

TsPay-Gateway

Integrazione con il servizio Gateway TsPay

- [Introduzione](#)
 - [Overview](#)
 - [Servizi integrati](#)
- [Getting started](#)
 - [Overview](#)
 - [Azienda: Attivazione e Onboarding](#)
 - [Azienda: Finalizzazione Onboarding](#)
- [API](#)
 - [Introduzione](#)
 - [Incasso](#)
 - [Conti](#)
 - [Pagamento](#)
 - [Consenso SCA](#)
 - [Banche e Prodotti](#)
 - [Settings](#)
- [Casi d'uso](#)
 - [Collegamento conto](#)
- [Notifiche](#)
 - [Overview](#)
 - [Flusso notifiche](#)
 - [Payload notifiche tspay_charge](#)
 - [Payload notifiche tspay_source](#)
 - [Payload notifiche tspay_pis](#)

- [Payload notifiche tspay_consent](#)
- [Payload notifiche tspay_wire](#)
- [Configurazione corrente NCS](#)

Introduzione

Overview

Il servizio pay-gateway permette l'integrazione del servizio TsPay all'interno del mondo Digital senza necessità di integrarsi direttamente con esso.

Essendo parte del mondo Digital, l'autenticazione verso il pay-gateway è analoga a quella usata negli altri servizi. È poi il gateway ad occuparsi dell'autenticazione dell'azienda verso TsPay, che può avvenire solo se il processo di onboarding è stato completato e confermato.

I servizi ad oggi offerti dal pay-gateway sono parte di tre servizi che la Banca d'Italia definisce come **mandatorio all'incasso** (num. 3), **disposizione di ordini di pagamento** (servizi PIS, num. 7), e **informazione e aggregazione conti** (servizi AIS, num. 8).

Oltre ad una serie di [endpoint](#) per operare sui servizi sopra citati, il servizio pay-gateway si occupa anche della gestione e reinoltro delle [notifiche](#) per tutti i casi previsti dal servizio TsPay.

Servizi integrati

I servizi attualmente integrati nel pay-gateway sono:

Mandatorio all'incasso (3):

- [Fattura attiva](#)

Disposizione di ordini di pagamento (servizi PIS, 7):

- [Fattura passiva](#)

Informazione e aggregazione conti (servizi AIS, 8):

- [Lista movimenti](#)
- [Lista conti collegati](#)

Getting started

Getting started

Overview

Per poter utilizzare il servizio TsPay si deve:

1. Lato azienda: attivare il servizio all'interno di Digital ed effettuare poi l'onboarding.
2. Lato applicativo integrante: configurare l'applicativo su NCS per ricevere le notifiche sulle operazioni effettuate.

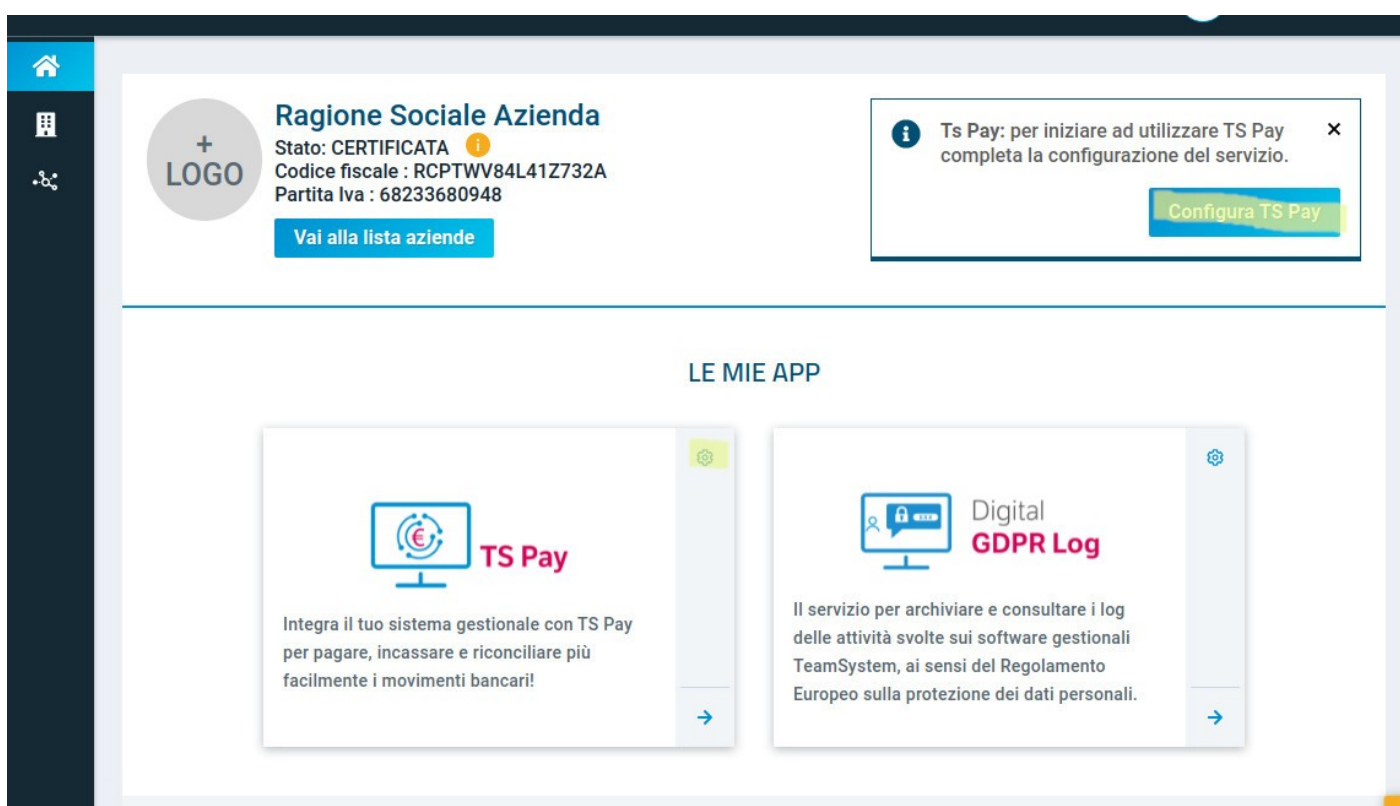
Studio e Gestita

Nel caso di uno studio che vuole utilizzare i servizi di TsPay su una sua gestita, non è necessario che lo studia faccia l'onboarding, ma è sufficiente che estenda il servizio sulla gestita, e poi sia questa a fare l'onboarding su TsPay (riconducendosi quindi al caso precedente).

Azienda: Attivazione e Onboarding

1. Attivazione

Attivare il servizio all'interno di Digital, quindi entrare nella pagina di configurazione tramite la cta "Configura TS Pay" o tramite l'icona in alto a destra nella card:



Una volta dentro, seguire la cta "Registra azienda" che reindirizzerà l'utente su TsPay per permettere l'onboarding.

2. Onboarding

Atterrati su TsPay, fare login con l'utenza desiderata (o crearne una nuova) e procedere con l'onboarding.

Per dettagli sull'onboarding **in ambiente di test** seguire i passaggi **dal punto 3 al punto 5 o 7** descritti [a questo link](#).

3. Finalizzazione (solo test e dev)

Una volta completato l'onboarding, in ambiente di test e dev il processo va finalizzato tramite una chiamata API: [qui dettagli](#). Se la chiamata va a buon fine si riceveranno due email, alla casella di posta dell'utente con cui ci si è loggati in TsPay:

1. Conferma di onboarding azienda, con codice titolare (merchant ref).
2. Conferma di creazione ApiKey, con link per l'attivazione della stessa.

4. Attivazione ApiKey

Seguire il link ricevuto nella seconda email, per l'attivazione dell'ApiKey.

Azienda: Finalizzazione Onboarding

SOLO TEST E DEV

Per finalizzare l'onboarding (step 3 di [questo processo](#)) va effettuata una chiamata ad uno dei due seguenti endpoint (sono alternativi):

1. Utilizzando la session-key dell'onboarding appena concluso:

```
[PUT] /internal/helpers/finalizeonboarding/by-sessionkey/{onboardingSessionKey}
```

2. Utilizzando l'itemId dell'azienda onboardata:

```
[PUT] /internal/helpers/finalizeonboarding/by-itemid/{itemId}
```

I due endpoint **NON** sono esposti dal pay-gateway ma dal servizio **ts-pay-config-write** e sono **disponibili solo in ambiente di test e dev**. Swagger:

- [DEV](#)
- [TEST](#)

Esito: se la chiamata va a buon fine si riceveranno due email, alla casella di posta dell'utente con cui ci si è loggati in TsPay:

1. Conferma di onboarding azienda completato, con codice titolare (merchant ref).
2. Conferma di creazione ApiKey, con link per l'attivazione della stessa.

API

Introduzione

Swagger

- [DEV](#)
- [TEST](#)

Headers

Per chiamare correttamente gli endpoint messi a disposizione dal servizio TS Pay Gateway, è necessario includere gli headers come descritto [qui](#).

Struttura dell'endpoint

Gli endpoint sono costruiti secondo la struttura:

```
/api/v{X}/{itemId}/...
```

dove

- `vX` dove X è la versione dell'API (al momento solo **v1**)
- `itemId` è il company registry dell'azienda per la quale si richiede il servizio

In questa guida tutti gli endpoint sono descritti senza il prefisso `/api/vX` che va dunque anteposto

Incasso

Fattura attiva

Descrizione

Consente di ricevere un pagamento attraverso il servizio di TS Pay; chiamando l'endpoint **link2pay** si riceverà un'url col quale sarà possibile fare un pagamento.

Richiesta link per effettuare un pagamento

Genera un link per effettuare un pagamento attraverso il servizio di TS Pay.

```
[POST] /{itemId}/link2pay
```

Nella **request** in particolare abbiamo:

- `template` contenente il testo ed eventualmente il logo da visualizzare nella pagina del pagamento generata e raggiungibile al link restituito in risposta
 - `title` titolo, generalmente usato per indicare la ragione sociale del merchant (es: "TeamSystem S.p.A.")
 - `desc` motivazione del pagamento (es: "Fattura di vendita")
 - `paymentRef` estremi della fattura (es: "Numero 123 del 15/06/2020")
 - `logo` link all'immagine da visualizzare come logo
- `callbackUrl` url cui reindirizzare l'utente una volta completata l'operazione. facoltativo
- `sourceTypes` sorgenti di pagamento accettate: carta di credito, SDD o entrambi (default)
- `maxPaymentsNumber` numero massimo di pagamenti consentiti (0=illimitato)
- `amount` l'importo da pagare espresso come numero intero in cui le ultime due cifre compongono la parte decimale (ad es. 1450 è da considerare come 14,50)
- `externalRef` un riferimento univoco al pagatore, da utilizzare in seguito come elemento di ricerca
- `metadata` elementi di tracciabilità, da utilizzare per la riconciliazione

Nella **response** l'API fornisce:

- `orderKey` identificativo dell'ordine
- `url` link da utilizzare per eseguire l'operazione

Richiesta link per memorizzare una sorgente di pagamento per addebiti futuri

Genera un link per salvare una sorgente di pagamento attraverso il servizio di TS Pay.

```
[POST] /{itemId}/link2save
```

Nella **request** in particolare abbiamo:

- `template` contenente il testo ed eventualmente il logo da visualizzare nella pagina del pagamento generata e raggiungibile al link restituito in risposta
 - `title` titolo, generalmente usato per indicare la ragione sociale del merchant (es: "TeamSystem S.p.A.")
 - `desc` motivazione del pagamento (es: "Fattura di vendita")
 - `paymentRef` estremi della fattura (es: "Numero 123 del 15/06/2020")
 - `logo` link all'immagine da visualizzare come logo
- `callbackUrl` url cui reindirizzare l'utente una volta completata l'operazione. facoltativo
- `sourceTypes` sorgenti di pagamento accettate: carta di credito, SDD o entrambi (default)
- `maxSourceNumber` numero massimo di sorgenti memorizzabili (0=illimitato, default=1)
- `maxAmount` l'importo massimo per gli addebiti, inteso come cumulativo, espresso come numero intero in cui le ultime due cifre compongono la parte decimale (ad es. 1450 è da considerare come 14,50)
- `contextId` il contesto a cui la memorizzazione della sorgente si applica, quindi nel caso specifico la fattura
- `externalRef` un riferimento univoco al pagatore, da utilizzare in seguito come elemento di ricerca
- `metadata` elementi di tracciabilità, da utilizzare per la riconciliazione

Nella **response** l'API fornisce:

- `orderKey` identificativo dell'ordine
- `url` link da utilizzare per eseguire l'operazione

Addebito

Esegue l'addebito sulla sorgente salvata con il LinkToSave.

```
[POST] /{itemId}/charges
```

Tra gli **header** è previsto:

- `x-operation-key` chiave univoca dell'addebito con lo scopo di prevenire i doppi incassi, in modo che se si tenta una seconda operazione con la stessa chiave viene restituito un errore (opzionale)

Nella **request** in particolare abbiamo:

- `sourceKey` identifica la sorgente di pagamento memorizzata dal customer attraverso il `LinkToSave`
- `contextId` l'eventuale contesto precedentemente impostato
- `description` è la motivazione del pagamento, che finisce nel movimento
- `amount` l'importo da addebitare espresso come numero intero in cui le ultime due cifre compongono la parte decimale (ad es. 1450 è da considerare come 14,50)

Nella **response** l'API fornisce:

- `orderKey` identificativo dell'ordine
 - `chargeKey` identificativo dell'addebito
-

Esito pagamento

Da utilizzare per conoscere tutti i dettagli dell'addebito.

```
[GET] /{itemId}/charges/orders/{orderKey}
```

I **parametri** richiesti sono:

- `orderKey` path variable - key dell'ordine, ritornato dalla [chiamata alla creazione della richiesta di pagamento](#).

Conti

Aggregazione conti e carte

Descrizione

Consente di recuperare le informazioni aggregate di saldo e movimenti relativi a conti corrente e carte di credito tramite il servizio di TS Pay. La SCA è unica ma va rinnovata periodicamente, in genere ogni 90 giorni.

Informazioni conti collegati

Restituisce l'elenco dei prodotti (conti e carte) collegati, comprensivo di saldo disponibile e informazioni relative al consenso.

```
[GET] /{itemId}/linked-accounts
```

Nella **response** l'API fornisce:

- `accountId` id del conto/carta
- `iban` iban del conto (se il prodotto è un conto corrente)
- `maskedPan` numero della carta mascherato (se il prodotto è una carta)
- `currency` valuta di conto
- `providerId` id del provider (ASPSP)
- `bankName` nome della banca
- `productCode` codice del prodotto
- `lastBalance` saldo
 - `amount` importo (espresso come numero intero in cui le ultime due cifre compongono la parte decimale)
 - `currency` valuta
- `lastBalanceDate` data di riferimento del saldo
- `accountNature` natura del conto
- `consentId` id del consenso
- `consentExpireDate` data di scadenza del consenso
- `lastRefreshStatus` stato ultimo aggiornamento
- `lastRefreshDate` data ultimo aggiornamento

Informazioni conti collegati o precedentemente utilizzati

Restituisce l'elenco dei prodotti (solo conti) collegati o precedentemente utilizzati. I primi comprensivi di saldo disponibile e informazioni relative al consenso.

```
[GET] /{itemId}/accounts
```

Nella **response** l'API fornisce:

- `accountId` id del conto/carta (solo per conti collegati)
- `iban` iban del conto
- `currency` valuta di conto
- `providerId` id del provider (ASPSP)
- `bankName` nome della banca
- `productCode` codice del prodotto
- `lastBalance` saldo (solo per conti collegati)
 - `amount` importo (espresso come numero intero in cui le ultime due cifre compongono la parte decimale)
 - `currency` valuta
- `lastBalanceDate` data di riferimento del saldo (solo per conti collegati)

Informazioni movimenti

Restituisce le informazioni relative alle transazioni dei conti collegati.

```
[GET] /{itemId}/transactions
```

I **parametri** richiesti sono:

- `dateFrom` data inizio movimenti (opzionale)
- `dateTo` data fine movimenti (opzionale)
- `bookingStatus` stato del movimento (booked o pending) (opzionale)
- `accounts` accounts per i quali mostrare i movimenti (opzionale)

Nella **response** l'API fornisce:

- `accountId` identifica il conto a cui la transazione fa riferimento
- `bookingStatus` stato del movimento (booked o pending)
- `bookingDate` data del movimento
- `remittanceInformationUnstructured` descrizione del movimento
- `transactionAmount` importo movimento
 - `amount` importo (espresso come numero intero in cui le ultime due cifre compongono la parte decimale)

- `currency` valuta
 - `valueDate` data valuta
-

Aggiornamento dati di un conto

Una volta collegato un conto/carta è necessario fare un aggiornamento dei dati, per scaricare transazioni e saldi. Questa operazione è schedulata tutte le notti, ma si può anche eseguire a richiesta tramite la seguente API. Va fatta per tutti gli account collegati con quel determinato consenso.

```
[POST] /{itemId}/accounts/{accountId}/refresh
```

I **parametri** richiesti sono:

- `accountId` path variable - identifica il conto

Nel corpo della **request** in particolare abbiamo:

- `psuIpAddress` IP del client che effettua la richiesta (è richiesto un indirizzo IP valido)

Nella **response** l'API fornisce:

- `accountId` identifica il conto
- `status` stato dell'operazione di refresh (tipicamente *"updating"*)

Pagamento

Fattura passiva

Descrizione

Consente di effettuare un bonifico singolo, per una o più fatture/scadenze relative allo stesso beneficiario. La SCA è richiesta ad ogni operazione, fatte salve specifiche esenzioni previste dalla normativa.

Inizializzazione pagamento

Inizializza un pagamento tramite SCT.

```
[POST] /{itemId}/payments
```

Nella **request** in particolare abbiamo:

- `externalRef` un riferimento univoco al creditore, da utilizzare in seguito come elemento di ricerca
- `metadata` elementi di tracciabilità, da utilizzare per la riconciliazione
- `providerId` id del provider (ASPSP, la banca)
- `productCode` id del prodotto selezionato
- `psuIpAddress` indirizzo IP del richiedente
- `debtorAccount` conto del pagatore
 - `iban` iban
 - `currency` valuta
- `creditorAccount` conto del creditore
 - `iban` iban
 - `currency` valuta
 - `creditorName` nome
- `amount` importo (espresso come numero intero in cui le ultime due cifre compongono la parte decimale)
- `currency` valuta
- `requestedExecutionDate` data di esecuzione
- `requestedExecutionTime` ora di esecuzione
- `remittanceInformation` descrizione (causale)
- `paymentProduct` SCT o SCTinst

- `tppRedirectUri` URL di redirect

Per conoscere `productCode` e `providerId` va preventivamente invocato [l'endpoint per la lettura dei conti](#).

Nella **response** l'API fornisce:

- `paymentId` id del pagamento (al momento ciascun gruppo contiene un solo pagamento)
 - `status` stato dell'operazione
 - `totalTransactionFees` commissioni totali della transazione
 - `scaManagerRedirectUrl` URL di redirect allo SCA manager
-

Esito pagamento

Interroga l'esito del pagamento.

```
[GET] /{itemId}/payments/{paymentId}
```

I **parametri** richiesti sono:

- `paymentId` path variable - id del pagamento, ritornato dalla [chiamata all'inizializzazione dello stesso](#)
- `psuIpAddress` query string - indirizzo IP del client

Nella **response** l'API fornisce:

- `paymentId` id del pagamento
- `status` stato dell'operazione
- `totalTransactionFees` commissioni totali della transazione
- `totalAmount` importo totale della transazione
 - `amount` importo (espresso come numero intero in cui le ultime due cifre compongono la parte decimale)
 - `currency` valuta
- `debtorAccount` iban del debitore
- `creditorAccount` iban del creditore
- `createdOn` data operazione

Consenso SCA

Effettuare una SCA

Descrizione

In questa sezione vengono elencati gli endpoint che consentono di ottenere e rinnovare i vari consensi necessari al collegamento di un conto.

Ottenimento consenso una-tantum per mostrare la lista dei conti

Consente di effettuare la SCA (usa e getta) per accedere ai conti/carte in proprio possesso presso la banca, per il prodotto definito nel productId.

```
[POST] /{itemId}/consents/accountsList
```

Nella **request** in particolare abbiamo:

- `providerId*` id della banca (o provider)
- `productCode*` codice del prodotto
- `accountNature*` natura dell'account ("account" o "card")
- `tppRedirectUri` URL di redirect dopo che la SCA viene effettuata

Nella **response** l'API fornisce:

- `consentId` id del consenso
- `consentStatus` stato del consenso
- `scaManagerRedirectUrl` URI per effettuare la SCA
- `validUntil` data fine validità

Ottenimento consenso ricorrente

Consente di effettuare la SCA (valida tipicamente 90 giorni) per collegare ai conti/carte in proprio possesso presso la banca, per il prodotto selezionato.

```
[POST] /{itemId}/consents/recurring
```

Nella **request** in particolare abbiamo:

- `providerId*` id della banca (o provider)
- `productCode*` codice del prodotto
- `tppRedirectUri` URL di redirect dopo che la SCA viene effettuata
- `accountAccess*` oggetto composto da
 - `balances*` lista di oggetti con la seguente forma:
 - `resourceId*`
 - `iban`
 - `maskedPan`
 - `currency*`
 - `accountNature`
 - `transactions*` lista di oggetti con la stessa forma di `balances`
 - `accounts` lista di oggetti con la stessa forma di `balances`

Nella **response** l'API fornisce:

- `consentId` id del consenso
- `consentStatus` stato del consenso
- `scaManagerRedirectUrl` URI per effettuare la SCA
- `validUntil` data fine validità

Rinnovo di un consenso

Consente di rinnovare la SCA (valida tipicamente 90 giorni) a partire da un consenso precedentemente valido.

```
[PUT] /{itemId}/consents/{consentId}/renew
```

I **parametri** richiesti sono:

- `consentId` path variable - id del consenso.

Nella **response** l'API fornisce:

- `consentId` id del consenso
- `consentStatus` stato del consenso
- `scaManagerRedirectUrl` URI per effettuare la SCA
- `validUntil` data fine validità

Revoca di un consenso

Consente di rinnovare la SCA (valida tipicamente 90 giorni) a partire da un consenso precedentemente valido.

```
[DELETE] /{itemId}/consents/{consentId}
```

I **parametri** richiesti sono:

- `consentId` path variable - id del consenso.

Nella **response** l'API torna il consenso stesso, con lo status aggiornato.

Recuperare i conti collegati ad un consenso

Consente di recuperare l'elenco dei conti/carte in proprio possesso presso la banca a fronte del consenso precedentemente fornito.

```
[GET] /{itemId}/consents/{consentId}/accounts
```

I **parametri** richiesti sono:

- `consentId` path variable - id del consenso.

Nella **response** l'API fornisce:

- `accounts` lista degli account composti dalle seguenti proprietà
 - `accountId` stato del consenso
 - `resourceId` id della risorsa
 - `iban` codice IBAN
 - `currency` valuta di conto
 - `providerId` id della banca
 - `productCode` codice del prodotto
 - `consentId` id del consenso

Banche e Prodotti

Descrizione

In questa sezione vengono elencati gli endpoint che consentono di ottenere informazioni riguardante le banche e i prodotti associati.

Banche

Consente di recuperare l'elenco delle banche.

```
[GET] /{itemId}/banks
```

Nella **response** l'API fornisce:

- `id` id della banca (o provider) da utilizzare come *bankId* (o *providerId*) nelle varie chiamate
 - `aspspCode` codice ASPSP
 - `satus` stato
 - `businessName` descrizione della banca
 - `aisInfo` informazioni relative al servizio [Conti](#)
 - `globalBankConsent` consenso globale supportato
 - `supportedNatures` nature supportate (conto, carta o entrambi)
 - `logo` URL del logo della banca
 - `readyToUse` indica che la banca è pronta per essere utilizzata dal servizio
 - `supportedPaymentProducts` modalità di pagamento supportate
 - `SCT-IT` SCT Italia
 - `SCT-EU` SCT Europa
 - `IP` SCT istantaneo
 - `CB` Cross-border
 - `T2` Target 2
-

Prodotti

Consente di recuperare l'elenco dei prodotti di una determinata banca.

```
[GET] /{itemId}/banks/{bankId}/products
```


Il parametro `bankId` fa riferimento all'id della banca ottenibile chiamando l'endpoint [Banche](#) sopra.

Nella **response** l'API fornisce:

- `aspspProductCode` codice del prodotto da utilizzare come *productCode* nelle varie chiamate
- `aspspProductDescription` descrizione prodotto estesa
- `aspspProductUuid` uuid del prodotto
- `aspspProductSuggestedLabel` descrizione sintetica del prodotto

Settings

Descrizione

In questa sezione vengono elencati gli endpoint riguardanti le impostazioni/configurazioni di pay-gateway

Configurazione di pagamento

[GET] /{itemId}/settings/config

Response:

- `minChargeAmount`: importo minimo transazione (in centesimi)
- `maxChargeAmount`: importo massimo transazione (in centesimi)
- `balanceRefundReserve`: esposizione massima per rimborso (in centesimi)
- `minWireAmount`: importo minimo accredito (in centesimi)
- `maxWireAmount`: importo massimo accredito (in centesimi)
- `netChargeEnabled`: incasso tramite SDD al netto abilitato
- `payoutBlocked`: accredito bloccato

Casi d'uso

Collegamento conto

Per collegare un conto o carta è necessario seguire i seguenti passi:

1. Selezione della banca a cui appartiene il conto da collegare - [Banche](#)
2. Selezione del prodotto bancario a cui appartiene il conto - [Prodotti](#)
3. Prestazione del consenso una-tantum per accedere ai conti e carte - [Consenso una-tantum](#)
4. Selezione dei conti e carte da collegare tra quelli disponibili legati al consenso precedente - [Recuperare i conti collegati ad un consenso](#)
5. Prestazione del consenso ricorrente per collegare i conti e carte - [Ottenimento consenso ricorrente](#)
6. Elenco dei conti e carte collegati con un determinato consenso - [Recuperare i conti collegati ad un consenso](#)
7. Aggiornamento dei dati - [Aggiornamento conti](#)

Notifiche

Overview

Il sistema di notifiche degli eventi che arrivano da TsPay sfrutta NCS per il loro smistamento all'interno di Digital.

Le applicazioni che si integrano via pay-gateway, e che hanno effettuato la configurazione per ricevere le notifiche via NCS, riceveranno tutte le notifiche, comprese quelle relative ad eventi scatenati da altre applicazioni, che sono anch'esse integrate via pay-gateway. Questo per due motivi:

1. Tutte le applicazioni che si integrano via pay-gateway si "presentano" a TsPay come singola applicazione.
2. Tutte le applicazioni del mondo digital potenzialmente operano sulle stesse aziende e quindi vogliono rimanere allineate sullo stato delle stesse.

È compito quindi della singola applicazione filtrare le notifiche desiderate dalle indesiderate, generalmente tenendo traccia dell'id dell'evento.

Flusso notifiche

Notifiche

Le notifiche vengono inviate al verificarsi di determinati “eventi”.

Ogni **Evento** è composto da due parti, **Entity** e **State**, nella forma `entity.state`: l'**Entity** è il soggetto principale (ad esempio `tspay_pis`) mentre **State** si riferisce allo status dell'Entity (ad esempio `active`).

Esempi di eventi:

- `tspay_charge.active`
- `tspay_payout.refunded`
- `tspay_registration.active`

Gli eventi previsti, con i relativi stati, sono i seguenti:

Servizio	Entity	State	Descrizione
Incasso (3)	tspay_source	<code>active</code>	Memorizzazione sorgente Link2Save andata a buon fine
		<code>error</code>	Memorizzazione sorgente Link2Save andata in errore
	tspay_charge	<code>pending</code>	Per l'SDD, nel tempo necessario all'elaborazione del pagamento
		<code>active</code>	Pagamento andato a buon fine
		<code>error</code>	Pagamento andato in errore
		<code>refunded</code>	Pagamento rimborsato
		<code>disputeCreated</code>	Contestazione creata
		<code>disputeWithdrawn</code>	Contestazione con importo trattenuto

		disputeUpdated	Contestazione aggiornata a seguito di invio prove
		disputeClosedLost	Contestazione persa
		disputeClosedWon	Contestazione vinta
		disputeClosed	Contestazione chiusa
		disputeRefunded	Contestazione rimborsata
	tspay_wire	pending	Accredito in corso
		done	Accredito eseguito
		failed	Accredito fallito
		reverted	Accredito stornato
		expired	Accredito non confermato
		error	Accredito in errore
Pagamento (7)	tspay_pis	pending	Pagamento in corso
		active	Pagamento eseguito
		failed	Pagamento fallito
		error	Pagamento andato in errore
Conti (8)	tspay_consent	created	Consenso creato
		renewed	Consenso rinnovato
		in_expiration	Consenso in scadenza

Schema

Tutte le notifiche hanno il seguente schema:

- `itemId` company registry dell'azienda per la quale si è verificato l'evento notificato
- `entity` entità
- `state` stato
- `event` evento

- `eventTime` timestamp
- `payload` parte variabile dipendente dal tipo di evento. Dettagli in basso.

Payload specifici per entità:

Di seguito i riferimenti allo schema dei vari payload:

- Notifiche da entità `tspay_source`: [specifiche payload](#)
- Notifiche da entità `tspay_charge`: [specifiche payload](#)
- Notifiche da entità `tspay_wire`: [specifiche payload](#)
- Notifiche da entità `tspay_pis`: [specifiche payload](#)
- Notifiche da entità `tspay_consent`: [specifiche payload](#)

Payload notifiche

tspay_charge

Campo	Tipo	Descrizione
amount	integer	importo lordo
currency	string	valuta
chargeKey	string	id riferimento pagamento
state	string	stato pagamento
upstreamRef	string	flusso stati
orderKey	string	id riferimento ordine
customerKey	string	id riferimento cliente
upstreamStatus	string	stato pagamento
fees	array	commissioni
netAmount	integer	importo netto
modifiedOn	string	data di modifica
createdOn	string	data di creazione
payMethod	string	metodo di pagamento (card, SDD)
order	array	dettaglio richiesta di riferimento

Payload notifiche

tspay_source

Campo	Tipo	Descrizione
orderKey	string	id richiesta di riferimento
sourceKey	string	id riferimento sorgente di pagamento
createdOn	string	data creazione
modifiedOn	string	data modifica
type	string	tipo sorgente di pagamento (card, sepa_debit)
last4	string	ultime 4 cifre del numero identificativo della sorgente (carta o IBAN)
expiration	string	data di scadenza in formato mm/yyyy (solo carta)
brand	string	tipo di network (Visa, MasterCard, American Express, ecc...) (solo carta)
funding	string	tipologia di carta (credit, debit, prepaid) (solo carta)
order	array	dettaglio richiesta riferimento

Payload notifiche tspay_pis

Campo	Tipo	Descrizione
paymentId	string	id del pagamento
amount	long	importo
currency	string	valuta
remittanceInfo	string	descrizione (causale)
debtorIban	string	iban del debitore
creditorIban	string	iban del creditore
creditorName	string	nome del creditore
externalRef	string	un riferimento univoco al creditore, da utilizzare in seguito come elemento di ricerca
metadata	string	elementi di tracciabilità, da utilizzare per la riconciliazione
providerId	string	id del provider (ASPSP)
productCode	string	id del prodotto
paymentProduct	string	SCT o SCTInst

Payload notifiche

tspay_consent

Campo	Tipo	Descrizione
<code>type</code>		non utilizzato
<code>consentId</code>	string	id del consenso
<code>providerId</code>	string	id della banca (o provider)
<code>productCode</code>	string	codice prodotto
<code>validUntil</code>	string	data scadenza consenso
<code>externalRef</code>	string	riferimento esterno
<code>metadata</code>	string	dati di tracciabilità
<code>accounts</code>	array	elenco conti/carte riferiti al consenso

ogni oggetto dell'array `accounts` ha i seguenti campi:

Campo	Tipo	Descrizione
<code>iban</code>	string	codice iban (per i conti)
<code>maskedPan</code>	string	numero della carta (criptato)
<code>currency</code>	string	valuta

Payload notifiche tspay_wire

Campo	Tipo	Descrizione
amount	integer	importo lordo
beneName	string	nome beneficiario
createdOn	string	data di creazione
currency	string	valuta
fees	array	commissioni
iban	string	iban beneficiario
modifiedOn	string	data di modifica
netAmount	integer	importo netto
payScheduleRef	string	referimento accredito
state	string	stato
wireDesc	string	descrizione accredito
wireKey	string	id accredito

Configurazione corrente NCS

Attualmente NCS è configurato per inviare le notifiche:

1. via **webhook** agli applicativi
2. via **email** agli utenti

1. Attualmente agli applicativi integrati, che si sono configurati su NCS, vengono inviate tutte le notifiche in arrivo da TsPay. Essendo via webhook, **l'applicativo deve esporre un endpoint su cui essere notificato**. Il payload della chiamata sarà (come già specificato [qui, in schema](#)):

```
{
  "itemId": "{{itemId}}",
  "entity": "{{entity}}",
  "state": "{{state}}",
  "event": "{{event}}",
  "eventTime": "{{eventTime}}",
  "payload": "{{payload}}"
}
```

2. Oltre alle notifiche via webhook per gli applicativi, sono configurate anche le notifiche via email per i soli seguenti eventi: `tspay_consent.renewed`, `tspay_consent.in_expiration`, `tspay_consent.created`.