence du référentiel technique Sandre pour le SIE,
- de s'assurer de la bonne application du référentiel technique Sandre.

I.B.2. Les composantes du référentiel

Le référentiel technique Sandre est composé de :

- spécifications techniques "standards",
- jeux de données de référence,
- services en réseau.

I.B.3. Organisation du Sandre

Le SANDRE forme un réseau d'organismes contributeurs du SIE chargé de construire le langage commun des données sur l'eau (i.e. référentiel technique Sandre). L'Office français de la biodiversité (OFB) assure l'animation et la coordination technique de différents systèmes d'information dont le SIE sous la tutelle du ministère de la Transition écologique et de la Cohésion des territoires et du ministère de l'Agriculture et de la Souveraineté alimentaire. Le SANDRE comporte un groupe de pilotage, des administrateurs de données présents au sein des organismes contribuant au SIE, un secrétariat technique central assuré par l'Office International de l'Eau (OiEau). Pour de plus amples renseignements sur le ©Sandre, vous pouvez consulter le site Internet du ©Sandre : www.sandre.eaufrance.fr ou vous adresser à l'adresse suivante :

Secrétariat technique Sandre

Office International de l'Eau

15 rue Edouard Chamberland

87065 LIMOGES Cedex

Tél. : 05.55.11.47.90

E-mail : sandre@sandre.eaufrance.fr

I.C. Convention du document

I.C.1. Notations dans le document

Les termes DOIT, NE DOIT PAS, DEVRAIT, NE DEVRAIT PAS, PEUT, OBLIGATOIRE, RECOMMANDE, OPTIONNEL ont un sens précis. Ils correspondent à la traduction française de la norme RFC2119 ([RFC2119](#)) des termes respectifs MUST, MUST NOT, SHOULD, SHOULD NOT, MAY, REQUIRED, RECOMMENDED et OPTIONAL.

I.C.2. Gestion des versions

Chaque document publié par le Sandre comporte un numéro de version évoluant selon les règles suivantes :

Si cet indice est composé uniquement d'un nombre réel positif supérieur ou égal à 1.0 et sans la mention « beta », alors le document en question est une version approuvée par l'ensemble des acteurs en charge de sa validation. Il est publié sur le site internet du Sandre et est reconnu comme un document de référence, en particulier pour tout déploiement informatique.

Si cet indice est composé d'un nombre réel strictement inférieur à 1.0 (exemple : 0.2, 0.3...) ou bien supérieur ou égal à 1.0 avec la mention « beta » (exemple : 1.0beta, 1.1beta...), alors le document en question est une version provisoire. Il s'agit uniquement d'un document de travail. Il n'est donc pas reconnu par les acteurs en charge de sa validation et ne doit pas être considéré comme un document de référence. Ce document est susceptible de subir des révisions jusqu'à sa validation définitive.

Si un indice de version évolue uniquement d'une décimale (exemple : 1.0 à 1.1), alors il s'agit généralement de la prise en compte de modifications mineures dans le document en question (exemple : mise à jour de définitions, d'attributs, de règles de gestion...).

Si en revanche un indice de version change d'entier naturel (exemple : 1.0 à 2.0, 1.2 à 2.0), accompagné d'une décimale égale à 0, alors il s'agit généralement de la prise en compte de modifications majeures dans le document en question (exemple : mise à jour d'un ensemble d'entités, d'associations, de règles de gestion...).

Le document actuel est la version 1.0 et constitue un document Validé

II. DESCRIPTION DU FORMAT D'ÉCHANGE JSON APPLIQUÉ AU SANDRE

II.A. Préambule

Parmi les formats informatiques standardisés et libres d'utilisation employés par le SANDRE, il y a le format JSON. Le format JSON (JavaScript Object Notation) est un format de données léger et structuré, largement utilisé pour l'échange de données entre applications informatiques orientées web et les API (Application Programming Interface) interrogeable en mode REST. Le format JSON présente de nombreux avantages tels que ceux mentionnés ci-dessous :

- Simplicité et lisibilité : JSON offre une syntaxe concise et intuitive, ce qui facilite sa lecture et son écriture, notamment pour les développeurs.
- Légèreté et performance : JSON est globalement compact, ce qui réduit la taille des fichiers, accélère le transfert de données et réduit l'empreinte numérique. De plus, les moteurs d'analyse JSON sont rapides, améliorant ainsi les performances des applications.
- Support natif dans les langages de programmation : la plupart des langages de programmation modernes (JavaScript, Python, Java, etc.) incluent un support natif pour JSON, ce qui simplifie considérablement l'intégration de ce format dans les applications.
- Utilisation dans les API REST (Representational State Transfer) : JSON est le format d'échange de données standard pour les API REST en raison de sa simplicité, de sa légèreté et de son support natif dans JavaScript, le langage principalement utilisé pour le développement web côté client.

En revanche, le format JSON ne supporte pas nativement certains types de données complexes comme les dates ou les données spécifiques à un domaine. Elles sont représentées par des chaînes de caractères ou des structures complexes. Ces choix d'implémentation sont décrits dans les scénarios d'échanges Sandre.

JSON ne supporte pas nativement certaines fonctionnalités avancées comme les références croisées, les inclusions de fichiers, ou les transformations et vérifications de données. Afin d'assurer une compatibilité entre les systèmes, le SANDRE utilise des mécanismes décrits dans la suite de ce document.

II.B. Construction générale des fichiers JSON

II.B.1. Références normatives

Un fichier d'échange JSON appliqué au Sandre DOIT être un fichier constitué selon les règles définies par la norme [ECMA-404 2nd Edition – décembre 2017](#) et le standard [RFC 8259 – décembre 2017](#), compatibles entre elles.

II.B.2. Encodage des fichiers JSON

Conformément au standard RFC 8259, l'encodage des fichiers JSON DOIT être UTF-8, afin de garantir l'interopérabilité des données. Cette information NE DOIT PAS être indiquée par un indicateur d'ordre d'octets ou BOM (de l'anglais Byte Order Mark) en début de fichier.

II.B.3. Règles de nommage

Le SANDRE n'impose pas de noms particuliers aux fichiers générés selon la spécification JSON. En revanche, des règles de nommage de fichiers d'échange PEUVENT être spécifiées dans les scénarios d'échange.

II.B.4. Extension des fichiers

Il est OBLIGATOIRE que les extensions de fichiers utilisées par les acteurs informatiques soient respectées :

- Fichier JSON : *[nom du fichier].json*
- Schéma JSON : *[nom du fichier].schema.json*

A titre indicatif, il existe d'autres extensions applicables à d'autres spécifications dont le GeoJSON et le JSON-LD qui reposent sur le JSON. Les extensions de fichiers retenues pour les normes GEOJSON et JSON-LD sont respectivement ".geojson" et ".jsonld", comme mentionnées dans la norme RFC-7946 et dans les recommandations W3C.

II.B.5. Type MIME des fichiers

Le type Multipurpose Internet Mail Extensions (type MIME) est un standard permettant d'indiquer la nature et le format d'un document. Il est défini au sein de la RFC 6838 (l'Internet Assigned Numbers Authority (IANA) est l'organisme officiel responsable du suivi de l'ensemble des types MIME officiels existants). Les types MIME à respecter DOIVENT obligatoirement correspondre à ceux mentionnés ci-dessous :

- Fichier JSON : application/json
- Schéma JSON : application/schema+json

II.B.6. Compression des fichiers JSON

Le SANDRE AUTORISE :

- la suppression des espaces, des retours à la ligne et indentations inutiles,
- la compression au format multipart/x-gzip pour les fichiers volumineux.

En plus des principes de construction propres au format JSON, le fichier d'échange au format JSON DOIT être valide au regard des principes de construction des objets JSON appliqué au Sandre , décrit dans la suite du présent chapitre. L'organisation des objets JSON entre eux au sein d'un fichier d'échange DOIT correspondre aux principes définissant une arborescence JSON, décrit également dans la suite de ce chapitre. La structure des fichiers d'échange Sandre au format JSON DOIT respecter les spécifications décrites dans le chapitre dédié et inclure les informations mentionnées correctement positionnées. Enfin, un fichier d'échange au format JSON DOIT être valide au regard du scénario d'échanges Sandre auquel il est rattaché.

II.C. Types de données JSON

La notation JSON définit et prend en charge les types de données de base suivants :

II.C.1. Chaîne de caractères

Dans le format JSON, les chaînes de caractères sont placées entre guillemets doubles et peuvent contenir n'importe quel caractère Unicode. Elles sont couramment utilisées pour stocker et transmettre des données textuelles, telles que des noms, des adresses ou des descriptions.

L'utilisation de caractères spéciaux dans les chaînes de caractères doit être précédée d'une barre oblique inverse (\) pour l'échappement, par exemple :

- Les guillemets dans les chaînes : "Il dit, \"Bonjour!\""
- Les barres obliques inverses dans un chemins d'accès: "C:\\Documents\\Nouveau"
- Les caractères de contrôle tels que les tabulations (\t), les sauts de ligne (\n)...

Pour plus d'informations sur les caractères spéciaux, le lecteur est invité à se reporter au chapitre suivant de la norme JSON : <https://datatracker.ietf.org/doc/html/rfc8259#section-7>

Une chaîne de caractères PEUT être vide, auquel cas cela s'écrit "".

Exemples de chaînes de caractère non vides :

```
"CdIntervenant"
```

```
"Jane Smith"
```

II.C.2. Nombre

Les nombres sont utilisés pour stocker des valeurs numériques à des fins diverses, telles que des calculs, des comparaisons ou une analyse de données. JSON prend en charge les nombres entiers positifs et négatifs, ainsi que les nombres réels qui comportent des décimales. Le séparateur décimal DOIT obligatoirement être le point. Attention, pour que le type de données numérique soit reconnu, un nombre ne DOIT PAS être encadré par des guillemets. Il ne DOIT PAS non plus commencer par 0 à moins d'être suivi OBLIGATOIREMENT par le séparateur décimal le point. 01 n'est pas un nombre.

Exemples de nombres :

```
13000938400020
```

```
-1.25
```

II.C.3. Valeurs booléennes

Il existe uniquement deux valeurs booléennes : true (vrai) ou false (faux). Les valeurs booléennes ne sont pas entourées de guillemets et sont traitées comme des chaînes de caractères. Attention, les valeurs 0 et 1 ne sont pas reconnues comme des valeurs booléennes dans le format JSON.

Exemples :

```
true
```

```
false
```

II.C.4. Type null

null représente un type de données particulier. Il n'accepte qu'une seule valeur possible, la valeur *null*. Lorsqu'aucune valeur n'est affectée à une clé, elle peut être traitée comme étant nulle. Une valeur nulle signifie qu'elle n'est pas définie, tandis qu'une valeur vide signifie qu'elle a été définie et qu'elle est vide.

Attention, *null* DOIT s'écrire en minuscule et ne DOIT PAS être encadré par des guillemets.

II.C.5. Tableau

Un tableau représente un ensemble ordonné de valeur(s), encadré(es) par des crochets []. Un tableau PEUT comporter 0 à N valeurs, les valeurs étant séparées entre elles par le caractère virgule suivi éventuellement de caractère(s) espace(s). Un tableau PEUT être vide (dépourvu de valeurs). La taille d'un tableau ou objet JSON est infini ou illimité (absence de limite).

Dans le format JSON, les valeurs de tableau PEUVENT être :

- des chaînes de caractères
- des nombres
- des objets JSON
- des tableaux
- des valeurs booléennes
- des valeurs nulles

Exemples de tableaux :

Tableau vide :

```
[]
```

Tableau comportant trois valeurs correspondant à trois chaînes de caractères :

```
["rouge", "vert", "bleu"]
```

Tableau comportant quatre valeurs correspondant à quatre nombres :

```
[0, 1, 2, 3]
```

Tableau comportant trois valeurs correspondant à trois valeurs nulles :

```
[null, null, null]
```

Tableau comportant cinq valeurs, correspondant à trois valeurs booléennes, une chaîne de caractères, et un nombre :

```
[true, true, false, "bleu", 1]
```

Tableau comportant deux tableaux lesquels comportent deux valeurs correspondant à des nombres :

```
[[1,2],[1,3]]
```

Tableau comportant deux valeurs correspondant à deux objets JSON :

```
[  
  {"CdIntervenant" : "1470"},  
  {"NomIntervenant" : "SANDRE"}  
]
```

II.C.6. Objet JSON

Un objet JSON est délimité par des accolades { } et PEUT comporter 0 à N propriétés d'objet (ou membres), chaque propriété d'objet étant constituée d'un couple d'informations représenté par une clef associée à une valeur. Si un objet JSON comporte plusieurs propriétés d'objet, alors celles-ci sont séparées par une virgule.

Un caractère deux-points est placé entre chaque clef et sa valeur.

Exemple d'un objet JSON comportant 3 propriétés d'objet :

```
{  
  "clef1": valeur1,  
  "clef2": valeur2,  
  "clef3": valeur3  
}
```

La clef DOIT obligatoirement être une chaîne de caractères encadrée par des guillemets doubles. Une clef est unique à l'échelle de l'objet JSON, il ne DOIT PAS exister deux clefs comportant le même nom auquel cas l'objet JSON est mal formé.

La valeur associée à une clef est une valeur correspondant à un type de données JSON précédemment décrits, les types possibles étant :

- une chaîne de caractères encadrée par des guillemets doubles
- un nombre
- une valeur booléenne (true ou false)
- un tableau
- un objet JSON
- la valeur *null*

A noter qu'il est possible de spécifier qu'une clef puisse accepter plusieurs valeurs correspondant à des types de données de base différents (exemple : il est possible de spécifier que le nom d'un intervenant puisse être une chaîne de caractères ou bien la valeur *null*).

Exemple d'objet JSON bien formé et comportant quatre propriétés d'objet :

```
{  
  "Nom" : "DUPONT",  
  "Prenom" : "Jean",  
  "Age" : 23,  
  "Email" : "jean.dupont@domaine.com"  
}
```

II.D. Particularités des objets JSON définis par le Sandre

Les objets JSON décrivent les informations transmises dans le cadre d'un scénario d'échange Sandre. Ils DOIVENT être construits en respectant les conventions d'écriture du format JSON et en adoptant la structure suivante :

Valeur associée à la clé

```
{  
  "NOM DE CLE" : "VALEUR"  
}
```

Exemple : Objet JSON relatif à un intervenant et comportant deux propriétés d'objet JSON (couple clé / valeur) :

```
{  
  "Intervenant": {  
    "CdIntervenant": "11006801200050",  
    "schemeAgencyID": "SIRET"  
  }  
}
```

Avec :

- **NOM DE CLE** : Nom attribué à la clé selon les contraintes définies par le format JSON et le scénario d'échange Sandre.
- **VALEUR** : Informations à échanger selon les règles définies dans le scénario d'échange Sandre correspondant. Les informations à échanger peuvent être un objet JSON, une liste ou une chaîne de caractères, etc.

II.E. Ordre d'apparition des propriétés d'un objet JSON

Bien que les normes JSON et JSON Schema ne gèrent pas la notion d'ordre ou séquençage d'apparition des propriétés dans un objet JSON, il est fortement RECOMMANDE de respecter tout au moins l'ordre défini dans les scénarii d'échange Sandre afin de permettre une plus grande lisibilité des fichiers d'échange au format JSON appliqué au Sandre.

II.F. Multiplicité d'objets JSON

Lorsque plusieurs objets JSON peuvent être multiples, cela DOIT faire l'objet de la création d'un tableau d'objets JSON, conformément aux recommandations du standard RFC 8259.

Exemple : Un tableau comportant 3 objets JSON relatifs à trois communes distinctes

```
{
  "Commune": [
    {
      "CdCommune": "34210",
      "LbCommune": "Le Pouget"
    }, {
      "CdCommune": "80213",
      "LbCommune": "Cottenchy"
    }, {
      "CdCommune": "83054",
      "LbCommune": "La Farlède"
    }
  ]
}
```

II.G. Caractère obligatoire ou facultatif d'une propriété d'un objet JSON

Dans chaque objet JSON défini au sein d'un scénario d'échange Sandre, chaque propriété d'objet est soit OBLIGATOIRE, soit FACULTATIVE. Les spécifications JSON permettent de ne pas échanger les propriétés d'objet facultatives dans le fichier d'échange. Lorsqu'une propriété d'objet est FACULTATIVE, il est donc possible d'omettre la paire de clé-valeur associée.

En revanche, lorsqu'une propriété d'objet est OBLIGATOIRE au sein d'un scénario d'échange Sandre, cela signifie que la clef correspondante DOIT OBLIGATOIREMENT être présente ainsi que sa valeur associée, qui quant à elle DOIT correspondre au(x) type(s) de données définis pour cette propriété. Cette valeur DOIT également être non vide.

II.H. Structure des fichiers JSON

II.H.1. Racine d'un fichier JSON

Un fichier JSON bien formé ne DOIT comporter qu'un seul élément à la racine du document. L'élément racine PEUT être :

- la valeur null
- un nombre
- une valeur booléenne (true ou false)
- une chaîne de caractères
- un tableau
- un objet JSON

Par définition, un fichier JSON vide (ne comportant aucun caractère) est également bien formé.

Par convention, tout fichier JSON appliqué au Sandre DOIT comporter OBLIGATOIREMENT un seul et unique élément racine correspondant à un objet JSON (délimité par des accolades). Attention, il DOIT bien s'agir d'un objet JSON et non pas d'un tableau.

Le contenu suivant n'est pas un fichier JSON bien formé car il comporte deux objets JSON au même niveau racine :

```
{
  "Intervenant": {
    "CdIntervenant": "11006801200050",
    "NomIntervenant": "MINISTERE DE LA TRANSITION ECOLOGIQUE"
  }
},
{
  "Intervenant": {
    "CdIntervenant": "21006801200080",
    "NomIntervenant": "IFREMER"
  }
}
```

Les fichiers d'échange au format JSON appliqué au Sandre contiennent des objets décrivant les données à échanger, ainsi que des objets décrivant les métadonnées du fichier. Ces derniers assurent la qualité et la sécurité de l'échange. L'organisation des objets dans un fichier JSON appliqué au Sandre DOIT suivre un ordre précis et contenir des objets spécifiques, les métadonnées figurent dans l'en-tête et les données d'échanges constituent le corps de l'objet.

Les fichiers d'échange au format JSON appliqué au Sandre DOIVENT donc répondre à des modalités d'échange qui sont définies au sein des scénarios d'échange Sandre. La structure globale du fichier d'échange est la suivante :

```
{
```

```
    EN-TÊTE,
```

```
    CORPS
```

```
}
```

II.H.2. En-tête du fichier JSON

[En-tête] : L'en-tête d'un fichier JSON se rapportant à un scénario d'échange SANDRE DOIT inclure les métadonnées descriptives du fichier d'échange, à savoir la référence au schéma JSON correspondant au scénario (adresse URL de localisation du schéma JSON). Cette référence se fait au moyen de la propriété « schema » située dans l'objet JSON racine du fichier JSON.

Une propriété d'objet « Scenario » située dans l'objet JSON racine du fichier d'échange et contenant les références du scénario d'échange auquel le fichier JSON se rapporte, tels que le code, la version et le nom du scénario, mais aussi la date de création du fichier JSON, l'identité de l'émetteur du fichier JSON.

Exemple :

```
{
```

```
  "schema" : "https://json.sandre.eaufrance.fr/scenario/referentiel/4/  
             sandre\_sc\_referentiel.schema.json",
```

```
  "Scenario": {
```

```
    "CodeScenario": "REFERENTIEL",
```

```
    "VersionScenario": "4",
```

```
    "NomScenario": "Diffusion des référentiels par le Sandre",
```

```
    "DateCreationFichier": "2025-06-09",
```

```
    "DateDebutReference": "2025-05-01",
```

```
    "DateFinReference": "2025-05-31",
```

```
    "ReferenceFichierEnvoi": "REFERENTIEL\_06\_2025.json",
```

```
    "Emetteur": {
```

```
      "CdIntervenant": "1470",
```

```
      "schemeAgencyID": "SANDRE",
```



```

    "NomIntervenant": "Service d'Administration Nationale des Données et Référentiels sur
l'Eau"

    }

    },

    ...

}

```

Pour éviter des problèmes de compatibilité de référentiels entre deux acteurs de l'échange, une liste composée d'au moins un objet {Referentiel} PEUT figurer dans l'objet {Scenario} présent dans le schéma du fichier d'échanges. Cet objet permet d'identifier les référentiels par défaut utilisés dans la suite du fichier. Il DOIT inclure les attributs suivants :

- **CdReferentiel** : le code du référentiel
- **schemeAgencyID** : l'organisme chargé de cette codification
- **version** : la version du référentiel utilisé

Il DOIT être indiqué autant d'objets {Referentiel} dans l'en-tête que de référentiels utilisés dans le corps du fichier JSON.

Exemple : Echange de résultats appuyés sur 2 référentiels différents :

```

{
  "Scenario": {
    ...
    "Referentiel": [
      {
        "CdReferentiel": "PAR",
        "schemeAgencyID": "SANDRE",
        "version": "2004-05-30"
      },{
        "CdReferentiel": "SEL",
        "schemeAgencyID": "IFREMER",
        "version": "2004-05-30"
      }
    ]
  }
}

```

```
}  
]  
}  
}
```

Par la suite, l'utilisation d'un référentiel PEUT être indiquée dans la partie [Informations complémentaires] de l'objet s'appuyant sur ce référentiel, ou être défini par défaut.

II.1. Caractère « bien formé » et « valide » d'un fichier JSON

Un fichier JSON est dit « bien formé » dès lors qu'il respecte l'ensemble des règles syntaxiques de la norme JSON en vigueur, actuellement celle de référence RFC-8259 (décembre 2017).

Un fichier JSON est dit « valide » dès lors qu'il respecte à la fois la norme JSON en vigueur, mais aussi l'ensemble des règles structurelles et les contraintes spécifiées dans un ou plusieurs schémas JSON.

Il existe de nombreux outils disponibles (site web en ligne, extensions de navigateurs internet, environnements de développement intégrés), pour faciliter le contrôle des fichiers JSON, permettant de s'assurer de leur caractère « bien formé » et « valide ».

Ces étapes de vérification sont essentielles pour une bonne qualité des échanges de données entre partenaires d'échange, via des fichiers au format JSON.

Le Sandre diffuse librement un ou plusieurs schémas JSON pour chaque scénario d'échange, afin d'aider les acteurs à se conformer rigoureusement aux règles syntaxiques et structurelles définies pour chaque scénario d'échange.

II.J. Exemple complet d'un fichier d'échange JSON appliqué au Sandre

```
{
  "schema": "https://json.sandre.eaufrance.fr/scenario/labo_dest/1.1/san-
dre_sc_labo_dest.schema.json",
  "Scenario": {
    "CodeScenario": "LABO_DEST",
    "VersionScenario": "1.1",
    "NomScenario": "Echanges informatisés entre Laboratoires et Commanditaires",
    "DateCreationFichier": "2025-02-21",
    "ReferenceFichierEnvoi": "labo_dest_250221_140028.json",
    "Emetteur": {
      "CdIntervenant": "22850001300765",
      "schemeAgencyID": "SIRET"
    },
    "Destinataire": {
      "CdIntervenant": "13000938400020",
      "schemeAgencyID": "SIRET"
    },
  },
  "Demande": {
    "Commanditaire": {
      "CdIntervenant": "13000938400020",
      "schemeAgencyID": "SIRET"
    },
    "CdDemandePrestataire": "L.2025.5636",
    "Prestataire": {
      "CdIntervenant": "22850001300765",
      "schemeAgencyID": "SIRET"
    }
  }
}
```

```
    },
    "TypeDemande": "3",
    "ContexteCodification": "1",
    "DateDemande": "2025-02-17",
    "DestinataireRsAna": {
      "CdIntervenant": "13000938400020",
      "schemeAgencyID": "SIRET"
    },
    "Prelevement": [{
      "RealisePrel": "1",
      "DatePrel": "2025-02-17",
      "HeurePrel": "14:30:00",
      "ConformitePrel": "1",
      "FinalitePrel": "25",
      "AccredPrel": "1",
      "PrelSousReserve": "0",
      "CommentairesPrel": "",
      "StationPrelevement": {
        "CdStationPrelevement": "85.08.03",
        "schemeAgencyID": "13"
      },
      "LocalPrelevement": {
        "CdLocalPrelevement": "60011635",
        "schemeAgencyID": "4"
      },
      "LocalExactePrel": "076 - Pertuis Breton - Prise du Bois (B) 85.08.03",
      "CoordXPrel": -1.264375,
      "CoordYPrel": 46.290545,
      "Support": {
        "CdSupport": "75"
```

```
    },
    "MethodePrel": {
      "CdMethode": "0",
      "NomMethode": "Inconnue"
    },
    "Preleveur": {
      "CdIntervenant": "22850001300765",
      "schemeAgencyID": "SIRET"
    },
    "Payeur": {
      "CdIntervenant": "33071536800107",
      "schemeAgencyID": "SIRET"
    },
    "Echantillon": [{
      "RefEchantillonLabo": "L.2025.5636-1-1",
      "DateReceptionEchant": "2025-02-18",
      "HeureReceptionEchant": "13:50:00",
      "CommentairesEchant": "",
      "Laboratoire": {
        "CdIntervenant": "22850001300765",
        "schemeAgencyID": "SIRET"
      },
      "CompletEchant": "1",
      "Analyse": [{
        "DateAna": "2025-02-18",
        "HeureAna": "17:00:00",
        "RsAna": 45,
        "RqAna": "1",
        "AccreAna": "2",
        "InsituAna": "2",
```

```
"Parametre": {  
  "CdParametre": "1449",  
  "NomParametre": "Escherichia coli (E. coli)"  
},  
"FractionAnalysee": {  
  "CdFractionAnalysee": "250"  
},  
"Methode": {  
  "CdMethode": "1234"  
},  
"UniteReference": {  
  "CdUniteReference": "221"  
},  
"Commemoratif": [{  
  "CdCommemoratif": "32",  
  "LbCommemoratif": "Difficulté de l'analyse",  
  "ValCommemoratif": "2"  
  }  
  ]  
},  
"Commemoratif": [{  
  "CdCommemoratif": "14",  
  "LbCommemoratif": "Type de prélèvement",  
  "DsCommemoratif": "",  
  "ValCommemoratif": "289"  
  }  
  ]  
}
```


III. TABLE DES MATIERES

I.	AVANT PROPOS	4
I.A.	LE SYSTEME D'INFORMATION SUR L'EAU ET LE SANDRE	4
I.B.	LE SANDRE	5
I.B.1.	Les missions du Sandre	5
I.B.2.	Les composantes du référentiel	5
I.B.3.	Organisation du Sandre	5
I.C.	CONVENTION DU DOCUMENT	6
I.C.1.	Notations dans le document	6
I.C.2.	Gestion des versions	6
II.	DESCRIPTION DU FORMAT D'ÉCHANGE JSON APPLIQUÉ AU SANDRE	7
II.A.	PREAMBULE	7
II.B.	CONSTRUCTION GENERALE DES FICHIERS JSON	8
II.B.1.	Références normatives	8
II.B.2.	Encodage des fichiers JSON	8
II.B.3.	Règles de nommage	8
II.B.4.	Extension des fichiers	8
II.B.5.	Type MIME des fichiers	8
II.B.6.	Compression des fichiers JSON	9
II.C.	TYPES DE DONNEES JSON	9
II.C.1.	Chaîne de caractères	9
II.C.2.	Nombre	10
II.C.3.	Valeurs booléennes	10
II.C.4.	Type null	10
II.C.5.	Tableau	10
II.C.6.	Objet JSON	11
II.D.	PARTICULARITES DES OBJETS JSON DEFINIS PAR LE SANDRE	13

II.E.	ORDRE D'APPARITION DES PROPRIETES D'UN OBJET JSON	13
II.F.	MULTIPLICITE D'OBJETS JSON.....	14
II.G.	CARACTERE OBLIGATOIRE OU FACULTATIF D'UNE PROPRIETE D'UN OBJET JSON	14
II.H.	STRUCTURE DES FICHIERS JSON	15
II.H.1.	<i>Racine d'un fichier JSON</i>	15
II.H.2.	<i>En-tête du fichier JSON</i>	16
II.I.	CARACTERE « BIEN FORME » ET « VALIDE » D'UN FICHIER JSON	18
II.J.	EXEMPLE COMPLET D'UN FICHIER D'ÉCHANGE JSON APPLIQUE AU SANDRE	19