

Documentation de l'API de paiement et d'impression de myTerminal



Figure 1: Ciss-logo

Versionning

Version	Auteur	Valideur	Commentaires
0.1	B.Alvarez	L.Frachon	
0.2	B.Alvarez	R.Riquoir	App2App requests

Généralités

En complément des services qu'elle propose, l'application myTerminal permet d'intégrer les différentes fonctionnalités disponibles sur un terminal de paiement bancaire : paiement carte bancaire agréé ou de type SoftPos, impression de ticket, e-ticketing, affichage écran,...

Ces services peuvent être intégrés à une application existante.

Licence

L'application est fournie sous forme de licence d'utilisation aux partenaires. Même intégrée dans un applicatif, elle reste sous la propriété intellectuelle exclusive de Ciss ou de ses ayants-droits.

Toute forme de retro-ingéneering est prohibée.

Pré-requis

A ce jour, l'application et le service sont compatibles avec les terminaux Android 5.1 ou supérieur.

Le programme myTerminal(API) doit être déclaré auprès de Ciss, et les différents appels & initialisations effectués dans les environnements d'intégration, préalablement à tout déploiement en production.

Limitations de responsabilité

L'application hôte intégrant le service devra prendre toutes les précautions de l'état de l'art pour protéger les informations vitales transmises.

Ciss ne pourra être tenu pour responsable de la divulgation des informations critiques en dehors de son propre écosystème applicatif.

Fonctionnement général

L'api de paiement et d'impression de myTerminal fonctionne via le mécanisme d'Intent ou de Broadcast d'Android.

(cf <https://developer.android.com/reference/android/content/Intent>)

(cf <https://developer.android.com/training/basics/intents/result>)

Une action accompagne l'intent, ainsi que des paramètres (extras). Un exemple complet d'utilisation est disponible à la fin de ce document.

Paielement (intent)

Action `com.ciss.myterminal.api.PAYMENT`

Paramètres

- **amount** *String obligatoire* en centimes (ex :100 = 1,00€)
- **action** *String optionnel (DEBIT par défaut)* DEBIT / CREDIT / CANCEL
- **reference** *String optionnel*

Impression (Broadcast)

Action `com.ciss.myterminal.api.PRINT`

Paramètres Au moins un paramètre est obligatoirement à envoyer.

- **image** *String base64*
- **text** *String*
- **lines** *String Array*

Impression / eTicket (App2App)

Action `com.ciss.myterminal.PRINT`

Paramètres Au moins un paramètre est obligatoirement à envoyer.

- **text** *String*
- **lines** *String Array*

Cashless (App2App)

Action com.ciss.myterminal.CASHLESS

Paramètres

- **amount** *String obligatoire* en centimes (ex :100 = 1,00€)
- **action** *String optionnel (DEBIT par défaut)* DEBIT / CREDIT (RUF 2023) / CANCEL (RUF 2023)

Initialisation myTerminal (App2App)

Action com.ciss.myterminal.INIT

Paramètres

- **posId** *String obligatoire*
- **licence** *String obligatoire*
- **rank** *String obligatoire*
- **replace** *Boolean optionnel* true / false (RUF 2023)

Retour L'initialisation de myTerminal retourne les éléments nécessaires pour faire des appels à l'API via le même équipement

- **merchantId** *Integer* Exemple: 12345
- **apiKey** *String* Exemple: "azerty123456"
- **rank** *String* Exemple: "001"
- **session** *String* Exemple: "azerty123456"
- **serial** *String* Exemple: "12345@azerty123456"
- **params** *Object* (paramétrage de caisse) Exemple: { "printer": "myTerminal eTicket" }

Exemple complet

```
TextView textResult;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    textResult = findViewById(R.id.text_result);

    // Demande de paiement CB
    (findViewById(R.id.button_payment)).setOnClickListener(v -> {
        Log.d("TEST_", "Start payment intent");
        textResult.setText("En attente...\n");
        Intent intent = new Intent("com.ciss.myterminal.api.PAYMENT");
```

```

        intent.putExtra("action", "DEBIT"); // DEBIT / CREDIT / CANCEL
        intent.putExtra("amount", "100"); // 100 = 1,00€
        intent.putExtra("reference", "abcd1234"); // Optionnel

        mStartForResult.launch(intent);
    });

    // Impression via broadcast
    (findViewById(R.id.button_print_image)).setOnClickListener(v -> {
        textResult.setText("En attente...\n");
        Intent intent = new Intent("com.ciss.myterminal.api.PRINT");

        intent.putExtra("image", "iVBORwOKGgoAAAANSUhEUgAAAcwAAAHMA...");

        sendBroadcast(intent);
    });

    // Impression via broadcast
    (findViewById(R.id.button_print_text)).setOnClickListener(v -> {
        textResult.setText("En attente...\n");
        Intent intent = new Intent("com.ciss.myterminal.api.PRINT");

        intent.putExtra("text", "Texte à imprimer\n    Texte avec retrait");

        sendBroadcast(intent);
    });

    // Impression via broadcast
    (findViewById(R.id.button_print_lines)).setOnClickListener(v -> {
        textResult.setText("En attente...\n");
        Intent intent = new Intent("com.ciss.myterminal.api.PRINT");

        String[] lines = {"Texte à imprimer", "    Texte avec retrait"};
        intent.putExtra("lines", lines);

        sendBroadcast(intent);
    });

    // Impression via broadcast
    (findViewById(R.id.button_print_all)).setOnClickListener(v -> {
        textResult.setText("En attente...\n");
        Intent intent = new Intent("com.ciss.myterminal.api.PRINT");

        intent.putExtra("image", "iVBORwOKGgoAAAANSUhEUgAAAcwAAAHMAQMAAABi...");

```

```

        intent.putExtra("text", "Texte à imprimer\n      Texte avec retrait");

        String[] lines = {"Texte à imprimer", "      Texte avec retrait"};
        intent.putExtra("lines", lines);

        sendBroadcast(intent);
    });

    // Impression & eTicket en App2App
    (findViewById(R.id.button_print_text)).setOnClickListener(v -> {
        textResult.setText("En attente...\n");
        Intent intent = new Intent("com.ciss.myterminal.PRINT");

        intent.putExtra("text", "Texte à imprimer\n      Texte avec retrait");

        mStartForResult.launch(intent);
    });

    // Impression & eTicket en App2App
    (findViewById(R.id.button_print_lines)).setOnClickListener(v -> {
        textResult.setText("En attente...\n");
        Intent intent = new Intent("com.ciss.myterminal.PRINT");

        String[] lines = {"Texte à imprimer", "      Texte avec retrait"};
        intent.putExtra("lines", lines);

        mStartForResult.launch(intent);
    });

    // Demande paiement cashless en App2App
    (findViewById(R.id.button_cashless)).setOnClickListener(v -> {
        Intent intent = new Intent("com.ciss.myterminal.CASHLESS");

        intent.putExtra("amount", "100"); // 100 = 1,00€
        intent.putExtra("action", "DEBIT"); // DEBIT / CREDIT (non disponible) / CANCEL (non

        mStartForResult.launch(intent);
    });

    // Initialisation de myTerminal en App2App
    (findViewById(R.id.button_init)).setOnClickListener(v -> {
        Intent intent = new Intent("com.ciss.myterminal.INIT");

        intent.putExtra("posId", "25283");
        intent.putExtra("licence", "252833");
        intent.putExtra("rank", "475");

```

```

        mStartForResult.launch(intent);
    });
}

ActivityResultLauncher<Intent> mStartForResult = registerForActivityResult(new ActivityY
    result -> {
        Log.d("TEST_", "result");
        if (result.getResultCode() == Activity.RESULT_OK) {
            Log.d("TEST_", "result OK");
            textResult.setText("RESULT OK\n");
        } else {
            Log.d("TEST_", "result KO");
            textResult.setText("RESULT KO\n");
            Intent intent = result.getData();
            if (intent != null) {
                Uri uri = intent.getData();
                if (uri != null) {
                    textResult.append(uri.toString());
                }
            }
        }
    });
}
});

```