

Platební brána ThePay

Implementační manuál

Obsah

Princip fungování brány.....	3
Kroky platby online metodou	3
Kroky platby offline metodou	3
Komponenta pro implementaci	4
PHP API implementační komponenty	4
Třída TpMerchantConfig	4
Třída TpDivMerchantHelper	4
Třída TpRadioMerchantHelper	4
Třída TpButtonMerchantHelper	4
Třída TpPayment	4
Třída TpEetDph	5
Třída TpReturnedPayment	5
Použití komponenty	5
Zobrazení platebního tlačítka	5
Návrat zákazníka – přijetí platby	5
Dodatečné ověření stavu platby	6
Použití platební komponenty v podobě radio buttonů	6
Zobrazení radio buttonů	6
Přizpůsobení vzhledu	7
Zpracování platby	7
Návrat zákazníka – přijetí platby	7
Testovací platební brána	7
Parametry API	8
Parametry platby	8
Generování podpisu	10
Parametry pro zobrazení platební komponenty s výběrem metody	10
Parametry pro výběr metody na straně ThePay	11
Napojení na EET	11
Návrat z brány na obchodníkovu stránku	11
Volání brány – vytvoření platby	12
Online platba	13

Offline platba	13
Kartové operace	13
Stržení částky	13
Storno platby	13
Opakované platby / zapamatování karty	14
Zjištění informací o kartě	14
Implementace	14
Metoda cardDepositPaymentRequest (TpCardHelper.depositPayment)	14
Metoda cardStornoPaymentRequest (TpCardHelper.stornoPayment)	14
Metoda cardCreateRecurrentPaymentRequest (TpCardHelper.createNewRecurrentPayment). ..	15
Metoda getCardInfoRequest (TpCardHelper. getCardInfo)	16
Trvalé platby	16
Co jsou trvalé platby	16
Omezení	16
Implementace	17
Metoda createPermanentPaymentRequest (TpPermanentPaymentHelper.createPermamentPayment)	17
Metoda getPermanentPaymentRequest (TpPermanentPaymentHelper.getPermamentPayment)	17
Response (objekt TpPermanentPaymentResponse)	18
Objekt TpPermanentPaymentResponseMethod	18
Žádost o vrácení platby	18
Implementace	18
Metoda returnPaymentRequest (TpPaymentReturnHelper. returnPayment)	18

Princip fungování brány

Fungování brány se liší dle použité platební metody. Platební metody mohou být buď online (celá transakce je provedena ihned online – např. platba kartou) nebo offline (zákazník musí provést nějakou akci, např. poslat SMS). Online metody jsou synchronní – zákazník provede transakci a vrací se zpět na Váš web. Offline metody jsou asynchronní – uživatel dostane instrukce pro platbu, někdy později platbu provede a Vaše aplikace je o tom informována HTTP requestem.

Kroky platby online metodou

1. Zákazník na Vašem webu (např. v e-shopu) běžným způsobem vybere zboží či službu, kterou chce zaplatit.
2. V okamžiku, kdy má zaplatit, je zobrazena platební komponenta ThePay.
3. Zákazník si vybere platební metodu, je přesměrován na platební bránu a zaplatí.
4. Po zaplacení je zákazník přesměrován zpět na Váš web a v požadavku jsou zaslány podrobnosti o platbě.
5. Vy zpracujete platbu (uložíte informaci o zaplacení) a zobrazíte zákazníkovi potvrzení o zaplacení, případně cokoliv jiného, co je potřeba.
6. Některé platební metody umožňují provést platbu, které nemusí být fyzicky realizována ihned (např. eKonto od Raiffeisenbank umožní provést platbu i když zákazník nemá na kontě dostatek financí. Pokud do dvou dnů peníze doplní, platba se realizuje, v opačném případě nikoliv). V takovém případě je zákazník vrácen se statusem `TpReturnedPayment::STATUS_WAITING` a v okamžiku kdy platbu přijmeme odešleme další notifikaci, tentokrát se stavem `TpReturnedPayment::STATUS_OK`. Pokud notifikace nepřijde, platba nebyla realizována.

Kroky platby offline metodou

1. Zákazník na Vašem webu (např. v e-shopu) běžným způsobem vybere zboží či službu, kterou chce zaplatit.
2. V okamžiku, kdy má zaplatit, je zobrazena platební komponenta ThePay.
3. Zákazník si vybere platební metodu a jsou mu zobrazeny instrukce pro platbu.
4. Zákazník provede platbu. To může provést ihned, ale i později (např. za dva dny).
5. V okamžiku, kdy přijmeme platbu, brána zavolá Váš skript (pošle HTTP HEAD požadavek na Vámi zadanou návratovou adresu) a předá mu informace o platbě.
6. Vy zpracujete platbu.

Jaké platební metody budou dostupné zákazníkům si můžete nastavit v administračním rozhraní. Pokud používáte jen online metody, nemusíte při implementaci řešit zpracování offline plateb a naopak.

Do aplikace je tedy třeba implementovat pouze dvě věci:

- Zobrazení platební komponenty
- Zpracování platby po zaplacení

Obě úlohy za Vás vyřeší námi dodaná komponenta, ale pokud ji z jakýchkoliv důvodů nemůžete či nechcete použít (např. používáte jiný jazyk než PHP), můžete tuto funkčnost implementovat sami.

Komponenta pro implementaci

PHP API implementační komponenty

Komponenta zapouzdřuje API platební brány – generování komponenty do stránky a zpracování platby po návratu zákazníka z brány (nebo přijetí notifikace o zaplacení). Veškerý kód komponenty je okomentovaný a je dodávána s ukázkovou implementací. Pokud tedy implementujete bránu do aplikace v jazyce PHP, nemusíte řešit samotné API, stačí si projít rozhraní a ukázkou použití komponenty.

Třída **TpMerchantConfig**

Slouží k uložení konfigurace obchodníka přístupu do systému ThePay.

Atribut **merchantId** obsahuje číselné ID obchodníka v systému.

Atribut **accountId** obsahuje číselné ID obchodu v rámci obchodníka účtu (jeden obchodník může mít více účtů, např. pro několik e-shopů).

Atribut **password** obsahuje heslo pro autentizaci obchodníka při komunikaci se systémem. Toto heslo by mělo být odlišné od toho, které používá obchodník pro přihlášení do administrace.

Všechny údaje jsou automaticky přiděleny systémem a zjistit je lze po přihlášení do administračního rozhraní. Všechny údaje jsou automaticky přidělovány systémem, ID jsou neměnná, heslo je možné v případě potřeby na žádost změnit.

Třída **TpDivMerchantHelper**

Výchozí metoda zobrazení platebního tlačítka. AJAXem stáhne kód komponenty ze serveru ThePay a zobrazí jej jako div element. Volitelně je možné vypnout výchozí vzhled a použít vlastní stylování všech prvků komponenty.

Třída **TpRadioMerchantHelper**

Alternativní metoda zobrazení platební komponenty v podobě radio buttonů. AJAXem stáhne kód komponenty ze serveru ThePay a zobrazí jej jako div element s radio buttony pro výběr platební metody místo tlačítek. Podporuje integraci s jinými platebními metodami na straně e-shopu (např. dobírka, osobní odběr atp.).

Třída **TpButtonMerchantHelper**

Slouží k zobrazení jednoduchého „donate“ tlačítka, po jehož stisknutí se provede přesměrování na server ThePay, kde bude možnost vybrat platební metodu. Helper je možné buď využít pro zobrazení grafického tlačítka, nebo pro získání URL adresy pro platbu, kterou můžete použít dle potřeb Vašeho systému. Z důvodu lepšího konverzního poměru doporučujeme tuto možnost využít pouze v případě, je-li to opravdu nutné, např. z technologických důvodů.

Třída **TpPayment**

Obsahuje informace o platbě. Pomocí metody `getSignature()` je možné vygenerovat kontrolní součet pro tuto platbu (parametr `signature` při volání brány).

Třída TpEetDph

Představuje rozdělení celkové částky platby na dílčí úhrny sazeb DPH pro systém EET. Nastavení částek a pojmenování atributů kopíruje oficiální pojmenování systému EET. Instance této třídy se předává instanci TpPayment metodou setEetDph().

Třída TpReturnedPayment

Tato třída slouží k ověření a zpracování návratového požadavku ze systému ThePay. Třída automaticky kontroluje kontrolní součet parametrů a v případě chybného kontrolního součtu vyhodí výjimku TpInvalidSignatureException.

Použití komponenty

Zobrazení platebního tlačítka

```
$p = new TpPayment(new TpMerchantConfig());
$p->setValue(499); // cena zboží
$p->setDescription("Nákup zboží v našem e-shopu"); // popis platby - nepovinné
$p->setMerchantData("1662"); // data obchodníka, např. ID objednávky- nepovinné
$p->setReturnUrl("http://www.adresa-eshopu.cz/url-pro-navrat/"); // na tuto adresu bude
// zákazník přesměrován po zaplacení
// případná další nastavení, viz. třída TpPayment
$hlp = new TpDivMerchantHelper($p);
echo $hlp->render(); // zobrazení komponenty s platebními tlačítky
```

Komponenta se zobrazí jako div vygenerovaný javascriptem. Je možné ji libovolně nastýlovat dle vašich požadavků. Načtení výchozího CSS je možné úplně vypnout (viz třída TpDivMerchantHelper) a nastýlovat si komponentu dle vlastních požadavků.

Návrat zákazníka – přijetí platby

```
$p = new TpReturnedPayment(new TpMerchantConfig());
if($p->verifySignature()){
    // platba je platná – obsahuje všechny nezbytné údaje a podpis je platný
    if($p->getStatus() == TpReturnedPayment::STATUS_OK){
        // platba je zaplacená, můžeme si ji uložit jako zaplacenou a dále zpracovat (vyřídít
        // objednávku atp.).
    } else if($p->getStatus() == TpReturnedPayment::STATUS_UNDERPAID){
        // zákazník zaplatil pouze část platby
    } else if($p->getStatus() == TpReturnedPayment::STATUS_CARD_DEPOSIT){
        // částka byla zablokována na účtu zákazníka – pouze pro platbu kartou
    } else if($p->getStatus() == TpReturnedPayment::STATUS_CANCELED){
        // zákazník platbu stornoval
    } else if($p->getStatus() == TpReturnedPayment::STATUS_ERROR){
        // při zpracování platby došlo k chybě
    } else if($p->getStatus() == TpReturnedPayment::STATUS_WAITING){
```

```

        // platba proběhla úspěšně, ale čeká na potvrzení od poskytovatele platební metody.
        S vyřízením objednávky je nutno počkat na potvrzovací request se statusem TpReturnedPayment::
        STATUS_OK, pokud nepřijde, platba nebyla dokončena.
    }
} else {
    // neplatný požadavek – nejde o návrat z brány, došlo k nějaké chybě, parametry byly
    podvrženy...
}

```

Jde-li o návrat z platební metody probíhající online, zobrazí stránka zákazníkovi potvrzení o zaplacení, nebo informaci o chybě.

Při obdržení informace o zaplacení přes offline platební metodu musí stránka vrátit HTTP kód 200 nebo 302 (může fungovat stejně jako pro zákazníka, tělo odpovědi je bránou ignorováno). Pokud odpověď obsahuje jiný HTTP kód, bude notifikace považována za nedoručenou a brána se během následujících několika hodin bude pokoušet o opakované doručení.

Notifikace offline platebních metod jsou zasílány HTTP metodou HEAD, přesměrování zákazníka u online metod je klasický GET požadavek.

Dodatečné ověření stavu platby

Pomocí datového API, konkrétně metody **getPaymentState**, můžete ověřit stav platby v našem systému dle id platby (návratový parametr *paymentId*).

Tím zvýšíte bezpečnost celého procesu platby. Teoreticky se může stát, že útočník nějakým způsobem získá obchodníkové heslo a podvrhne požadavek od našeho systému. Dodatečné ověření stavu platby takovému typu útoku zabrání.

Chcete-li zvýšit zabezpečení ještě více, použijte metodu **getPayment**, která vrací všechny informace o platbě, a zkontrolujte nejen stav platby, ale i hodnotu *merchantData* (oproti očekávané hodnotě, nikoliv oproti hodnotě z requestu). Tím zabráníte možnosti podvržení id jiné platby.

Podrobnější informace naleznete v dokumentaci k datovému API v souboru **Datové API v1.3.doc**

Použití platební komponenty v podobě radio buttonů

Při použití komponenty v podobě radio buttonů je zobrazení a zpracování platby rozděleno na dva samostatné kroky, je tedy třeba implementace na dvou místech. Pro zobrazení není třeba znát údaje o platbě, zobrazení slouží pouze pro výběr platební metody. Samotné zpracování se provede později, až to bude potřeba (po zpracování objednávky e-shopem atp.).

Zobrazení radio buttonů

```

$help = new TpRadioMerchantHelper($config, 'method_select', '3');
echo $help->renderRadio();

```

Atribut **name** obsahuje jméno původního radiobuttonu s výběrem platebních metod.

Atribut **value** obsahuje hodnotu původního radio buttonu s platební metodou ThePay.

Máte-li ve Vašem systému např. platbu na dobírku s id 1, platbu v hotovosti při osobním převzetí s id 2 a platbu přes ThePay s id 3 a radio buttony s výběrem platby mají name="method_select", použijete výše uvedený ukázkový kód.

Pokud používáte jen platbu přes ThePay a nemáte jiné platební metody, můžete oba atributy nechat prázdné.

Helper v této fázi nepotřebuje znát údaje o platbě, slouží jen k výběru platební metody.

Přizpůsobení vzhledu

Atribut **showIcon** slouží pro zobrazení nebo skrytí grafických ikon platebních metod (defaultně zapnuté).

Atribut **appendCode** může obsahovat HTML kód, který se zobrazí na konci každého divu s platební metodou (můžete zde vložit např. cenu, nějaký grafický prvek atp.).

Je-li na stránce definovaná JavaScriptová funkce **tp_after_load_callback** bude tato zavolána po vyrenderování platební komponenty (po stažení kódu AJAXem a po jeho vložení do stránky).

Zobrazený kód není nijak nastýlovaný, stylování výsledného zobrazení je plně ve Vašich rukou.

Zpracování platby

Po zpracování platby na straně e-shopu (uložení objednávky do db atp.) je potřeba zavolat metodu na přesměrování zákazníka na platební bránu pro online platby

```
if($hlp->isTpMethodChosen()){
    $p = new TpPayment($config);
    // inicializace platby
    $hlp->redirectOfflinePayment($p);
}
```

Metoda **isTpMethodChosen** vrací true pokud byla v předchozím kroku vybrána platební metoda ThePay.

Metoda **redirectOfflinePayment** provede přesměrování zákazníka na platební bránu pokud byla vybrána online platební metoda.

Pokud vybraná metoda není online, je třeba zákazníkovi zobrazit instrukce pro platbu na stránce. Na stránce s potvrzením objednávky zobrazíme instrukce pro platbu takto:

```
echo $hlp->showPaymentInstructions($p);
```

Obě metody pracují jen pokud byla vybrána platba přes ThePay a platební metoda příslušného typu. Tzn. podmínka na `$hlp->isTpMethodChosen()` není nutná, ale ušetří vytváření objektů pro platbu.

Návrat zákazníka – přijetí platby

Zpracování návratu zákazníka je totožné s platební komponentou v podobě tlačítek – popis viz výše.

Testovací platební brána

Pro otestování funkčnosti Vaší implementace můžete použít testovací platební bránu. Testovací brána funguje totožně jako ostrá brána, jen místo přesměrování na platební systém zobrazí

parametry platby a tlačítko pro simulaci zaplacení. Návrat do aplikace obchodníka proběhne identicky jako při skutečném zaplacení.

Asynchronní notifikace o dokončení platby lze simulovat přes speciální formulář.

Adresa brány:	https://www.thepay.cz/demo-gate/
Id obchodníka:	1
Id účtu:	3
Heslo:	my\$up3rsecr3tp4\$\$word
Heslo pro Datové API	my\$up3rsecr3tp4\$\$word
Test notifikací	https://www.thepay.cz/demo-gate/testNotif.php

Parametry API

Parametry platby

Tyto parametry jsou společné všem způsobům zobrazení platebního tlačítka i přímému volání brány.

Nevyužíváte-li napojení na EET přes ThePay, můžete ignorovat všechny parametry začínající na *eet*.

Parametr	Datový typ	Výchozí hodnota	Popis
merchantId	int	-	ID obchodníka
accountId	int	-	ID obchodníkovy účtu
value	decimal	-	Částka k zaplacení ve formátu \d+\.\d\d (např. 350.00 nebo 4.80)
currency	string	CZK	Kód měny dle ISO 4217
description	string	""	Stručný popis platby (zobrazen zákazníkovi)
merchantData	string	""	Libovolná data, která se po zaplacení vrátí obchodníkovi
customerEmail	string	null	E-mailová adresa zákazníka. Je-li vyplněna, bude mu odeslán notifikační e-mail o vytvoření platby a potvrzení po zaplacení.
customerData	string	null	Fakturační adresa zákazníka v podobě JSON encoded objektu. Používáno pro autorizaci karetních plateb v systému 3D Secure 2.0. Data by měla být shodná s údaji majitele karty. Data jsou nepovinná, ale jejich absence může vést ke zvýšení počtu zamítnutých plateb kartou.

	Parametry objektu (všechny jsou nepovinné)		
	Parametr	Popis	
	full_name	Jméno a příjmení zákazníka	
	country	Země ve formátu ISO 3166-1 Alpha-2	
	city	Město	
	postcode	PSČ	
	street	Ulice včetně čísla popisného	
	email	E-mailová adresa. Není-li předána, použije se customerEmail	
returnUrl	string	-	Adresa, kam má být zákazník přesměrován po zaplacení a na kterou budou zasílány notifikace o změně stavu platby
backToEshopUrl	string	URL adresa účtu obchodníka	Adresa, na kterou bude odkazovat tlačítko "Zpět do e-shopu" na stránce s platebními údaji offline platebních metod
merchantSpecificSymbol	string	-	Specifický symbol zadáný obchodníkem. (neplést s návratovým parametrem specificSymbol) Je zobrazen zákazníkovi, ale pro zpracování platby se nevyužívá.
eetZaklNepodlDph	decimal	-	EET - Celková částka plnění osvobozených od DPH, ostatních plnění v Kč
eetZaklDan1	decimal	-	EET - Celkový základ daně se základní sazbou DPH v Kč
eetDan1	decimal	-	EET - Celková DPH se základní sazbou v Kč
eetZaklDan2	decimal decimal	-	EET - Celkový základ daně s první sníženou sazbou DPH v Kč
eetDan2	decimal	-	EET - Celková DPH s první sníženou sazbou v Kč
eetZaklDan3	decimal	-	EET - Celkový základ daně s druhou sníženou sazbou DPH v Kč
eetDan3	decimal	-	EET - Celková DPH s druhou sníženou sazbou v Kč
eetCestSluz	decimal	-	EET - Celková částka v režimu DPH pro cestovní službu v Kč
eetPouzitZboz1	decimal	-	EET - Celková částka v režimu DPH pro prodej použitého zboží se základní sazbou v Kč
eetPouzitZboz2	decimal	-	EET - Celková částka v režimu DPH pro prodej použitého zboží s první sníženou sazbou v Kč
eetPouzitZboz3	decimal	-	EET - Celková částka v režimu DPH pro prodej použitého zboží s druhou sníženou sazbou v Kč
eetUrcenoCerpZuct	decimal	-	EET - Celková částka plateb určená k následnému čerpání nebo zúčtování v Kč
eetCerpZuct	decimal	-	EET - Celková částka plateb, které jsou následným čerpáním nebo zúčtováním platby v Kč
signature	string	-	MD5 hash všech výše uvedených a předaných parametrů v pořadí uvedeném

			v dokumentaci + obchodníková tajného hesla.
--	--	--	---

V případě přímého volání brány jsou tyto parametry součástí požadavku (funguje GET i POST metoda). Při použití PHP komponenty se vezmou z třídy TpMerchantConfig (merchantId, accountId a heslo obchodníka), nastaví příslušnými set metodami (value, currency, description, merchantData, returnUrl) či automaticky vygenerují (signature)

Generování podpisu

Podpis je MD5 hash všech výše uvedených parametrů platby, které nejsou nulové (prázdné parametry jsou ignorovány), kromě parametru customerData (ten není součástí podpisu), a obchodníková hesla ve tvaru parametrů pro GET požadavek. Pro účely podpisu nejsou parametry urlencodovány.

Pořadí parametrů v podpisu musí odpovídat pořadí uvedenému v tomto manuálu.

Všechny částky jsou ve formátu s dvěma desetinnými místy a tečkou jako oddělovačem desetinných míst (\d+.\d\d). Parametry typu boolean jsou pro účely podpisu reprezentovány hodnotou **0** nebo **1**.

Příklad:

Jsou-li hodnoty následující: merchantId = 1, accountId = 1, value = 999, description = popis, returnUrl = http://www.nejaky-obchod.cz/return/, heslo = 12345

potom je signature vygenerován takto: signature =
md5("merchantId=1&accountId=1&value=999.00&description=popis&returnUrl=http://www.nejaky-obchod.cz/return/&password=12345")

Parametry pro zobrazení platební komponenty s výběrem metody

PHP komponenta na základě zadaných parametrů vygeneruje adresu JS skriptu (viz třída TpDivMerchantHelper), který stáhne kód platební komponenty a zobrazí ji na stránce jako obyčejný DIV element.

Speciální parametry:

Parametr	Dat. typ	Vých. hodnota	Popis
disableButtonCss	boolean	false	Vypne includování CSS stylu pro nastavení vzhledu platební komponenty. Komponentu je pak nutné nasylovat ručně.
disablePopupCss	boolean	false	Vypne includování CSS stylu pro nastavení vzhledu pop-up okna s instrukcemi pro offline platby. Pop-up okno je pak nutné nasylovat ručně.

Po výběru platební metody jde uživatel na adresu <https://www.thepay.cz/gate/>, předané parametry viz Volání brány – vytvoření platby.

Při ručním vytvoření kódu komponenty je možné tento způsob platby použít také na statickém webu.

Parametry pro výběr metody na straně ThePay

Potřebujete-li, aby si zákazník vybral platební metodu až na platební bráně ThePay, nikoliv již na Vašem webu, musí být v požadavku přítomen parametr **methodSelectionAllowed**. Parametr může mít libovolnou hodnotu, může být i prázdný, důležitá je jeho přítomnost. Třída `TpButtonMerchantHelper` parametr doplňuje automaticky.

Tento parametr **není součástí podpisu**.

Napojení na EET

Platební brána ThePay umožňuje automatické napojení na EET. Pro funkčnost napojení je nutné, aby obchodník ve své administraci nahrál certifikát pro komunikaci, nastavil svým účtům číslo provozovny a komunikaci s EET aktivoval.

Pokud obchodník prodává v rámci jednoho e-shopu (jedno `accountId`) pouze zboží v jedné sazbě DPH, používá výpočet DPH shora (vypočítává částku daně a daňový základ z částky s DPH) a celkovou částku platby nezaokrouhluje na celé koruny, není nutné měnit implementaci a předávat data pro EET. Platební brána si vše potřebné dopočítá sama dle sazby DPH, kterou si obchodník zvolí v nastavení účtu.

V ostatních případech je nutné předat rozpad celkové částky na dílčí úhrny podle jednotlivých sazeb DPH. Jednotlivé částky jsou předávány jako parametry při vytváření platby, viz [Parametry platby](#). Při použití implementační komponenty je pro tento účel připravena třída `TpEetDph`. Dílčí částky předané touto třídou budou beze změny odeslány do systému EET. Platební brána pouze ověří základní smysluplnost údajů, za jejich správnost odpovídá obchodník.

Návrat z brány na obchodníkovu stránku

Po úspěšném dokončení platby je provedeno přesměrování zpět na stránku obchodníka specifikované v `returnUrl` parametru volání brány.

Předané parametry GET metodou na obchodníkovu stránku jsou uvedeny v následující tabulce

Parametr	Datový typ	Popis
value	decimal	Částka platby.
currency	string	Měna platby.
methodId	int	Identifikace platební metody použité k zaplacení.
description	string	Popisek specifikovaný při zadávání platby.
merchantData	string	Libovolná data specifikovaná při odesílání požadavku.
status	enum	Stav platby, možné hodnoty (konstanty třídy <code>TpReturnedPayment</code>) jsou: STATUS_OK (2) – úspěšně zaplaceno STATUS_CANCELED (3) – nezaplaceno, platba byla zrušena zákazníkem STATUS_ERROR (4) – při platbě došlo k chybě. Objeví-li se tento stav, kontaktujte naši technickou podporu. STATUS_UNDERPAID (6) - zákazník zaplatil nižší než požadovanou částku. Může nastat u platebních metod využívajících např. obvyčejné bankovní převody. STATUS_WAITING (7) - zákazník platbu provedl, ale je nutné

		počkat na potvrzení od poskytovatele platebního systému (až platba dorazí, bude zaslána notifikace) STATUS_CARD_DEPOSIT (9) - pouze u plateb kartou – částka je blokována na účtu zákazníka, ke stržení dojde po odeslání požadavku paymentDeposit skrz API
paymentId	int	Identifikační číslo platby vygenerované systémem ThePay. Toto ID je vždy zcela unikátní v rámci celého systému.
ipRating	int	DEPRECATED
isOffline	0 1	Zda se jedná o offline (isOffline=1) nebo online metodu (isOffline=0).
needConfirm	0 1	DEPRECATED
isConfirm	1	DEPRECATED
customerAccountNumber	string	Číslo účtu zákazníka. Pouze pro platební metody, které číslo účtu používají a poskytují. Tato informace je přístupná pouze na vyžádání, pro bližší info nás kontaktujte.
customerAccountName	string	Jméno účtu zákazníka (obvykle jeho jméno). Pouze pro platební metody, které používají a poskytují číslo účtu (stejně jako pro customerAccountNumber). Tato informace je přístupná pouze na vyžádání, pro bližší info nás kontaktujte.
signature	string	MD5 hash všech zaslaných parametrů + obchodníková hesla.

Parametry *needConfirm* a *isConfirm* byly plně nahrazeny stavem STATUS_WAITING a budou v budoucnu z API odstraněny.

Pro notifikace trvalých plateb jsou navíc předávány tyto parametry:

Parametr	Datový typ	Popis
specificSymbol	string	Specifický symbol zadáný zákazníkem, pokud byl zadán

Parametr jsou součástí signature v případě, že jde o trvalou platbu a symbol byl při platbě vyplněn (jinak se vůbec neposílá a není součástí podpisu).

Pro platbu kartou jsou navíc předávány tyto parametry:

Parametr	Datový typ	Popis
deposit	0 1	Příznak, zda má být částka rovnou stržena z účtu (1) nebo jen zablokována (0) a stržena bude později (voláním API, viz níže). Výchozí hodnota 1
isRecurring	0 1	Zda se má platba zaregistrovat jako opakovaná (později bude možné přes API strhávat další platby). Výchozí hodnota 0.

Parametr jsou součástí signature v případě, že je vyplněn (při použití výchozí hodnoty se nemusí posílat a pak není součástí podpisu).

Volání brány – vytvoření platby

Odesláním GET/POST požadavku na <https://www.thepay.cz/gate/> dojde k samotnému zpracování platby. Všechny výše uvedené metody nakonec vedou na toto volání, kde dojde k samotnému zavolání platební metody.

Online platba

Z této stránky je zákazník přesměrován na stránku platební brány a po provedení platby okamžitě zpět na adresu specifikovanou v parametru returnUrl.

Brána přijímá následující parametry, předané buď GET, nebo POST metodou:

Parametr	Datový typ	Výchozí hodnota	Popis
merchantId	int	-	ID obchodníka
accountId	int	-	ID obchodníkovy účtu
value	decimal	-	Částka k zaplacení
currency	string	CZK	Kód měny dle ISO 4217
description	string	""	Stručný popis platby (zobrazen zákazníkovi)
merchantData	string	""	Libovolná data, která se po zaplacení vrátí obchodníkovi
returnUrl	string	-	Adresa, kam má být zákazník přesměrován po zaplacení.
backToEshopUrl	string	URL adresa účtu obchodníka	Adresa, na kterou bude odkazovat tlačítko "Zpět do e-shopu" na stránce s platebními údaji offline platebních metod
methodId	int	-	Identifikace platební metody, která se má použít k zaplacení.
signature	string	-	MD5 hash všech výše uvedených a předaných parametrů v pořadí uvedeném v dokumentaci + obchodníková tajného hesla.

Offline platba

V případě offline platby jsou zobrazeny instrukce pro zaplacení (konkrétní informace se mohou lišit dle konkrétní platební metody) a tlačítko „Pokračovat“, které vede zpět na stránku obchodníka

Parametry offline platby jsou stejné jako u online platby.

Při uhrazení offline platby je zavolána adresa specifikovaná v parametru returnUrl, aby obchodník mohl zpracovat platbu ve svém systému.

Kartové operace

Platby kartou umožňují několik speciálních operací v závislosti na parametru deposit a isRecurring specifikovaných při vytvoření platby. Platby jsou identifikovány na základě hodnoty parametru merchantData – pro jejich využití tedy musí atribut mít jedinečnou hodnotu (což jinak není nutné).

Stržení částky

V případě, že byla platba vytvořena s parametrem deposit=0, byla částka na účtu zákazníka pouze zablokována a nedošlo k stržení částky a jejímu převodu na naše účty. Je tedy nutné zavolat speciální operaci pro stržení částky. Pokud to nebude provedeno, částka bude nějaké době automaticky odblokována a platba nebude realizována.

Storno platby

V případě, že byla platba vytvořena s parametrem deposit=0, je možné tuto platbu stornovat a prostředky na účtu zákazníka opět odblokovat.

Opakované platby / zapamatování karty

Byla-li platba vytvořena s parametrem `isRecurring=1`, je možné kdykoliv později vytvořit novou platbu a strhnout zákazníkovi další peníze. Vytvoří se kopie původní platby s novou, vámi specifikovanou, částkou (pozor na příznak `deposit`, pokud měla původní platba `deposit=0`, bude s touto hodnotou vytvořena i nová platba).

Zjištění informací o kartě

O platební kartě, kterou byla platba provedena, je možné přes API zjistit podrobnější informace. Konkrétně masku čísla karty, karetní asociaci, která kartu vydala, banku, zemi a typ karty. Tyto informace lze např. zobrazit zákazníkovi při využití opakovaných plateb, aby věděl, z jaké karty platí, nebo například využít pro detekci potenciálních podvodů.

Implementace

Služba je implementována jako web service, jejímž voláním realizujete požadovanou operaci.

WSDL je umístěno na adrese <https://www.thepay.cz/gate/api/gate-api.wsdl>, WSDL pro testovací účely je na adrese <https://www.thepay.cz/demo-gate/api/gate-api-demo.wsdl>

Pro implementaci můžete použít třídu `TpCardHelper`.

Metoda `cardDepositPaymentRequest (TpCardHelper.depositPayment)`

Strhne z účtu zákazníka peníze zablokované na základě platby s parametrem `deposit=0`. V rámci účtu obchodníka musí existovat právě jedna platba s předanou hodnotou `merchantData`, platba musí mít příznak `deposit=0` a musí být ve stavu umožňujícím stržení peněz (nesmí být stornované, peníze nesmí být již strženy atp.). Peníze na účtu jsou blokovány jen určitou dobu (záleží na bance), takže pro starší platby nemusí operace fungovat.

Request

Parametr	Datový typ	Popis
<code>merchantId</code>	<code>int</code>	ID obchodníka
<code>accountId</code>	<code>int</code>	ID obchodníkovy účtu
<code>merchantData</code>	<code>string</code>	Data obchodníka platby
<code>signature</code>	<code>string</code>	MD5 hash všech výše uvedených parametrů v pořadí uvedeném zde + obchodníkovy tajného hesla.

Response

Parametr	Datový typ	Popis
<code>status</code>	<code>boolean</code>	Výsledek operace. <code>TRUE</code> = platba byla úspěšně stržena z účtu zákazníka, <code>FALSE</code> = došlo k chybě
<code>errorDescription</code>	<code>string</code>	Popis chyby, pokud <code>status = FALSE</code>

Metoda `cardStornoPaymentRequest (TpCardHelper.stornoPayment)`

Stornuje platbu, která ještě nebyla stržena z účtu zákazníka. V rámci účtu obchodníka musí existovat právě jedna platba s předanou hodnotou `merchantData`, platba musí mít příznak `deposit=0` a musí být ve stavu umožňujícím stornování (nesmí být již stornovaná, peníze nesmí být již strženy atp.).

Request

Parametr	Datový typ	Popis
merchantId	int	ID obchodníka
accountId	int	ID obchodníkovu účtu
merchantData	string	Data obchodníka platby
signature	string	MD5 hash všech výše uvedených parametrů v pořadí uvedeném zde + obchodníkovu tajného hesla.

Response

Parametr	Datový typ	Popis
status	boolean	Výsledek operace. TRUE = platba byla úspěšně stornována, FALSE = došlo k chybě
errorDescription	string	Popis chyby, pokud status = FALSE

Metoda cardCreateRecurrentPaymentRequest (TpCardHelper.createNewRecurrentPayment)

Vytvoří novou opakovanou platbu na základě již existující platby vytvořené s parametrem isRecurring=1. V rámci účtu obchodníka musí existovat právě jedna platba s předanou hodnotou merchantData a musí mít příznak isRecurring=1. Pozor na příznak deposit - byla-li původní platba vytvořena s deposit=0, bude mít deposit=0 i nová platba. Pro vytvoření platby je třeba předat hodnotu merchantData pro novou platbu.

Request

Parametr	Datový typ	Popis
merchantId	int	ID obchodníka
accountId	int	ID obchodníkovu účtu
merchantData	string	Data obchodníka platby první platby, která byla autorizována zákazníkem
newMerchantData	string	Data obchodníka nově vytvořené platby
value	decimal	Částka nové platby, která bude stržena ze zákaznickova účtu
signature	string	MD5 hash všech výše uvedených parametrů v pořadí uvedeném zde + obchodníkovu tajného hesla.

Response

Parametr	Datový typ	Popis
status	boolean	Výsledek operace. TRUE = nová platba byla úspěšně vytvořena a stržena z účtu zákazníka, FALSE = došlo k chybě
errorDescription	string	Popis chyby, pokud status = FALSE

Metoda `getCardInfoRequest` (`TpCardHelper.getCardInfo`)

Získá informace o kartě, kterou byla platba uhrazena. Obchodník musí mít tuto funkci povolenou a platba musí být uhrazena kartou.

Request

Parametr	Datový typ	Popis
merchantId	int	ID obchodníka
accountId	int	ID obchodníkovy účtu
paymentId	string	ID platby
signature	string	MD5 hash všech výše uvedených parametrů v pořadí uvedeném zde + obchodníkovy tajného hesla.

Response

Parametr	Datový typ	Popis
status	boolean	Výsledek operace. TRUE = platba byla úspěšně stornována, FALSE = došlo k chybě
errorDescription	string	Popis chyby, pokud status = FALSE
cardNumberMasked	string	Maska čísla karty ve tvaru 123456*****1234
cardBrand	string	Jméno karetní asociace, např. MC nebo VISA
countryCode	string	Jméno banky, která kartu vydala
bankName	string	Dvoumístný kód země dle ISO 3166-1 alpha-2
cardType	string	Typ zaúčtování karty, např. DEBIT, CREDIT
cardLevel	string	Druh karty, např. BUSINESS, CLASSIC, PREPAID

Trvalé platby

Co jsou trvalé platby

Běžná platba je pouze jednorázová a nelze znovu použít její variabilní symbol. Pro další platbu je nutné zákazníkovi znovu zobrazit platební komponentu, kde si vybere platební metodu a vytvoří se nová platba.

Trvalá platba umožňuje získat jedinečný variabilní symbol, pod kterým může zákazník opakovaně posílat platby. Všechny příchozí platby pod příslušným variabilním symbolem budou vyplaceny obchodníkovi.

Trvalé platby mají smysl pouze pro omezené množství použití - např. placení předplatného, splátek atp. Pokud to není Váš případ, nemusíte se jimi vůbec zabývat.

Omezení

- trvalé platby je možné použít pouze pro platbu bankovním převodem, nikoliv pro platební tlačítka a jiné platební metody

- pod jedním VS je možné přijmout nejvýše jednu platbu za den

Implementace

Služba je implementována jako web service, jejímž voláním založíte trvalou platbu a získáte údaje pro platbu, které zobrazíte zákazníkovi.

WSDL je umístěno na adrese <https://www.thepay.cz/gate/api/gate-api.wsdl>, WSDL pro testovací účely je na adrese <https://www.thepay.cz/demo-gate/api/gate-api-demo.wsdl>

Pro implementaci můžete použít třídu `TpPermanentPaymentHelper`.

Metoda `createPermanentPaymentRequest`

(`TpPermanentPaymentHelper.createPermanentPayment`)

Vytvoří novou trvalou platbu a vrátí platební informace pro zobrazení zákazníkovi. Pokud pro použitý obchodníkův účet již existuje trvalá platba se stejnou hodnotou `merchantData`, jsou vráceny platební údaje pro tuto platbu a nová platba se nevytváří.

Request (objekt `TpPermanentPayment`)

Parametr	Datový typ	Popis
config	<code>TpMerchantConfig</code>	Konfigurace obchodníka
merchantData	string	Jednoznačný identifikátor transakce. Používá se při vyhledávání trvalých plateb. Posílá se obchodníkovi v notifikaci příchozí platby v atributu <code>merchantData</code> .
description	string	Volitelný popis platby pro potřeby obchodníka, uživateli se nezobrazuje.
returnUrl	string	Adresa, na kterou bude zaslána notifikace po přijetí platby.
signature	string	MD5 hash všech výše uvedených a předaných parametrů v pořadí uvedeném v dokumentaci + obchodníkova tajného hesla. Vytváří se automaticky.

Response

Stejně jako u metody `getPermanentPaymentRequest`

Metoda `getPermanentPaymentRequest`

(`TpPermanentPaymentHelper.getPermanentPayment`)

Získá platební informace pro platbu s příslušnou hodnotou `merchantData`. Novou platbu nevytváří, pokud neexistuje, vrátí prázdný výsledek.

Request

Parametr	Datový typ	Popis
merchantId	int	ID obchodníka
accountId	int	ID obchodníkova účtu
merchantData	string	Data obchodníka platby
signature	string	MD5 hash všech výše uvedených parametrů v pořadí

		uvedeném zde + obchodníková tajného hesla.
--	--	--

Response (objekt TpPermanentPaymentResponse)

Parametr	Datový typ	Popis
status	boolean	Stav požadavku. True pokud byly předané parametry validní a platba byla vytvořena/nalezena. False pokud došlo k chybě.
errorDescription	string	Textový popis chyby ke které došlo. Pokud status=true, je prázdné
paymentMethods	Array of TpPermanentPaymentResponseMethod	Pole platebních údajů pro trvalou platbu. Tyto údaje zobrazte zákazníkovi.

Objekt TpPermanentPaymentResponseMethod

Reprezentuje platební údaje pro konkrétní platební metodu.

Parametr	Datový typ	Popis
methodId	integer	Jedinečný identifikátor metody.
methodName	string	Název platební metody.
url	string	URL adresa internetového bankovníctví.
accountNumber	string	Číslo účtu (kompletní včetně kódu banky případně předčíslí).
vs	integer	Variabilní symbol.

Žádost o vrácení platby

Tato funkce je určena pro vrácení uhrazené platby zpět zákazníkovi v případě, že není možné splnit objednávku z důvodů na straně obchodníka (nedostupné zboží atp.) nebo zákazník žádá zrušení platby. Vrátit je možné pouze celou částku platby.

O vrácení uhrazené platby zpět zákazníkovi lze požádat dvěma způsoby. První možností je ruční zadání požadavku v detailu platby v klientské sekci. Druhou možností je vytvoření žádosti přes API.

Žádost může být zamítnuta v případě, že obchodník nemá na účtu dostatečný zůstatek.

O výsledku zpracování žádosti je obchodník informován e-mailem.

Implementace

Služba je implementována jako web service, jejímž voláním realizujete požadovanou operaci.

WSDL je umístěno na adrese <https://www.thepay.cz/gate/api/gate-api.wsdl>, WSDL pro testovací účely je na adrese <https://www.thepay.cz/demo-gate/api/gate-api-demo.wsdl>

Pro implementaci můžete použít třídu TpPaymentReturnHelper.

Metoda returnPaymentRequest (TpPaymentReturnHelper. returnPayment)

Vytvoří žádost o vrácení platby identifikované parametrem paymentId (ID platby v platební bráně), volitelně je možné předat důvodem vrácení. Platbu je možné vrátit pouze v případě, že to umožňuje její stav (je uhrazená nebo částečně uhrazená) a žádost o její vrácení ještě nebyla vytvořena.

Request

Parametr	Datový typ	Popis
merchantId	int	ID obchodníka
accountId	int	ID obchodníkovu účtu
paymentId	Int	ID platby
reason	string	Volitelný důvod vrácení platby
signature	string	MD5 hash všech výše uvedených parametrů v pořadí uvedeném zde + obchodníkovu tajného hesla.

Response

Parametr	Datový typ	Popis
status	boolean	Výsledek operace. TRUE = žádost o vrácení platby byla vytvořena, FALSE = došlo k chybě
errorDescription	string	Popis chyby, pokud status = FALSE