

CHAPITRE 24 — Les formulaires publics

InfraStudio embarque un moteur de formulaires publics conçu pour transformer la moindre saisie web en lead qualifié dans Dolibarr. Un formulaire de contact rempli sur un site géré par le module ne déclenche pas un simple envoi d'email : il alimente automatiquement le CRM, ouvre un ticket, prévient les équipes commerciales et déclenche, au besoin, un rappel en agenda. Ce chapitre détaille la mécanique interne, la configuration côté administrateur et l'intégration dans un site Dolibarr Website.

Architecture du moteur

L'architecture repose sur trois pièces : un descripteur JSON par formulaire, un point d'entrée unique côté serveur, et une chaîne d'adapters exécutés à chaque soumission. Aucun code PHP n'est nécessaire pour ajouter ou modifier un formulaire : il suffit d'éditer le descripteur depuis l'interface d'administration ou directement depuis l'éditeur Studio.

Pièce	Rôle
Endpoint unique	<code>public/forms/submit.php</code> reçoit toutes les soumissions. Le formulaire est identifié par son nom inséré en champ caché <code>form_name</code> .
Descripteur JSON	Une ligne dans <code>llx_infrasstudio_form_config</code> par formulaire. Décrit les champs, les règles de validation, les paramètres anti-spam et les adapters à exécuter.
Pipeline d'adapters	Six étapes activables à la carte : validation, anti-spam, tiers, contact, ticket, notification, agenda.

Le pipeline étape par étape

Chaque soumission acceptée traverse la chaîne ci-dessous. Chaque étape s'active indépendamment dans le descripteur, ce qui permet de couvrir aussi bien un formulaire de newsletter minimaliste qu'une demande de démonstration commerciale complète.

Adapter	Effet sur la soumission
---------	-------------------------

antispam	Honeypot, délai minimum de remplissage, rate-limit par adresse IP, captcha délégué au gestionnaire Dolibarr actif.
tiers	Recherche ou création d'une société dans <code>llx_societe</code> . Application d'une catégorie et d'une origine.
contact	Recherche ou création d'un contact dans <code>llx_socpeople</code> , libre ou rattaché au tiers.
ticket	Ouverture d'un ticket Dolibarr avec sujet, message, catégorie et sévérité.
notification	Envoi d'un email interne à l'équipe commerciale et d'un accusé de réception au visiteur.
agenda	Création d'un événement de rappel lié au tiers et au ticket.

Configurer un formulaire

Deux interfaces complémentaires sont disponibles. **L'administration avancée** se trouve dans *Outils* → *InfraS* → *Formulaires* : grille des configurations, statistiques de soumission, accès au descripteur JSON brut. **L'éditeur Studio** propose désormais un onglet *Formulaires* avec un inspector inline qui couvre la majorité des réglages courants sans repasser par l'administration Dolibarr.

Créer un nouveau formulaire

Le bouton « **+ Nouveau formulaire** » de l'onglet Formulaires de l'éditeur Studio ouvre une modale en quatre champs :

- le **libellé affiché en administration** (facultatif) ;
- l'**identifiant technique** (a-z, 0-9, tiret, underscore — utilisé dans le HTML, unique par entité) ;
- le **type** (*contact*, *newsletter*, *demo* ou *generic*) — le type définit les champs par défaut et le pipeline par défaut ;
- un **modèle de design** (facultatif) — voir la section *Les starter design templates* ci-dessous.

Le formulaire fraîchement créé est immédiatement sélectionné dans l'inspecteur et prêt à recevoir d'éventuels ajustements (champs, anti-spam, design, pipeline). L'identifiant technique sert d'ancrage tout au long du cycle de vie : il est référencé dans le HTML, dans le viewer de soumissions et dans les journaux.

Supprimer un formulaire

La section « **Zone dangereuse** » en bas de l'inspecteur expose un bouton *Supprimer ce formulaire*. La suppression efface uniquement la configuration : les soumissions déjà enregistrées dans `llx_infrasstudio_form_submission` sont conservées (traçabilité RGPD préservée). Une modale de confirmation Studio garde le doigt sur le bouton — la fenêtre `confirm()` native du navigateur n'est jamais utilisée. Le même contrôle est disponible depuis *Outils* → *InfraS* → *Formulaires* via le bouton *Supprimer* de chaque ligne.

Le descripteur JSON

Le cœur de la configuration est un descripteur JSON qui décrit le formulaire sous une forme structurée. L'éditeur d'administration valide la syntaxe à l'enregistrement et propose une référence dépliable de toutes les clés supportées pour éviter d'avoir à mémoriser la grammaire.

```
{
  "antispam": { "honeypot": true, "min_fill_seconds": 3, "rate_limit_per_hour": 5,
"captcha": true },
  "consent": { "required": true, "field_name": "consent", "text": "... " },
  "fields": {
    "name": { "required": true, "type": "text", "maxlength": 100 },
    "email": { "required": true, "type": "email", "maxlength": 200 },
    "message": { "required": true, "type": "text", "maxlength": 5000 }
  },
  "tiers": { "enabled": true, "lookup_by_email": true, "category_label": "Lead web"
},
  "ticket": { "enabled": true, "category_code": "COMMERCIAL" },
  "notification": { "autoreply_enabled": true, "autoreply_template_label": "Accusé de
réception" },
  "template_override": "site:contact.tpl.php"
}
```

La clé `template_override` est facultative — sans elle, le moteur utilise le template par défaut du type. Voir la section *Intégrer le formulaire dans un site* pour les trois modes de résolution.

Connecter le formulaire à votre CRM

L'intérêt principal du moteur réside dans la chaîne de traitement qu'il déclenche au moment de la soumission. Chaque adapter peut être activé indépendamment selon les besoins.

Tiers et contact

Lorsque l'adapter tiers est actif, le moteur recherche d'abord une société existante dont l'adresse email correspond à celle saisie ; à défaut, il regarde si l'email appartient à un contact rattaché à

une société, puis tente une correspondance par nom d'entreprise. Si aucune correspondance n'est trouvée, un nouveau tiers est créé, catégorisé automatiquement et associé à un canal d'origine (extrafield `origine` renseigné depuis le dictionnaire `c_input_reason`). La même logique existe pour les contacts, utile notamment pour les inscriptions à la newsletter qui ne nécessitent pas la création d'une société.

Ouverture d'un ticket

L'adapter ticket ouvre un ticket Dolibarr rattaché au tiers résolu. Le sujet et le message sont construits soit à partir des champs du formulaire, soit à partir de gabarits permettant d'injecter dynamiquement le nom du visiteur, sa demande, l'adresse IP d'origine ou tout autre élément du contexte. Catégorie et sévérité sont déterminées par les codes du dictionnaire Dolibarr (`llx_c_ticket_category`, `llx_c_ticket_severity`).

Notification et accusé de réception

Deux emails peuvent être envoyés automatiquement à la suite d'une soumission : une notification interne adressée à l'équipe commerciale, et un accusé de réception destiné au visiteur. L'accusé de réception peut s'appuyer sur un modèle d'email Dolibarr standard stocké dans `llx_c_email_templates`, ce qui permet de centraliser la rédaction et d'utiliser la même charte pour tous les formulaires d'un site.

Rappel automatique en agenda

Pour les formulaires à fort enjeu commercial (typiquement une demande de démonstration), un événement de rappel peut être créé dans l'agenda. Le délai est configurable, les week-ends peuvent être évités automatiquement, et l'événement est lié au ticket et au tiers pour garantir la traçabilité.

Intégrer le formulaire dans un site

Une fois la configuration en place, il reste à exposer le formulaire dans le site. Trois approches sont possibles selon le degré de personnalisation souhaité.

Le shortcode `{{form:name=...}}`

La voie la plus simple, surtout pour un rédacteur. Dans n'importe quelle page du site (slot richtext ou directement dans le tpl) :

```
{{form:name=contact-site}}
```

Le hook `completeHtmlOutput` du module remplace le token au rendu par le HTML complet du formulaire — anti-spam et style scopé inclus. Aucune ligne de PHP à toucher côté site.

L'helper de rendu unifié

Pour passer des options dynamiques (référence produit sur une landing, libellés sur mesure, etc.), appeler directement le helper depuis un tpl.php :

```
dol_include_once('/infrasstudio/core/lib/infrasstudio.lib.php');
infrasstudio_render_public_form('contact-site', array(
    'fk_website' => $website->id,
    'fk_page'    => $object->id,
    'extra'     => array('productRef' => 'monproduit'),
));
```

Les starter design templates

Quatre templates « clé en main » sont livrés dans `templates/forms/_starter-*.tpl.php`. Chacun est auto-suffisant (CSS inline scopé, rendu dynamique des champs déclarés dans `config.fields`) — idéal pour démarrer rapidement sur un nouveau site, avant d'éventuellement basculer sur du HTML maison.

Starter	Style
<code>_starter-minimal.tpl.php</code>	Sobre, labels au-dessus, focus indigo.
<code>_starter-card.tpl.php</code>	Carte avec ombre douce et bouton en gradient.
<code>_starter-inline.tpl.php</code>	Inputs alignés horizontalement, compact, idéal newsletter/footer.
<code>_starter-modern.tpl.php</code>	Coins arrondis, fond teinté, gradient bouton, labels uppercase.

Le starter choisi dans la modale « + Nouveau formulaire » est automatiquement posé comme `template_override` dans la configuration. Il peut être changé à tout moment via l'éditeur JSON avancé.

Conserver un design existant — préfixe `site:`

Pour intégrer le formulaire dans une charte graphique déjà existante (classes CSS du site, structure HTML spécifique), le mécanisme officiel consiste à livrer son propre template depuis le **dossier source du site Dolibarr Website** et à le référencer via le préfixe `site:` dans la configuration :

1. Créer un dossier `forms/` dans le source du site, soit `DOL_DATA_ROOT/<entity>/website/<ref>/forms/`.
2. Y déposer un fichier `contact.tpl.php` qui réutilise les classes CSS du site.
3. Dans le descripteur du formulaire, poser : `"template_override": "site:contact.tpl.php"`.

Au rendu, le moteur résout vers `DOL_DATA_ROOT/<entity>/website/<ref>/forms/contact.tpl.php` en utilisant le `fk_website` du contexte. Cette approche permet à chaque site de livrer ses propres

templates sans déposer le moindre fichier dans le module générique — qui reste 100% indépendant du métier de chaque client.

Alternative : si le design impose seulement quelques champs cachés à injecter dans un `<form>` déjà existant, le helper `llx_infrasstudio_render_form_security($formName, $fkWebsite, $fkPage, $_SERVER['PHP_SELF'])` émet d'un seul tenant le `form_name`, le timestamp anti-bot, le honeypot et le captcha conditionnel.

Suivre et auditer les soumissions

Chaque soumission acceptée est persistée dans `llx_infrasstudio_form_submission` avec son contenu sanitisé, l'adresse IP d'origine, l'agent utilisateur, la page de provenance et la trace du consentement RGPD. Le viewer d'administration (*Outils* → *InfraS* → *Soumissions*) permet de filtrer par formulaire, statut ou plage de dates, d'ouvrir le détail complet d'une soumission et d'accéder en un clic au tiers, au contact, au ticket et à l'événement d'agenda qui en ont découlé. Cette traçabilité complète est précieuse à la fois pour le suivi commercial et pour répondre aux demandes RGPD des visiteurs.

Récapitulatif

Le moteur de formulaires d'InfraSStudio transforme un simple formulaire web en véritable point d'entrée du CRM. Configurable sans code, sécurisé par défaut et entièrement intégré à l'écosystème Dolibarr, il évite la fragmentation des outils tout en gardant la souplesse nécessaire à chaque projet. Pour aller plus loin, voir le Chapitre 28 (constantes), le Chapitre 31 (modèle SQL des trois tables `llx_infrasstudio_form_*`) et l'Annexe B (FAQ) pour les questions opérationnelles courantes.

Revision #2

Created 2026-05-11 10:39:00 UTC by Lucky Ranasolonirina

Updated 2026-05-12 11:47:23 UTC by Lucky Ranasolonirina