

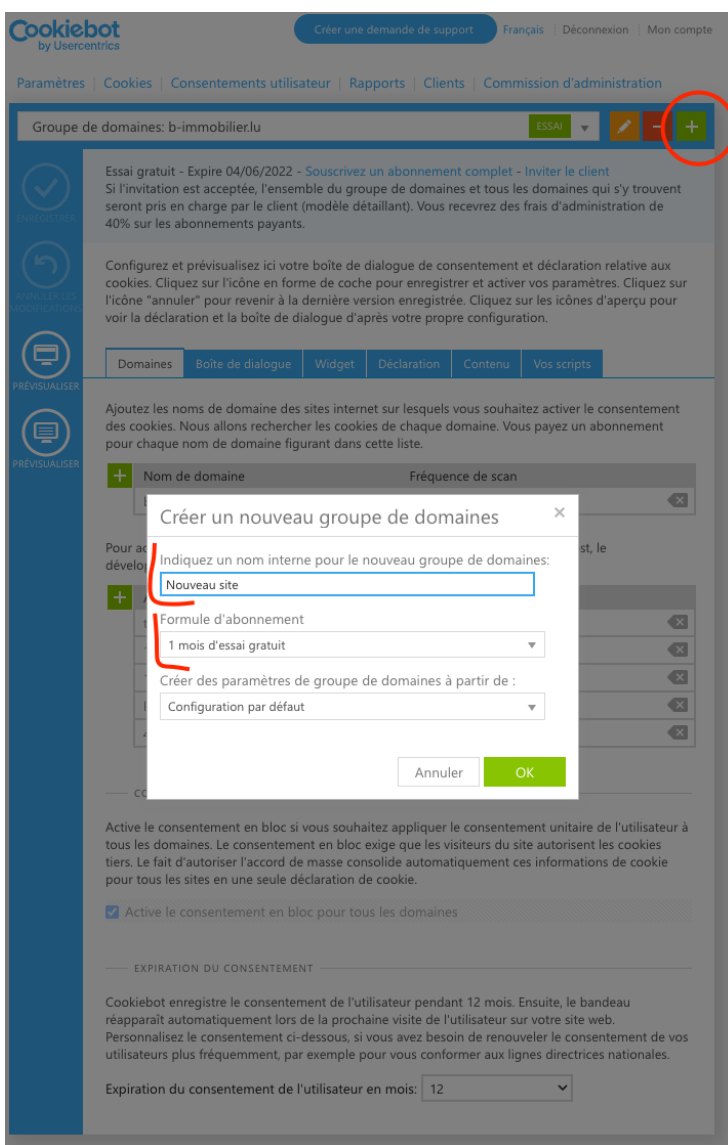
# Installation Cookie Bot

## Création compte cookiebot

1. Se connecter à l'administration de cookiebot en utilisant les identifiants enregistrés dans KeeWeb -

<https://manage.cookiebot.com/fr/login>

2. Créer un nouveau groupe de domaine



Mettre en essai gratuit dans un premier temps et à la première facturation, repasser la souscription en Premium et prévenir un SCRUM pour l'activation de la facturation. Rappel de la facturation (ZOHO).

Il existe 3 plans de facturations :

- Bannière Cookie - premium Small - tarif en 2022 → €388,00 (le tarif peut changer d'une année à l'autre)
- Bannière Cookie - premium Medium- tarif en 2022 → €772,00 (le tarif peut changer d'une année à l'autre)
- Bannière Cookie - premium Large- tarif en 2022 → €1.276,00 (le tarif peut changer d'une année à l'autre)

En cas de changement de plan une notification est envoyée à [mcc.econnect@gmail.com](mailto:mcc.econnect@gmail.com) et le data analyst. Merci de prévenir un SCRUM pour effectuer le changement de plan dans ZOHO.

### 3. Configurer le manager -

[https://support.cookiebot.com/hc/en-us/articles/4408356523282-Getting-started#h\\_01FJBY30BMEXSVQMD824XRCW41](https://support.cookiebot.com/hc/en-us/articles/4408356523282-Getting-started#h_01FJBY30BMEXSVQMD824XRCW41)

## Configuration CookieBot pour GTM

Se référer au site suivant : <https://insights.unnest.co/rgpd-et-web-analyse/installation-de-cookiebot-avec-le-consent-mode-via-google-tag-manager>

Pour plus de spécificité voir avec le data analyst.

## Code à utiliser

Code actuellement utilisé sur les sites e-connect, ce code est à modifier suivant les besoins:

- Partie gtag consent "granted" ou "denied"
- Id CookieBot (data-cbid)
- Dans le code JS remplacer si besoin les selectors
- Ajouter data-cookieconsent="ignore" sur les scripts considéré comme essentiels (par exemple le script principal JS app.js ou main.js)

```
<script id="Cookiebot" src="https://consent.cookiebot.com/uc.js" data-cbid="{{ id CookieBot }}" data-blockingmode="auto" type="text/javascript"></script>
```

```
<script data-cookieconsent="ignore">
```

```
  // INIT VAR
```

```
  let main = ''; // Define main content with a CSS selector. If empty, 'main' is used
```

```
  const lang = document.querySelector('html').attributes.lang.value; // get page
```

```
  language
```

```
  // INIT CONSENT
```

```
  window.dataLayer = window.dataLayer || [];
```

```
  function gtag() {
```

```
    dataLayer.push(arguments);
```

```
  }
```

```
  gtag("consent", "default", {
```

```
    ad_storage: "granted",
```

```
    analytics_storage: "granted",
```

```
    functionality_storage: "granted",
```

```
    personalization_storage: "granted",
```

```
    security_storage: "granted",
```

```
    wait_for_update: 1000,
```

```
  });
```

```
  gtag("set", "ads_data_redaction", true);
```

```
  //DISPLAY NOTICE MESSAGE FOLLOWING THE COOKIE BLOCKED
```

```
  window.addEventListener("load", function(){
```

```
    let optout, warning, pref, stat, mkt
```

```
    switch(lang){
```

```
      case "en":
```

```
        pref = 'Please <a href="javascript: Cookiebot.renew()">accept preferences-cookies</a> to load this content.';
```

```
        stat = 'Please <a href="javascript: Cookiebot.renew()">accept statistics-cookies</a> to load this content.';
```

```
        mkt = 'Please <a href="javascript: Cookiebot.renew()">accept marketing-cookies</a> to load this content.';
```

```
        break;
```

```

        case "fr":
            pref = 'Veuillez accepter les <a href="javascript: Cookiebot.renew()">cookies
de préférences</a> pour charger ce contenu.';
            stat = 'Veuillez accepter les <a href="javascript: Cookiebot.renew()">cookies
statistiques</a> pour charger ce contenu.';
            mkt = 'Veuillez accepter les <a href="javascript: Cookiebot.renew()">cookies
marketing</a> pour charger ce contenu.';
            break;
        case "de":
            pref = 'Bitte akzeptieren Sie <a
href="javascript: Cookiebot.renew()">Präferenz-Cookies</a>, um diesen Inhalt zu
laden.';
            stat = 'Bitte akzeptieren Sie <a
href="javascript: Cookiebot.renew()">Statistik-Cookies</a>, um diesen Inhalt zu
laden.';
            mkt = 'Bitte akzeptieren Sie <a
href="javascript: Cookiebot.renew()">Marketing-Cookies</a>, um diesen Inhalt zu
laden.';
            break;
        case "nl":
            pref = 'Accepteer <a href="javascript: Cookiebot.renew()">preferences-
cookies</a> om deze inhoud te laden.';
            stat = 'Accepteer <a href="javascript: Cookiebot.renew()">statistiek-
cookies</a> om deze inhoud te laden.';
            mkt = 'Accepteer <a href="javascript: Cookiebot.renew()">marketing-
cookies</a> om deze inhoud te laden.';
            break;
        default:
            pref = 'Please <a href="javascript: Cookiebot.renew()">accept preferences-
cookies</a> to load this content.';
            stat = 'Please <a href="javascript: Cookiebot.renew()">accept statistics-
cookies</a> to load this content.';
            mkt = 'Please <a href="javascript: Cookiebot.renew()">accept marketing-
cookies</a> to load this content.';
    }

    main = main == "" ? "main" : main; //if main is empty, css selector is 'main' else
css selector is a custom value

    const nodes = document.querySelectorAll(`.cookieconsent-optin-preferences,
.cookieconsent-optin-statistics, .cookieconsent-optin-marketing`)
    for(let i=0; i< nodes.length; i++){

```

```

        let node = nodes[i];
        console.log(node)
        // Add other conditions here. Complete nodes' selectors if needed
        /*if(node.tagName == "SCRIPT"){
            node.classList.add(`cookieconsent-optin-
${node.dataset.cookieconsent}`);
        }
        if(node.id == "entitylistmapgeo" || node.id ==
"entitylistdatamap"){
            node.classList.add('cookieconsent-optin-marketing');
        }*/
        let parent = node.parentNode;
        if(node.classList.contains("cookieconsent-optin-preferences")){
            optout = "cookieconsent-optout-preferences";
            warning = pref;
        }else if(node.classList.contains("cookieconsent-optin-
statistics")){
            optout = "cookieconsent-optout-statistics";
            warning = stat;
        }else if(node.classList.contains("cookieconsent-optin-
marketing")){
            optout = "cookieconsent-optout-marketing";
            warning = mkt;
        }
        console.log(i)
        let node_cc = document.createElement("div");
        node_cc.className = optout;
        node_cc.innerHTML = warning;
        parent.insertBefore(node_cc, node);
    }
});
// REMOVE NOTICE IF CLICKED
document.addEventListener("DOMContentLoaded", () => {
    window.addEventListener('CookiebotOnAccept', function (e) {
        if (Cookiebot.consent.preferences) {
            const nodesOptout = document.getElementsByClassName('cookieconsent-optout-
preferences')
            for(let j=0; j< nodesOptout.length; j++){
                nodesOptout[j].remove();
            }
        }
    });
});

```

```
    }
    if (Cookiebot.consent.statistics) {
        const nodesOptout = document.getElementsByClassName('cookieconsent-optout-
statistics')
        for(let j=0; j< nodesOptout.length; j++){
            nodesOptout[j].remove();
        }
    }
    if (Cookiebot.consent.marketing) {
        const nodesOptout = document.getElementsByClassName('cookieconsent-optout-
marketing')
        for(let j=0; j< nodesOptout.length; j++){
            nodesOptout[j].remove();
        }
    }
    },false);
});
</script>
```

---

Révision #3

Créé 11 May 2022 07:35:33

Mis à jour 27 May 2022 08:44:55 par Jean-Paul Diaz-Caneja