
SpirHAL Documentation

Version stable

janv. 15, 2023

1	Pré-requis	3
2	Installation	5
2.1	Récupération du script	5
2.2	Importer les fichiers sur votre page web	6
3	Affichage pour une personne	7
3.1	Page complète	8
4	Affichage pour une structure	9
5	Installation dans une application React	11
5.1	Pré-requis	11
5.2	Installation	11
5.3	Affichage pour une personne	11
5.4	Affichage pour une structure	12
6	Modifier SpirHAL	13
6.1	Modifier une norme bibliographique	13
6.2	Ajouter une norme bibliographique	13
7	Présentation du projet	17
7.1	Affichage pour une personne	17
7.2	Affichage pour une structure	17
8	Pour commencer	19
9	À propos	21



le code source de SpirHAL est disponible [ici](#).

Plan de la documentation :

CHAPITRE 1

Pré-requis

Pour installer l'affiche des notices de HAL pour un membre il est nécessaire d'avoir un site hébergé. Ce module vient s'intégrer dans la page pour afficher les notices d'un membre donné ou d'une structure donnée. Aucune autre pré requis n'est nécessaire.

2.1 Récupération du script

Pour récupérer SpirHAL vous avez deux possibilités :

2.1.1 CDN

La première possibilité est d'utiliser directement la dernière version de SpirHAL sur un serveur qui l'héberge :

```
https://cdn.jsdelivr.net/npm/spirhal@latest/dist/bundle.js
```

Si pour des raisons de rétro-compatibilité vous souhaitez utiliser une version spécifique de SpirHAL, il est possible de récupérer cette version en utilisant l'URL suivante :

```
https://cdn.jsdelivr.net/npm/spirhal@{VERSION}/dist/bundle.js
```

Par exemple pour récupérer la version *v4.0.1* vous pouvez utiliser l'URL suivante :

```
https://cdn.jsdelivr.net/npm/spirhal@v4.0.1/dist/bundle.js
```

Cette méthode peut permettre de fixer une version et éviter des problème de compatibilité lors des prochaines mises à jours de SpirHAL.

2.1.2 Compilation

La dernière possibilité est de compiler vous même SpirHAL. Pour compiler vous même le script il est nécessaire d'avoir installé [Node.JS](#) et [GIT](#) sur votre ordinateur.

Tout d'abord il vous faut cloner le git du projet SpirHAL :

```
git clone git@framagit.org:Murloc6/Spirhal.git
```

Une fois le contenu du projet récupéré, il faut se rendre dans le dossier et lancer la commande de compilation :

```
cd Spirhal
npm ci
npm run build:prod
```

Lorsque le processus est terminé, les fichiers compilés de SpirHAL sont disponibles dans le dossier dist. Notamment le fichier *bundle.js* qui contient tout ce qui est nécessaire pour faire fonctionner SpirHAL.

2.2 Importer les fichiers sur votre page web

Maintenant que vous avez les fichiers compilés de SpirHAL (soit par CDN soit en les compilant vous même) il vous faut les importer dans le code de votre page web. Pour cela il faut rajouter les balises d'import dans le head de votre page web. Par exemple si nous avons utilisé la version hébergée sur le serveur CDN, le script à importer est :

```
<script src="https://cdn.jsdelivr.net/npm/spirhal@latest/dist/bundle.js"></script>
```

Ce script importé contient toutes les dépendances nécessaires à SpirHAL pour fonctionner.

Affichage pour une personne

Une fois le fichier importé il faut maintenant initialiser SpirHAL. Pour cela, SpirHAL utilise le principe de “Web-Component” (plus d’informations [ici](#)). Ce qui signifie que l’on utilise une balise spécifique pour SpirHAL avec les paramètres correspondants. Ici la balise s’appelle “spirhal-app”. Nous pouvons alors insérer dans notre page cette balise et la paramétrer pour qu’elle affiche les notices de la personne concernée. Par exemple si nous souhaitons afficher les notices de “Hélène Débax” du laboratoire Framespa (structID HAL 620) avec la norme “EHESS” alors nous pouvons insérer la balise suivante :

```
<spirhal-app researcherName="Hélène Débax" norm="EHESS" structId="620"/>
```

- On peut remarquer le nom de la balise est bien “spirhal-app” et ensuite les paramètres sont des attributs de la balise :
- “researcherName” est la personne dont on souhaite afficher les notices venant de HAL (attention de bien respecter “Prénom Nom” dans cet ordre avec ces majuscules pour que la requête sur HAL fonctionne correctement)
 - “norm” est le nom de la norme bibliographique à utiliser (seul les normes implémentées peuvent être utilisées). S’il n’y a pas de norme spécifiée alors SpirHAL utilisera la représentation utilisée sur HAL
 - “structId” est le structId sur HAL du laboratoire auquel appartient la personne (ce paramètre est optionnel)
 - il est possible de connaître le structId d’un laboratoire sur [AuréHAL](#)

Une fois cette balise ajoutée, les notices provenant de la personne spécifiée dans “researcherName” seront affichées à l’endroit où est la balise.

Les paramètres suivants sont disponibles comme attributs de la balise :

- **structId : Identifiant HAL du laboratoire de la personne**
 - peut être une liste d’identifiants HAL (e.g. *structId= »[620,42] »*)
 - peut être laissé vide s’il n’est pas connu
- **norm : nom de la norme bibliographique dans laquelle les notices doivent être présentées**
 - choix entre APA, MLA, EHESS
 - possibilité de laisser vide pour la citation de HAL par défaut
- **idHal** : Identifiant HAL de la personne
- **researcherName : nom complet de la personne**
 - utilisé si aucun identifiant HAL n’est renseigné
- **portal : si un portail HAL est à renseigner pour la recherche**
 - peut-être laissé vide
- **adminLink** : lien du site d’administration SpirHAL pour la personne, s’il est déployé

- **noGrouping** : si ajouté, les publications seront toutes affichées sans être groupées par type de document
 - e.g. `<spirhal-app ... noGrouping />`

3.1 Page complète

Un exemple de page complète pour “Hélène Débax” et ne contenant que la liste des notices est la suivante :

```
<html>
  <head>
    <script src="https://cdn.jsdelivr.net/npm/spirhal@latest/dist/bundle.js"></
↪script>
  </head>
  <body>
    <spirhal-app researcherName="Hélène Débax" norm="EHESS" structId="620"/>
  </body>
</html>
```

Affichage pour une structure

Pour afficher les notices provenant d'une structure plutôt que d'une personne (pour par exemple afficher toutes les notices sur la page d'accueil d'un laboratoire) il est possible d'utiliser la balise *spirhal-app-structure* :

```
<spirhal-app-structure struct-id="620" norm="EHESS" limit="20" noGrouping />
```

Les paramètres suivants sont disponibles sur cette balise :

- **structId** : **Identifiant HAL de la structure**
 - nécessaire
- **norm** : **nom de la norme bibliographique dans laquelle les notices doivent être présentées**
 - choix entre APA, MLA, EHESS
 - possibilité de ne pas le renseigner pour la citation de HAL par défaut
- **limit** : nombre de notices maximales à récupérer par type de notice
- **portal** : **si un portail HAL est à renseigner pour la recherche des notices**
 - peut-être laissé vide
- **adminLink** : lien du site d'administration SpirHAL, s'il est déployé
- **noGrouping** : **si ajouté, les publications seront toutes affichées** sans être groupées par type de document

Installation dans une application React

5.1 Pré-requis

Ce mode d'utilisation de la bibliothèque nécessite d'être intégré dans une application React déjà existante avec une chaîne de compilation déjà configurée ([tutoriel React](#)).

5.2 Installation

L'application React est disponible sur npmjs.com. Il est donc possible de l'installer directement avec npm :

```
npm install --save spirhal
```

Ou bien encore avec yarn :

```
yarn add --save spirhal
```

5.3 Affichage pour une personne

De la même manière que pour l'utilisation sur une page Web, il suffit d'importer le composant et de l'insérer dans le JSX :

```
import React from "react";
import {SpirhHAL} from "spirhal";

function App() {
  return (
    <div>
      <Spirhal idHal="42" noGrouping />
    </div>
  );
}
```

(suite sur la page suivante)

```

    );
}

```

Les paramètres disponibles sont :

- **structId : Identifiant HAL du laboratoire de la personne**
 - peut être une liste d'identifiants HAL (e.g. *structId= »[620,42] »*)
 - peut être laissé vide s'il n'est pas connu
- **norm : nom de la norme bibliographique dans laquelle les notices doivent être présentées**
 - choix entre APA, MLA, EHESS
 - possibilité de laisser vide pour la citation de HAL par défaut
- **idHal** : Identifiant HAL de la personne
- **researcherName : nom complet de la personne**
 - utilisé si aucun identifiant HAL n'est renseigné
- **portal : si un portail HAL est à renseigner pour la recherche**
 - peut-être laissé vide
- **adminLink** : lien du site d'administration SpirHAL pour la personne, s'il est déployé
- **noGrouping : si ajouté, les publications seront toutes affichées sans être groupées par type de document**
 - e.g. *<spirhal ... noGrouping />*

5.4 Affichage pour une structure

De la même manière il est possible d'afficher les notices pour une structure donnée :

```

import React from "react";
import SpirhalStructure from "spirhal";

function App() {
  return (
    <div>
      <SpirhalStructure struct-id="620" norm="EHESS" limit="20" noGrouping />
    </div>
  );
}

```

Les paramètres suivants sont disponibles sur cette balise :

- **structId : Identifiant HAL de la structure**
 - nécessaire
- **norm : nom de la norme bibliographique dans laquelle les notices doivent être présentées**
 - choix entre APA, MLA, EHESS
 - possibilité de ne pas le renseigner pour la citation de HAL par défaut
- **limit** : nombre de notices maximales à récupérer par type de notice
- **portal : si un portail HAL est à renseigner pour la recherche des notices**
 - peut-être laissé vide
- **adminLink** : lien du site d'administration SpirHAL, s'il est déployé
- **noGrouping : si ajouté, les publications seront toutes affichées sans être groupées par type de document**

Le projet SpirHAL étant du logiciel libre il est tout à fait possible de modifier tous les éléments de ce projet. Nous allons ici présenter comment modifier ou ajoute une nouvelle norme dans SpirHAL.

Le code source de SpirHAL est développé en utilisant la technologie [React](#) il est donc fortement recommandé de maîtriser cette technologie avant de proposer des modifications à SpirHAL.

6.1 Modifier une norme bibliographique

Chaque norme bibliographique est proposée comme un “Component” React. Toutes les normes sont présentes dans le dossier “/src/components/spirhal/norms”. Trois sont implémentées pour l’instant : EHESS, APA, MLA.

La logique de formatage d’une notice suivant une norme est contenue dans un sous-dossier de “/src/components/spirhal/norms” ayant le nom de la norme. Chaque norme doit fournir un composant React pour affichage d’une notice selon la norme bibliographique dont les propriétés implémentent l’interface *Notice* (« src/components/spirhal/Notice.tsx »).

Une norme exporte aussi une liste *renderedGroups* de types de notices qui nécessitent un traitement particulier par la norme pour leur affichage. Les autres types de notices sont traités « par défaut » par la norme.

```
export const renderedGroups = ["OUV", "COUV", "ART", "COMM", "DOUV", "POSTER", "THESE  
→", "HDR"];  
  
export const APANotice: React.FC<Notice> = (notice) => {  
  ....  
}
```

6.2 Ajouter une norme bibliographique

Pour ajouter une nouvelle norme nommée « EXEMPLE » dans SpirHAL, il faut créer un dossier “/src/components/spirhal/norms/EXEMPLE”. Dans celui-ci créer un fichier « EXEMPLE.tsx » qui exportera un com-

posant React comme suit et la liste des types de notices suivant un traitement particulier par la norme.

```
...
export const renderedGroups = ["OUV", "COUV", "ART", "COMM", "DOUV", "POSTER", "THESE
↔", "HDR"];

...

export const EXEMPLENotice: React.FC<Notice> = (notice) => {
  ....
}
```

D'autres fichiers peuvent être créés dans le dossier « EXEMPLE » pour contenir la logique propre à la norme.

Ensuite, dans « src/components/spirhal/Notice.tsx », il faut :

- importer le composant EXEMPLENotice
- ajouter la valeur à l'énumération Norm
- ajouter un cas dans le switch de GenericNoticeComponent

```
...
import { EXEMPLENotice } from "../norms/EXEMPLE/EXEMPLE";

...

enum Norm {
  EHESS = "EHESS",
  APA = "APA",
  MLA = "MLA",
  DEFAULT = "default",
  EXEMPLE = "EXEMPLE",
}

...

const GenericNoticeComponent: React.FC<Notice> = (notice) => {
  ...
  switch (norm) {
    case Norm.EXEMPLE:
      noticeComponent = <EXEMPLENotice {...notice} />;
      break;
    ...
  }
  ...
}
```

Dans le fichier « src/components/spirhal/Groups.tsx », il faut :

- importer le contenu de la norme (ou au moins la liste des renderedGroups)
- ajouter la liste des renderedGroups au dictionnaire *normRenderedGroups*

```
import * as EXEMPLE from "../norms/EXEMPLE/EXEMPLE";

...

const normRenderedGroups: { [key: string]: string[] } = {
  EHESS: EHESS.renderedGroups,
  APA: APA.renderedGroups,
```

(suite sur la page suivante)

(suite de la page précédente)

```
MLA: MLA.renderedGroups,  
EXEMPLE: EXEMPLE.renderedGroups,  
};
```

Il faudra enfin utiliser le nom de la norme lors de l'implémentation de SpirHAL pour utiliser cette nouvelle norme dans votre page de votre site web avec :

```
<spirhal researcherName="Hélène Débax" norm="EXEMPLE" structId="620"></spirhal>
```

Présentation du projet

SpirHAL est un logiciel libre et open-source en JavaScript/React permettant le déploiement d'un module d'affichage sur un site web des notices de publications scientifiques figurant sur la plateforme [HAL](#).

7.1 Affichage pour une personne

Ce mode permet l'affichage des notices des publications pour une personne en respectant une norme bibliographique donnée. Il peut être utilisé pour un affichage sur une fiche annuaire des membres d'un laboratoire.

7.2 Affichage pour une structure

Ce mode permet l'affichage des notices des publications pour une structure, identifiées par un structId dans HAL. Par exemple, FRAMESPA a le structId 620 dans HAL :

« https://hal.archives-ouvertes.fr/search/index/q/*/structId_i/620/ »

CHAPITRE 8

Pour commencer

Il y a deux possibilités d'intégrer SpirHAL. La première est l'utilisation par WebComponent pour insérer SpirHAL dans une page Web déjà existante :

[Documentation pour l'utilisation de WebComponent.](#)

La deuxième méthode est l'utilisation par un composant React donc la [documentation est disponible ici.](#)

CHAPITRE 9

À propos

le code source de SpirHAL est disponible [ici](#).

Ce projet est développé avec la technologie React, et est disponible sous forme de Web component, ce qui permet un déploiement facile et rapide sur tout site web. L'utilisation de cette technologie permet aussi la compatibilité avec la plupart des navigateurs.

Licence [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License](#)