

Integrações

Manuais de Integração de ferramentas de terceiros com o Priax.

- [GLPI Integration](#)

GLPI Integration

GLPI Integration

Instalação — Plugin GLPI “Priax CMDB”

1) Pré-requisitos

1.1) Requisitos do GLPI

- **GLPI 10.x** (recomendado manter no último patch da sua série 10.x)
- Acesso de **Super-Admin** no GLPI para instalar/habilitar plugins

1.2) Dependências opcionais

- **Formcreator** (somente se você for usar integração via formulários)
- O plugin Priax CMDB funciona sem o Formcreator, mas a seção de formulários exige o Formcreator instalado e habilitado.

1.3) Requisitos de infraestrutura / rede

O servidor GLPI precisa conseguir acessar o endpoint do Priax:

- **HTTPS** para a URL base configurada (porta 443 normalmente)
- Resolução DNS e rota de saída adequadas
- Se houver proxy corporativo, o GLPI/PHP deve estar configurado para sair via proxy (quando aplicável)

2) Instalação do plugin

Há dois caminhos principais:

- **A**) Instalação manual (ZIP / Git / pasta do plugin) — mais comum em ambientes offline
- **B**) Instalação via Marketplace — se o plugin estiver publicado/instalável por lá (caso aplicável no seu cenário)

A) Instalação manual (recomendada para ambientes controlados)

Passo A1 — Obter o pacote do plugin

Você terá um ZIP do plugin (ex.: `priaxcldb_glpi_plugin_vX_Y_Z.zip`).

Passo A2 — Copiar para a pasta de plugins do GLPI

No servidor onde o GLPI está instalado, descompacte o ZIP dentro de:

- **Linux (padrão)**:

```
/var/www/glpi/plugins/
```

O resultado deve ser:

- `/var/www/glpi/plugins/priaxcldb/``

“ Importante: o **nome da pasta** precisa ser exatamente ``priaxcldb`` (minúsculo).

“ Se você deixar ``priaxcldb_glpi_plugin_v0_4_28/`` ou similar, o GLPI não reconhece corretamente.

Passo A3 — Ajustar permissões

Garanta que o usuário do serviço web (ex.: `www-data`, `apache`, `nginx`) tenha permissão de leitura na pasta do plugin.

Exemplos:

- Debian/Ubuntu (apache/php-fpm com ``www-data``):

```
chown -R www-data:www-data /var/www/glpi/plugins/priaxcldb
find /var/www/glpi/plugins/priaxcldb -type d -exec chmod 755 {} \;
```

```
find /var/www/glpi/plugins/priaxcmdb -type f -exec chmod 644 {} \;
```

- RHEL/CentOS (apache com `apache`):

```
chown -R apache:apache /var/www/glpi/plugins/priaxcmdb
```

“ Se o seu GLPI roda em container (Docker/Kubernetes), faça isso no build da imagem ou via volume com permissões compatíveis.

B) Instalação via Marketplace (se aplicável)

1. Acesse: **Configurar** → **Plugins** → **Marketplace**
2. Busque por **Priax CMDB**
3. Clique em **Instalar**
4. Ao finalizar, siga para **habilitar** (próximo tópico)

“ Se sua instância não usa Marketplace (ou está bloqueado), use o método manual.

3) Habilitar e instalar no GLPI

Depois que os arquivos estiverem na pasta correta:

1. Acesse: **Configurar** → **Plugins**
2. Encontre **Priax CMDB**
3. Clique em **Instalar** (isso executa migrations/tabelas do plugin)
4. Após instalar, clique em **Ativar**

Se tudo ocorreu bem:

- o plugin passa a aparecer com status ****Ativo****
- surgem os menus do plugin na área de configuração

4) Verificações pós-instalação

4.1) Verificar logs do GLPI

Se houver erro de instalação/ativação, confira:

- **GLPI logs** (interface):
- **Configurar → Logs**
- **Arquivo de log PHP/Apache/Nginx**:
- `/var/log/apache2/error.log`` (Debian/Ubuntu)
- `/var/log/httpd/error_log`` (RHEL/CentOS)
- logs do PHP-FPM, se aplicável

Erros comuns:

- Pasta errada (`plugins/priaxcldb_glpi_plugin...`` em vez de `plugins/priaxcldb``)
- Permissão insuficiente
- Incompatibilidade de versão do GLPI/PHP

5) Configuração inicial (primeiro uso)

Após ativar:

1. Vá em: **Configurar → Plugins → Priax CMDB → Configuração**

2. Preencha:

- **Base URL**
- **Authentication mode**
- **Bearer token** ou **Basic user/password**
- **Timeout**
- **Verify TLS certificate** (conforme ambiente)

3. Salve

5.1) Teste de conectividade

O plugin normalmente valida chamadas reais quando você:

- abre telas de lookup (fields/profiles)

- executa buscas (PAQL search, etc.)

Recomendação de teste:

1. Crie um **Priax Field** simples com PAQL que retorne poucos itens.
 2. Abra um Ticket e verifique se o lookup carrega resultados.
-

6) Instalar/ativar Formcreator (opcional)

Se você for usar formulários:

1. Instale e ative o **Formcreator**
2. Em seguida, no Formcreator:
 - crie um formulário
 - adicione perguntas do tipo ****GLPI object****
 - selecione ****Priax Asset**** ou ****Priax Field Value**** (itemtypes fornecidos pelo plugin)

“ Se os itemtypes não aparecem na lista do Formcreator:

- confirme que ****Priax CMDB está ativo****
 - limpe cache do GLPI
 - verifique erros em logs
-

7) Atualização do plugin (upgrade)

7.1) Atualização manual

1. Faça backup:
 - banco do GLPI
 - pasta atual do plugin (`plugins/priaxcmdb`)
2. Substitua os arquivos pelo novo pacote:
 - sobrescreva `/plugins/priaxcmdb/`
3. No GLPI:

- **Configurar → Plugins**
- clique em **Atualizar** (se aparecer) ou **Instalar** (o GLPI aplica migrations pendentes)

4. Confirme que o plugin voltou a ficar **Ativo**

7.2) Cuidados

- Não renomeie a pasta do plugin.
 - Não apague as tabelas do plugin, senão você perde configurações e vínculos.
-

8) Checklist rápido de instalação (resumo)

- Pasta correta: `/glpi/plugins/priaxcldb/`
- Permissões OK (webserver lê a pasta)
- Plugin instalado e ativado em **Configurar → Plugins**
- Configuração inicial preenchida (URL + auth)
- Lookup/PAQL funcionando em um teste simples
- (Opcional) Formcreator instalado e itemtypes aparecem

Utilização do Plugin GLPI “Priax CMDDB” (v0.4.28)

9) Visão geral

O plugin “Priax CMDDB” integra o GLPI ao **Priax**, permitindo:

1. **Campos Priax (Priax Fields)**: campos dinâmicos (lookup) que consultam o Priax via **PAQL** e armazenam seleção/valor de forma estruturada.

2. **Perfis de Ativo Priax (Priax Asset Profiles)**: regras de busca/seleção de ativos no Priax e (opcionalmente) **criação/atualização** de itens locais no GLPI via mapeamento de atributos.

3. **Formcreator**: integração usando **questões do tipo “GLPI object”**. Suporta dois “objetos” Priax:

- **Priax Asset**
- **Priax Field Value**

“ Observação: a integração antiga com Formcreator (bridge/iframe) foi removida nas versões 0.4.x.

10) Configuração geral do plugin

Acesse: **Configurar** → **Plugins** → **Priax CMDB** → **Configuração**

10.1) Priax connector settings

Base URL

URL base do Priax (ex.: `https://demo.app.priax.io`).

Authentication mode

- **Bearer token**
- **Basic**

Basic user / Basic password

Usados no modo **Basic**. Se a senha ficar em branco ao salvar, o plugin mantém a senha atual.

Bearer token

Usado no modo **Bearer token**. Se ficar em branco ao salvar, o plugin mantém o token atual.

Timeout (seconds)

Tempo máximo para chamadas HTTP ao Priax.

Verify TLS certificate

Valida certificado TLS do servidor Priax.

****Cache TTL (seconds)****

TTL global de cache para listas/consultas quando aplicável (0 desabilita).

10.2) Integração com Ativos (Priax ? GLPI)

****Enable asset integration (Priax ? GLPI)****

Ativa “Asset Profiles”, endpoints e UI de vínculo/sincronismo.

****Show Priax Assets on Ticket tab****

Exibe a aba “Priax Assets” no Ticket.

****Show Priax Assets on Ticket right block****

Exibe um bloco/box de ativos Priax na lateral do Ticket.

11) Priax Fields

Acesse: **Configurar** → **Plugins** → **Priax CMDB** → **Manage Priax fields**

11.1) Opções do Priax Field

Identificação

- ****Name (slug)****: nome técnico/identificador.
- ****Label****: rótulo amigável.
- ****Help text****: texto de ajuda.
- ****Placeholder****: texto do input.

Ativação e comportamento

- ****Active****: liga/desliga.
- ****Mandatory****: obrigatório (onde suportado).
- ****Multi-value****: múltiplos valores (onde suportado).
- ****Show icon****: tenta exibir ícone quando disponível.
- ****Display weight****: ordem (peso).

Regras de exibição no Ticket

- **Ticket: Request types**: se vazio, qualquer tipo.
- **Ticket: Type (Criar) / (Editar)**: filtra Incidente/Requisição.
- **Ticket: Categories**: filtra por categoria.
- **Incluir subcategorias**: inclui categorias filhas.

Busca no Priax (PAQL)

- **Result limit**: limite de retorno.
- **Cache TTL override**: TTL específico do field (sobrescreve o global).
- **Save snapshot JSON**: salva JSON do(s) item(ns) selecionado(s).
- **PAQL initial**: PAQL principal do field.

Dependencies (cascata)

- **Parent field**: field pai.
 - **Value path (in parent payload)**: caminho no `payload_json` do pai.
 - **Placeholder**: placeholder usado na PAQL do field filho.
-

11.2) Portal (Helpdesk / Simplified)

- **Exibir este field no Portal**
 - Filtros de request type/tipo/categoria para portal
 - **Portal: PAQL initial** (se vazio, usa a PAQL inicial padrão)
-

12) Priax Asset Profiles

Acesse: **Configurar** → **Plugins** → **Priax CMDB** → **Manage asset profiles**

12.1) Opções do Asset Profile

Identificação

- **Name**: nome do profile.
- **Itemtype**: tipo GLPI alvo (Computer, Software, etc.).

- **Active**: liga/desliga.

Busca no Priax

- **PAQL search (use `{q}`)**: PAQL de busca usando `{q}`.
- **Portal: PAQL search**: variação para portal.

Portal

- **Exibir este profile no Portal**
- Filtros por request type/tipo (criar/editar)

Snapshot/cache

- **Save snapshot payload on select**
- **Cache TTL (seconds)**

Integração com item local

- **Update local asset on select**
- **Create local asset on select**

12.2) Attribute mappings (mapeamento de atributos)

Cada linha:

- **Active**
 - **Priax path** (caminho no JSON)
 - **GLPI field** (campo do item GLPI)
 - **Policy**:
 - `always_overwrite``
 - `overwrite_if_empty``
 - `never_overwrite``
 - **Resolver**:
 - `none``, `int``, `bool``, `from_ticket_entity``, `static_map``
 - **Transform**:
 - `trim``, `upper``, `lower``
-

13) Formcreator — usando os 2 tipos de campos Priax

A integração é feita com perguntas do tipo **GLPI object**, usando itemtypes do plugin:

1. **Priax Asset**
2. **Priax Field Value**

13.1) Criar pergunta “Priax Asset”

1. Formcreator → editar formulário
2. Adicionar pergunta **GLPI object**
3. Selecionar itemtype **Priax Asset**
4. Nas **Opções Priax**:
 - **Asset Profile** (obrigatório)
 - **Multi-select** (existente, mas comportamento atual tende a seleção única)
 - **Usar variáveis do usuário**
 - **Salvar opções Priax**

13.2) Criar pergunta “Priax Field Value”

1. Adicionar pergunta **GLPI object**
2. Selecionar itemtype **Priax Field Value**
3. Nas **Opções Priax**:
 - **Field** (obrigatório)
 - **PAQL override** (opcional)
 - **Multi-select** (reservado para evolução)
 - **Usar variáveis do usuário**
 - **Salvar opções Priax**

13.3) Após envio do formulário

O plugin persiste:

- ativos selecionados (Priax Asset)
- valores de fields selecionados (Priax Field Value)

vinculados ao alvo gerado (Ticket/Change/Problem).

14) PAQL

Este manual **não detalha a linguagem PAQL para uma compreensão da PAQL consulte o [manual específico](#).**

6.1) Variáveis de usuário para substituição na PAQL

O plugin suporta substituição de variáveis no template, por exemplo:

- `{{user.id}}`
- `{{user.login}}`
- `{{user.firstname}}`
- `{{user.realname}}`
- `{{user.name}}`
- `{{user.email}}`

No Formcreator, habilite **“Usar variáveis do usuário”** para aplicar a substituição antes de executar a PAQL.