

TItems — PlaceholderAPI & API Développeur / Developer API

PlaceholderAPI

FR — TItems expose des placeholders via PlaceholderAPI (identifiant : `tboss`). Ces placeholders fonctionnent dans un **rayon de 50 blocs** autour d'un boss actif.

EN — TItems exposes placeholders via PlaceholderAPI (identifier: `tboss`). These placeholders work within a **50-block radius** of an active boss.

⚠ Nécessite le plugin **PlaceholderAPI** installé. / Requires **PlaceholderAPI** installed.

Placeholders disponibles / Available Placeholders

Placeholder	Description FR	Description EN	Valeur si aucun boss
<code>%tboss_near_boss%</code>	Boss dans les 50 blocs ?	Boss within 50 blocks?	ERROR
<code>%tboss_health%</code>	HP restants du boss proche	Nearby boss remaining HP	ERROR
<code>%tboss_type%</code>	Type du boss (<code>FINAL_DAMAGE</code> ...)	Boss type	ERROR
<code>%tboss_lead_name_1%</code>	Nom du joueur #1 au classement dégâts	Player name at damage rank #1	Nobody
<code>%tboss_lead_name_2%</code>	Nom du joueur #2 au classement	Player name at damage rank #2	Nobody
<code>%tboss_lead_name_N%</code>	Nom du joueur #N	Player name at rank N	Nobody
<code>%tboss_lead_health_1%</code>	Dégâts infligés par le joueur #1	Damage dealt by player #1	0

Placeholder	Description FR	Description EN	Valeur si aucun boss
<code>%tboss_lead_health_N%</code>	Dégâts infligés par le joueur #N	Damage dealt by player #N	<code>0</code>

Usage exemple (scoreboard, HUD...)

```

Boss proche: %tboss_near_boss%
Vie restante: %tboss_health%
#1 - %tboss_lead_name_1% (%tboss_lead_health_1% dmg)
#2 - %tboss_lead_name_2% (%tboss_lead_health_2% dmg)

```

API Développeur / Developer API

Le package d'API est `fr.totallycorp.titems.api.event`. Il expose des **Bukkit events personnalisés** que les plugins tiers peuvent écouter.

The API package is `fr.totallycorp.titems.api.event`. It exposes **custom Bukkit events** that third-party plugins can listen to.

HarvestEvent

Déclenché lors de la récolte d'une culture via les items TItems (Harvester).

Fired when a crop is harvested via TItems items (Harvester).

Méthodes / Methods :

Méthode / Method	Retour / Return	Description
<code>getPlayer()</code>	<code>Player</code>	Joueur qui récolte / Harvesting player
<code>getMaterial()</code>	<code>Material</code>	Type de culture récoltée / Harvested crop type

Exemple / Example :

```

@EventHandler
public void onHarvest(HarvestEvent event) {
    Player player = event.getPlayer();
    Material crop = event.getMaterial();
    // votre logique / your logic
}

```

InstantKillEvent (*Cancelable*)

Déclenché quand un monstre est tué en un coup par la Farming Sword. **Annulable**.

Fired when a mob is one-shot by the Farming Sword. **Cancelable**.

Méthodes / Methods :

Méthode / Method	Retour / Return	Description
<code>getKiller()</code>	<code>Player</code>	Joueur attaquant / Attacking player
<code>getEntity()</code>	<code>LivingEntity</code>	Entité tuée / Killed entity
<code>setCancelled(true)</code>	<code>void</code>	Annule le one-shot / Cancels the one-shot

Exemple / Example :

```
@EventHandler
public void onInstantKill(InstantKillEvent event) {
    if (someCondition) {
        event.setCancelled(true);
    }
}
```

PlayerDropOnDeathEvent

Déclenché à la mort d'un joueur pour gérer les drops d'items TItems. Permet de **forcer la conservation** d'un item spécifique.

Fired on player death to manage TItems item drops. Allows **forcing item retention**.

Méthodes / Methods :

Méthode / Method	Retour / Return	Description
<code>getPlayer()</code>	<code>Player</code>	Joueur décédé / Deceased player
<code>getDrops()</code>	<code>List<ItemStack></code>	Items normaux dropés / Normal item drops
<code>getArmor()</code>	<code>List<ItemStack></code>	Armures dropées / Armor drops
<code>getAllDrops()</code>	<code>List<ItemStack></code>	Tous les drops combinés / All drops combined
<code>keepItem(ItemStack)</code>	<code>void</code>	Force la conservation d'un item / Forces item retention

Méthode / Method	Retour / Return	Description
<code>getKeep()</code>	<code>List<ItemStack></code>	Liste des items conservés / Kept items list

Exemple / Example :

```
@EventHandler
public void onDeath(PlayerDropOnDeathEvent event) {
    // Conserver tous les Nether Stars / Keep all Nether Stars
    event.getDrops().stream()
        .filter(item -> item.getType() == Material.NETHER_STAR)
        .forEach(event::keepItem);
}
```

EventStartEvent / EventStopEvent

Événements génériques déclenchés au **démarrage** et à l'**arrêt** d'un event TItems (boss, etc.).
Generic events fired on TItems event **start** and **stop**.

```
@EventHandler
public void onEventStart(EventStartEvent event) { /* ... */ }

@EventHandler
public void onEventStop(EventStopEvent event) { /* ... */ }
```

Accès aux Managers / Manager Access

Tous les managers sont accessibles via l'instance principale du plugin.
All managers are accessible via the plugin's main instance.

Manager	Accès / Access	Rôle FR	Role EN
<code>BossManager</code>	<code>TItems.getInstance().getBossManager()</code>	Gère les boss actifs	Manages active bosses
<code>ScepterManager</code>	<code>TItems.getInstance().getScepterManager()</code>	Gère les PvP sceptre	Manages scepter PvP
<code>MaskManager</code>	<code>TItems.getInstance().getMaskManager()</code>	Gère les masques équipés	Manages equipped masks
<code>SellItemsManager</code>	<code>TItems.getInstance().getSellItemsManager()</code>	Gère la vente d'items	Manages item selling

Exemple — Vérifier si un boss est à proximité

```
BossManager bm = TItems.getInstance().getBossManager();

boolean isBossNearby = bm.getBossMap().values().stream()
    .anyMatch(boss -> boss.getLoc().distance(player.getLocation()) <= 50);
```

Exemple — Récupérer le classement dégâts d'un boss

```
for (Boss boss : bm.getBossMap().values()) {
    HashMap<UUID, Double> damages = boss.getPlayerDamage();
    // trier et afficher / sort and display
}
```

Revision #1

Created 2026-03-14 06:36:53 UTC by Neyzo

Updated 2026-03-14 06:37:07 UTC by Neyzo