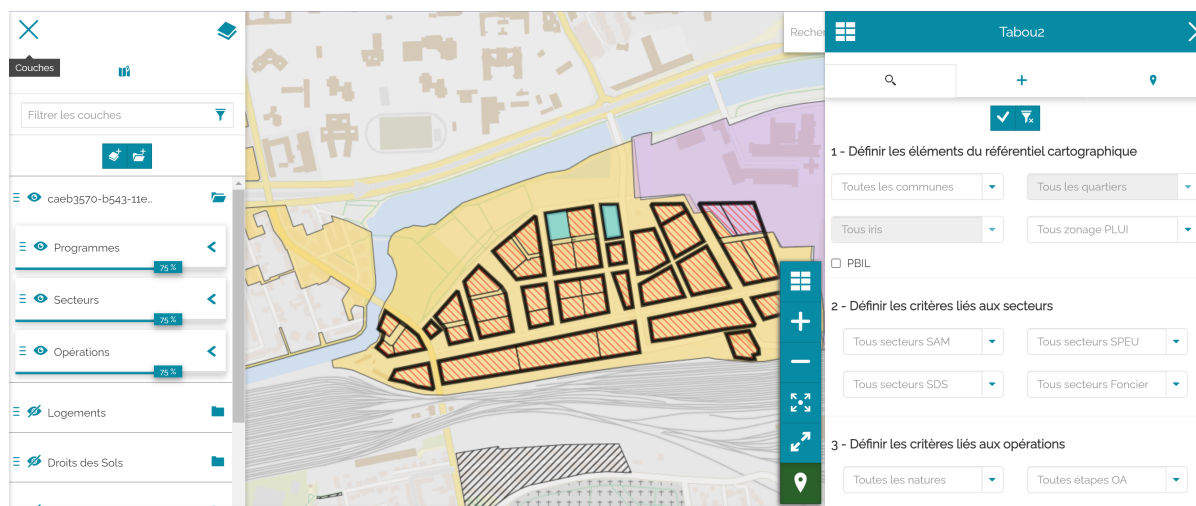

Tabou2

sigrennesmetropole gaetanbri

janv. 20, 2023

1	Contribuer	3
1.1	GitHub Workflow	3
1.2	Proposer une modification	4
1.3	Documentation	4
2	Généralités techniques	5
2.1	Installation	5
2.2	Build	5
2.3	Répertoires utiles	5
2.4	Répertoires du plugin	6
3	Configuration du plugin	7
3.1	Accès à la configuration	7
3.2	Paramètres disponibles	7
3.3	Explications détaillées	8
3.4	Styles de la sélection	9
3.5	Exemple complet	10
4	Documentation	15
4.1	Construire et déployer la documentation	15
4.2	Installer XAMPP (windows)	16
4.3	Documentation	16

Cette documentation Tabou2 aborde la configuration du module MapStore2, son installation et la description des ressources techniques du module.



1.1 GitHub Workflow

Pour contribuer, corriger ou modifier, ce dépôt respecte les bonnes pratiques du suivantes :

1. Les modifications doivent porter sur une branche spécifique
2. Les modifications sont centralisées vers une branche **develop** qui contient les modifications de la version en cours de développement
3. La branche master (ou main) contient le code de la dernière release
4. Chaque release correspond à un tag et une branche spécifique
5. Chaque pull request externe (correctif, évolution) doit cibler la branche **develop**



1.2 Proposer une modification

Pour proposer une correction d'anomalie ou une évolution, vous devez suivre ces étapes :

- Créer une issue sur Github
- Faire un fork du code
- Créer une branche dans votre fork portant le numéro de l'issue (ex : issue-2287)
- Apporter vos modifications sur cette branche
- Partager cette branche via l'issue pour que les autres puissent tester et obtenir des conseils ou des avis
- Réaliser une pull request via GitHub

La pull request permettra d'importer votre modification dans le code natif. Vous disposerez alors de votre modification de manière native sans vous en préoccuper ultérieurement.

1.3 Documentation

Pour mieux contribuer :

1. [Première contribution](#)
2. [Comment contribuer](#)

Généralités techniques

Cette section décrit les généralités à propos du fonctionnement du plugin et les principaux répertoires que vous trouverez dans le dépôt GitHub.

2.1 Installation

L'installation et les prérequis sont décrits dans le fichier README.md du plugin Tabou2 :

https://github.com/sigrennesmetropole/geor_tabou2_front/blob/main/README.md

2.2 Build

Le build est également décrit dans le fichier README.md.

Cette étape permet de générer un fichier .zip qui pourra être importé dans l'interface d'import d'un plugin MapStore2.

2.3 Répertoires utiles

Nous expliquons certains répertoires utiles.

- MapStore2

Ce répertoire est un sous-module qui contient l'ensemble de MapStore2. Grâce à ce répertoire, nous pouvons utiliser le framework et si besoin l'API MapStore2.

- assets

Ce répertoire contient les traductions et contiendra les fichiers CSS et autres ressources utiles.

- js

Ce répertoire contient le plugin Tabou2

- dist

Ce répertoire apparaîtra lorsqu'un premier build sera réalisé. il contiendra alors le fichier .zip à utiliser pour importer le plugin dans MapStore2 via l'interface prévu à cet effet.

2.4 Répertoires du plugin

Nous aborderons ici l'arborescence des répertoires du plugin Tabou2 dans le répertoire `/js/extension`.

- plugins

C'est le répertoire principale. Pour bien comprendre le plugin Tabou2, vous devez commencer par ce répertoire qui est chargé par MapStore2 en premier. Il contient l'emplacement du bouton d'accès au module et le composant parent en haut de la chaîne des composants.

Pour bien commencer, il est fortement conseiller de lire la documentation sur les plugins : <https://mapstore.readthedocs.io/en/latest/developer-guide/plugins-howto.html>

- api

Contient l'ensemble des méthodes et appels (URL) du plugin Tabou2 vers un service externe (geoserver, API). La librairie axios fournie par MapStore2 est utilisée.

- components

Contient tous les composants (fenêtres, comportements) du module.

- actions

Pour en savoir plus sur les actions :

<https://mapstore.readthedocs.io/en/latest/developer-guide/writing-actions-reducers.html> <https://redux.js.org/tutorials/fundamentals/part-3-state-actions-reducers>

- epics

Consulter <https://redux-observable.js.org/docs/basics/Epics.html> pour plus d'informations.

- reducers

Pour en savoir plus sur les reducers :

<https://mapstore.readthedocs.io/en/latest/developer-guide/writing-actions-reducers.html> <https://redux.js.org/tutorials/fundamentals/part-3-state-actions-reducers>

- selectors

Les selectors permettent de ne pas modifier et interagir directement avec l'état (props) des composants. L'ensemble des actions asynchrones (requêtes) doivent en majorité passer par un selectors.

Voici les bonnes pratiques :

<https://mapstore.readthedocs.io/en/latest/developer-guide/code-conventions.html#access-to-the-state-using-state-selectors>

- utils

Contient des méthodes utiles à l'ensemble du plugin.

- constant.js

ce fichier contient les constantes utiles et obligatoires pour le bon fonctionnement du plugin.

Configuration du plugin

Cette section vous permettra de comprendre la configuration du plugin en abordant tous les éléments essentiels de la configuration.

3.1 Accès à la configuration

Prérequis :

- Vous devez détenir les droits suffisants pour modifier un contexte MpaStore2-geOrchestra.
- Le plugin doit être installé (voir section « généralité technique »)

La configuration du plugin Tabou2 est accessible comme pour tous les autres plugins.

1. Vous devez passer le contexte en mode édition
2. Naviguez jusqu'à la section des plugins et voir les **plugins activés**
3. Recherchez le plugin « Tabou2 »
4. Ouvrir la configuration du module Tabou2 en cliquant sur **Modifier la configuration du plugin** :

Fig. 1 – Ouvrir la configuration Tabou2.

3.2 Paramètres disponibles

La configuration d'un plugin MapStore est disponible en JSON. Le formalisme JSON doit être respecté. Un exemple complet de la configuration est fourni dans la partie suivante.

Voici la liste des paramètres et les explications correspondantes.

3.3 Explications détaillées

Les filtres du plugin Tabou2 permettent de réaliser des intersection spatiale entre une couche présente ou non dans le contexte Tabou2 avec une couche PA, SA, OA.

Ces cas pratiques permettront de bien comprendre la configuration.

1. Quartiers

— Comportement attendu

Les valeurs des quartiers sont fournis par l'API Tabou2 afin d'être chargées et visibles dans la liste du filtre **Quartier**.

L'API fournit le numéro du quartier (utile pour la recherche dans la couche) et son nom (utile pour afficher le nom à la sélection).

Lors de la sélection d'un quartier, le plugin utilise la valeur sélectionnée (ex : Centre) pour réaliser une intersection spatiale (via geoserver cross layer) entre la couche quartier et chacune des couches programme, secteur et opération.

Le résultat permettra d'obtenir les IDs des entités de ces couches programme, secteur et opération qui intersectent le quartier sélectionné (ex : Centre).

Une fois les IDs récupérés, le plugin applique un filtre attributaire MapStore sur la couche OA, PA, SA selon les IDs récupérés et pour chaque couche.

— Configuration à appliquer

```

1 "quartiers": {
2   "layer": "ladm_terri:quartier",
3   "geom": "shape",
4   "placeholder": "Quartiers",
5   "filterField": "nuquart",
6   "apiField": "nuQuart",
7   "apiLabel": "nom"
8 },

```

— Explication de la configuration

Tableau 1 – Paramètres appliqués - filtre Quartier

Para-mètre	Explications
layer	Nous utiliserons la layer ladm_terri :quartier disponible dans le geoserver pour le croisement spatial
geom	Le champ géométrique est nommé shape dans geoserver (utile pour l'intersection)
place-holder	Si l'utilisateur vide la sélection, il sera affiché Quartier dans le filtre
filter-Field	Le champ de la couche ladm_terri :quartier qui contient la valeur du quartier sélectionné, permettant donc le filtre, est nommé nuquart
apiField	Le champ contenant la valeur à rechercher dans la couche est appelé nuQuart (différent du nom du champ dans la couche)
apiLabel	Le champ de l'API qui contient le label à afficher dans la liste du filtre est nommé nom

2.Date DOC

— Comportement attendu

L'utilisateur peut sélectionner une date et filtrer les couches programme, secteur, opération selon la date de début et de fin sélectionnée. La difficulté est que seule la couche programme contient cette information.

A la sélection d'une date, nous souhaitons donc :

- a. Récupérer les entités de la couche programme qui correspondent à ces dates

- b. Récupérer les couches operations et secteur qui intersectent les programmes et uniquement les programmes qui correspondent aux dates choisies
 - c. filtrer dans MapStore2 les couches selon les IDs récupérés par l'intersection
- Configuration à appliquer

```

1 "doc": {
2   "layer": "app:tabou_v_oa_programme",
3   "geom": "shape",
4   "filterField": "doc_date",
5   "type": "date",
6   "placeholder": "Choisir une date"
7 },

```

— Explication de la configuration

3.4 Styles de la sélection

Lorsqu'un utilisateur clique sur une ou plusieurs couches, la fiche information affiche :

1. Une liste des couches (visibles) cliquées qui retournent des features au clic
2. Une liste qui permet de sélectionner la feature à afficher pour la couche sélectionnée (non visible si une seule feature cliquée)

Fig. 2 – Styles de sélection Tabou2.

Le style des entités par défaut (style « default ») et le style de l'entité affichée dans la fiche (style « selection ») doit correspondre aux paramètres possibles pour le style OpenLayers. D'autres paramètres de style peuvent donc être ajoutés pour plus de personnalisation.

Enfin, en utilisant les paramètres d'opacité il est alors possible d'afficher l'entité en consultation dans la fiche et / ou les autres entités retournées par le clic pour la couche sélectionnée dans la liste.

— Exemple de paramètres :

```

1 "styles": {
2   "selection": {
3     "fillColor": "red",
4     "fillOpacity": 0.8,
5     "opacity": 1,
6     "color": "yellow",
7     "weight": 2
8   },
9   "default": {
10    "fillColor": "#6f718e",
11    "fillOpacity": 0,
12    "opacity": 0.8,
13    "color": "yellow",
14    "weight": 2,
15    "dashArray": [5, 5]
16  }
17 }

```

3.5 Exemple complet

Voici un exemple de configuration :

```

1  {
2      "cfg": {
3          "styles": {
4              "showClick": true,
5              "selection": {
6                  "fillColor": "red",
7                  "fillOpacity": 0.5,
8                  "opacity": 1,
9                  "color": "red",
10                 "weight": 4
11             },
12             "default": {
13                 "fillColor": "#6f718e",
14                 "fillOpacity": 0,
15                 "opacity": 0.8,
16                 "color": "red",
17                 "weight": 2,
18                 "dashArray": [
19                     5,
20                     5
21                 ]
22             }
23         },
24         "help": {
25             "url": "https://public.sig.rennesmetropole.fr/ressources/app/georchestra/Portail
↪%20web%20guide%20rapide%20de%20prise%20en%20main.pdf",
26             "contact": "d.cottencin@rennesmetropole.fr",
27             "agapeo": "https://public.sig.rennesmetropole.fr/ressources/app/georchestra/
↪Portail%20web%20guide%20rapide%20de%20prise%20en%20main.pdf",
28             "ddc": "https://public.sig.rennesmetropole.fr/ressources/app/georchestra/Portail
↪%20web%20guide%20rapide%20de%20prise%20en%20main.pdf"
29         },
30         "geoserverURL": "/geoserver",
31         "showIdentify": false,
32         "layersOrder": [
33             "Opérations",
34             "Secteurs",
35             "Programmes"
36         ],
37         "layersCfg": {
38             "layerPA": {
39                 "nom": "tabou_test_v_oa_programme",
40                 "geomField": "shape",
41                 "idField": "objectid",
42                 "idType": "number"
43             },
44             "layerOA": {
45                 "nom": "tabou_test_v_oa_operation",
46                 "geomField": "shape",

```

(suite sur la page suivante)

(suite de la page précédente)

```

47     "idField": "objectid",
48     "idType": "number"
49 },
50 "layerSA": {
51     "nom": "tabou_test_v_oa_secteur",
52     "geomField": "shape",
53     "idField": "objectid",
54     "idType": "number"
55 }
56 },
57 "searchCfg": {
58     "limit": 1500,
59     "communes": {
60         "layer": "ladm_terri:commune_emprise",
61         "geom": "shape",
62         "placeholder": "Communes",
63         "filterField": "code_insee",
64         "apiField": "codeInsee",
65         "apiLabel": "nom"
66     },
67     "quartiers": {
68         "layer": "ladm_terri:quartier",
69         "geom": "shape",
70         "placeholder": "Quartiers",
71         "filterField": "nuquart",
72         "apiField": "nuQuart",
73         "apiLabel": "nom"
74     },
75     "iris": {
76         "layer": "dem_stats:iris",
77         "geom": "shape",
78         "placeholder": "Iris",
79         "filterField": "code_iris",
80         "apiField": "codeIris",
81         "apiLabel": "nmiris"
82     },
83     "etapesoa": {
84         "layer": "tabou_test_v_oa_operation",
85         "geom": "shape",
86         "filterField": "etape",
87         "placeholder": "Etapas OA",
88         "apiField": "libelle",
89         "apiLabel": "libelle"
90     },
91     "etapespa": {
92         "layer": "tabou_test_v_oa_programme",
93         "geom": "shape",
94         "filterField": "etape",
95         "apiLabel": "libelle",
96         "apiField": "libelle",
97         "type": "string",
98         "placeholder": "Etapas PA"

```

(suite sur la page suivante)

```

99     },
100     "secteurs-sam": {
101         "layer": "urba_zona:v_chargedoperation_secteur",
102         "geom": "geom",
103         "filterField": "nom_secteur",
104         "apiLabel": "nomSecteur",
105         "apiField": "nomSecteur",
106         "placeholder": "Sec. SAM"
107     },
108     "secteurs-speu": {
109         "layer": "urba_zona:v_referent_urbaniste_secteur",
110         "geom": "geom",
111         "filterField": "nom_secteur",
112         "apiLabel": "nomSecteur",
113         "apiField": "nomSecteur",
114         "placeholder": "Sec. SPEU"
115     },
116     "secteurs-sds": {
117         "layer": "urba_zona:v_instructeur_secteur",
118         "geom": "geom",
119         "filterField": "secteur",
120         "apiField": "secteur",
121         "apiLabel": "secteur",
122         "placeholder": "Sec. SDS"
123     },
124     "secteurs-foncier": {
125         "layer": "urba_zona:v_negociateurfoncier_secteur",
126         "geom": "geom",
127         "filterField": "negociateur",
128         "apiField": "negociateur",
129         "apiLabel": "negociateur",
130         "placeholder": "Sec. Foncier"
131     },
132     "natures": {
133         "layer": "tabou_test_v_oa_operation",
134         "geom": "shape",
135         "service": "operations?nature",
136         "attribute": "layerOA",
137         "spatial": [
138             "layerPA",
139             "layerSA"
140         ],
141         "filterField": "nature",
142         "apiField": "libelle",
143         "apiLabel": "libelle",
144         "placeholder": "Natures"
145     },
146     "amenageurOA": {},
147     "promoteurOA": {},
148     "types-financements": {
149         "cqlCondition": {
150             "ACC_AIDEE": "logements_access_aide > 0",

```

(suite sur la page suivante)

(suite de la page précédente)

```

151     "ACC_LIBRE": "logements_access_libre > 0",
152     "ACC_MAITRISEE": "logements_access_maitrise > 0",
153     "LOC_AIDE": "logements_locatif_aide_prevu > 0",
154     "LOC_REGULE_HLM": "logements_locatif_regule_hlm_prevu > 0",
155     "LOC_REGULE_PRIVIE": "logements_locatif_regule_hlm_prevu > 0"
156 },
157     "layer": "tabou_test_v_oa_programme",
158     "geom": "shape",
159     "filterField": "code",
160     "apiField": "code",
161     "apiLabel": "libelle",
162     "placeholder": "Type de financement"
163 },
164     "plui": {
165         "layer": "urba_docs_plui:plui_plan_zonage_simplifie",
166         "geom": "shape",
167         "filterField": "etiquette",
168         "apiField": "libelle",
169         "apiLabel": "libelle",
170         "placeholder": "Zonage PLUI"
171     },
172     "daact": {
173         "layer": "tabou_test_v_oa_programme",
174         "geom": "shape",
175         "filterField": "daact_date",
176         "type": "date",
177         "placeholder": "Choisir une date"
178     },
179     "doc": {
180         "layer": "tabou_test_v_oa_programme",
181         "geom": "shape",
182         "filterField": "doc_date",
183         "type": "date",
184         "placeholder": "Choisir une date"
185     },
186     "livraison": {
187         "layer": "tabou_v_oa_programme",
188         "geom": "shape",
189         "filterField": "livraison_date",
190         "type": "date",
191         "placeholder": "Choisir une date"
192     },
193     "pbil": {
194         "layer": "app:tabou_v_pbil",
195         "geom": "shape"
196     }
197 },
198     "apiCfg": {
199         "apiURL": "/tabou2",
200         "documentsByPage": 5
201     }
202 },

```

(suite sur la page suivante)

(suite de la page précédente)

```
203     "override": {}  
204 }
```

Cette section vous permettra de modifier, générer et déployer la documentation que vous êtes en train de consulter.

4.1 Construire et déployer la documentation

- Les sources de la documentation sont localisées dans le répertoire **/docs** du dépôt
- Nous voulons que notre documentation soit construite (build) dans le dossier à créer :

```
/var/www/tabou2/doc
```

- Si vous avez utilisé XAMPP (voir « xampp » :), le dossier cible où sera construite la documentation sera dans ce dossier à créer (sous windows) :

```
C:\xampp\tabou2-doc\
```

- Nous avons ensuite à passer la commande (les chemins sont à adapter) :

```
sphinx-build -b html git/geor_tabou2_front/docs /var/www/tabou2-doc/
```

- La documentation est maintenant dans le dossier de notre choix :

```
/var/www/tabou2-doc/
```

ou pour XAMPP:

```
C:\xampp\tabou2-doc\
```

- Si vous mettez à jour les sources de la documentation ou bien si vous la modifiez, vous devrez relancer cette commande systématiquement
- Avec XAMPP Accédez à la documentation via localhost/tabou2-doc (tabou2-doc étant le nom de dossier que vous avez utilisé)

4.2 Installer XAMPP (windows)

- Téléchargez [XAMPP](#)
- Lancez XAMPP pour afficher l'interface d'administration (GUI)
- Sur la ligne du module « Apache », à droite cliquez sur « Start » au sein des actions
- « Apache » doit passer en vert dans la colonne « Module »
- Cliquez sur « Explorer » dans la colonne tout à droite
- Une fenêtre d'exploration s'affiche (par défaut vers C :xampp)
- Recherchez « htdocs » dans la fenêtre d'exploration
- Créez un dossier « tabou2-doc »

C'est dans le dossier « tabou2-doc » que sera déployée la documentation après la phase de build (voir plus haut).

- Accédez au dossier avec votre navigateur via l'URL :

`localhost/tabou2-doc`

4.3 Documentation

Pour obtenir plus d'information sur la syntaxe et sphinx :

1. [Sphinx](#)
2. [Sphinx syntaxe tutoriel](#)
3. [Sphinx syntaxe infos](#)
4. [Sphinx exemple syntaxe](#)