PayBook : Alternative Trading System (ATS) with Paypal Express Checkout API Guide Dev EN

(Chrysoberyl version)

The French version is the primary source for all translations. For non French-speaking or non English speaking, there are tools as Translation Google translate (http://translate.google.fr/translate_t#) to read documents in your language.

Problems & Vision:

Go to principles: Creating a world without poverty in order to fund and improve education, reducing disease and reducing mortality.

The paradigm shift

The existing paradigm solves some problems, so it was accepted by the past. However, at this moment, it does not meet the demand of everyone. So there was questioning and demand change.

I studied the market for existing solutions. Not being satisfied (because if I was happy the issue was already resolved), so I decided to create this software. Because, in my opinion, we must phase out barriers to trade. (ref Treaty establishing the European Community, 1957)

The software "Paybook" is designed to build a capital market alternative to existing financial markets. The software specializes in raising funds online. It facilitates free trade in the international economy. This software was created to meet the financing needs of economic agents in Europe and globally in order to finance, among other things, research, education and innovation.

Principles & Qualities:

- Freedom, Citizenship, Responsibility, Equality, Solidarity,
- Open, rule based, predictable, nondiscriminatory

Objective:

Make a pilot project (functional, no set-up fees, free to use, legal, available (7/7j, 24/24h) and based on a win-win strategy) for an international trading platform from a PSP payment API (Paypal Express Checkout) and the FreePay Trading System (FTS) to help raise account balance of participants. (+1 Euro each time)

Directed:

The software has been tested in the "PayPal Sandbox" with the first accounts for the 3 possible cases of purchase (less, equal and higher) and in each case, it worked perfectly.

Simple example: when 1 Euro was sent, are received 2 Euros (factor 2: 1 * 2 = 2). The starting amount is chosen by the user and may use the system as often as he wants. Example: 1234 Euros sent -> 2468 Euros received.

Target:

Target User: This software is being implemented, primarily targeting users of the PSP used speaking French or English and have minimal skills in finance (PayPal registers 141 million customers, according to PayPal). The goal is not to limit it to this category but can reach the largest number of users.

Target Developer: Those targeted for downloading and installing the software are programmers (preferably web developer) and / or companies wishing to establish a trading platform.

The platform does not take a percentage of the funds it raises. In this sense, an organization that would use could be of the organization non-profit.

Strategies for doing business

There are 3 possible strategies for 2 people: lose, no deal or win. (ie 9 combinations in total)

Case study: I do not want to lose money (destruction of money) I do not want to lose money for someone wins. (theft of money) I want to change. I do not make money for someone to lose. (theft of money) I do not want to make money alone. (counterfeit) I want to make money and someone else wins, too.

The only solution that seems feasible is: the win-win strategy.

Business Model of a trading platform in line with a win-win strategy:

5 parts: Provider, Consumer, Competitor, Partner and Himself.

Provider:
- The Payment Service Provider (PSP) have a business model that works by commissions (fixed and variable) on the transactions of their customers (about 2%). To increase their profits, they want to increase their volume of transactions and that customers send the most money.
- Their main partners are the merchants and trading platform that allows them to increase their volume of transactions and amounts over the users who use them. This is done by increasing the number of customers that is proportional to the access of their information systems: the API (Application Programming Interface) that allows anyone to automate payments.
- Their suppliers are one or more banks. The financial messages are handled electronically by the bank. The PSP received confirmation via an API.

Consumer:

- The internet users want to meet their needs. The needs of humanity are recurrent (must be met every day). This need is either a product or service, or money. The products are among the online marketplaces (e-commerce). The easy money is on the trading platform.
- To send money, the user needs a PSP. PSP and asked to have a bank account to fulfill its electronic wallet. Compared to the PSP, the user earns more per transaction if the gain is at its default value (maximum).

Competitor, Partner and Self:
- The business model of the trading platform is either the same as the PSP (for transactions), or an entrance fee or monthly fee or premium sponsorships (link id) or is free. (The money is earned in the same way as users.)
- Providers of trading platforms are the PSP with their API. (the logo is highlighted)
- The users use a service that allows them to move money. The strategy of moving money is defined either by the trading platform or by users themselves.
(depends on the internal politics of the trading platform)
- The trading platform is a software layer above the PSP. The design and evolution of the computer product can be partially outsourced to one or more persons caring for an open source trading platform that would reduce costs and development time.

Interoperability: How do I know if the API is an electronic money institution financial supports of the FSX FreePay?

Financial Institution:

1. It must be able to create an account.

2. Supply: The user can supply his account with different means of payment (check, credit card, bank transfer etc.) and remove.

3. It must have a minimum of funds in its account (balance at least 1 euro). (must also take into account the costs of financial institution)

Optional (but strongly recommended): Approval Financial: The financial institution must be approved by at least one regulator.

Separation of tests and the real: The customer transfers between normal and test client are prohibited.

On the FSX:

1. It must be able to pass an order on the FSX and get on the payment platform. (POST or GET)

2. B2B, B2C, C2B & C2C: Let the payment works in 2 directions (merchant-merchant while being accessible client-customer, merchant-customer and customer-merchant), briefly

allow P2P. The right of withdrawal depends on the status of persons making the transactions and all, it is defined on the site of PSP.

2a. (optional: but it's better to do) should be able to enable process automation. (XML)

3. XML: It is necessary that the source site (merchant / FSX) to obtain a trace of the transaction from the payment platform. (xml sent and saved in the database tables in sql)

Why a relationship with a PSP:

The division of roles: The software can be seen as a plugin that interacts with the main software (financial institution) to bring him a new feature. The software is an open system that sends information to the internal (history) and outside (order).

- The financial institution converts capital into e-money, make payments and make the conversion of checks.
- The software can place orders for payment, exchanging payment orders and can make claims.

Dependence: The software is towards simplicity as compared to the previous version (FreePay), he subtracted the processes needed to manage money. This software saves the cost of initial capital (1 million Euros) for the creation of a financial institution issuing electronic money (in: e-money issuer) within the European Union. (ref: Article 4 paragraph 1 of Directive 2000 46 EC).

Independence: Each organization that sets up the software Give1Get2 is autonomous from other organizations. It is dependent only financial institution which helps to make payments (1 to many relationship).

PSP's Choice:

We should not make a money transfer (transaction) before checking that we can save the trace. Because otherwise the person will be disappointed to have paid without obtaining its counterpart.

Given the rapid evolution of payment systems, it is preferable that the API manages versions so that older versions still work.

The issue of security has also been a decisive factor.

The PayPal Express Checkout API was chosen because it repect constraints.
http://www.paypal.fr/presentation

Paypal SandBox API : https://developer.paypal.com/

We must upgrade the account to buy and sell with the system: Upgrade the account> choose Premier (or business account).

To obtain credentials to the API:
- Click on My Acount> Profile> API Access> Request API Credentials> API Signature> Agree and Submit
- Copy and Paste the Username, Password and the signature to authenticate to the platform.

== PayPal Sandbox ==

paypal sandbox login email: me@mypreferedhost.com
password paypal sandbox: a

== alice (premier) ==

login email PayBook : alice@mypreferedhost.com
password PayBook : ab

email Paypal : alice_1234524871_per@mypreferedhost.com
password Paypal : abc

API Username: alice_1234524871_per_api1.mypreferedhost.com
API Password: PSHQHGQGRDMH2Y7F
Signature: Ajd8FF2KxieV-6BxPDy4odWHclXnAjl3TA568AvD4KSTbwfGqKT84ljS

== bob (premier) ==

login email PayBook : bob@mypreferedhost.com
password PayBook : a

login email Paypal : bob_1228753395_per@mypreferedhost.com
password Paypal : ab

API Username: bob_1228753395_per_api1.mypreferedhost.com
API Password: 5396CWRJP62HES74
Signature: AFcWxV21C7fd0v3bYYYRCpSSRl31AXl353cqalhXTzu4SnL2g5ZYdjTa

To raise money on his paypal account:
Accommodation> Main Page> My recent activity> Application Status / Action> Collect / capture> collection batch> see all search> check all> check the selected items> Collections> batch sent> My Account

The financial institution may allow constraints on the visual integration of its logo on a home page at different mode of payments and options.
https://www.paypal.com/fr/cgi-bin/webscr?cmd=xpt/Marketing/general/AcceptanceMarkLogos-outside

What is "Paybook?

The trading system is based on a win-win. The trading platform is a place of confrontation in the supply of financial securities and the demand for money under the idea of laissez-faire economics. There are no goods exchanged on the system. It is a zero sum game in terms of the payment platform but not the trading system (1 euro securitized token issued for the initiation, exchange +1 +1 for each party to each transaction). It's a virtuous circle. There is no entrance fee. It is a system of person to person (P2P) which further allows users to place trades on a payment platform. This was designed so that there is no risk of inverse proportionality. Since there is no order of sale, it may not be a stock market crash. All system users can get rich, but not at the same time. The user can then become, as it makes a trader. (en: Market Operator)

The gain is also adjustable (0 to 100% Sample: 25 euros become real to 100% -> 50 euros securitized). This allows the user to speculate whether or not to do so. This allows the user to transform its capital and more capital represented by shares. (And then convert its shares capital by the sale, eg 50 euros securitized -> 50 euro real). Finally: EUR 25 real -> 50 euro real. This was to be demonstrated.

Legal Explanation:

The ISIN code is used again in this software (ISO 6166). The consolidation created the ZZ is to make a clear distinction and there have no ambiguity with the countries or territories with securities, according to ISO 3166-1.

Economic Explanation: The software is not intended to create inflation.

From what I know, there are two types of inflation:

- Inflation of prices: Higher prices for goods and services during a period of time. (source: Wikipedia) The income increases more slowly as rising commodity prices. → decrease in purchasing power. What is problematic. But what has Give1Get2 is to increase the income of players in the system, thereby increasing purchasing power. There are no services for sale on the platform and use is free.

The "products" are selling financial claims payments. The purchase price is determined by the buyers themselves. If they decide to buy more expensive it is to earn more.

- Inflation of the money: When money suffers a global money creation. Money in circulation increases via interest rates. However, the software Give1Get2 not intended to increase the money supply or decrease it. There is no interest rate not in this system. For only the responsibility of banks.

The software allows the movement of money between players.

Monetary Explanation: There is no money exchanged in the trading system. The money is only exchanged on the PSP.

Financial Explanation:

Bob sends money to Alice. Bob is the debtor. Alice is the creditor. If variable costs and fixed costs are higher than that received by Alice, Bob and Alice are negatively charged so it is therefore a lose-lose situation.

For a win-lose, we can establish a minimum quota of money to be set in automatic settings for all purchases through the API.

The trading process works as follows: The seller exchange goods cons money from the buyer. The seller of the property does that against a sum greater than what it cost him at first with these suppliers.

Similarly, when that person A makes a claim and receives money from a person B. This acts as a compensatory mechanism that allows B in turn place a claim for a price equal to or greater than what it cost him to his predecessor.

Computer Science's Explanation:

Traditionally, it is a relationship with 2 people only. The problem if one of the 2 party may pay or reimburse. (Relationship 1 to 1) With this software alternative market, there are multiple suppliers and multiple plaintiffs. (Relation of many to many) So, this reduces the risk (1 to many relationship in 2 directions) and there is an advantage of opportunities for success.

As I demonstrate, we can turn a losing strategy in a win-win strategy. That's what we do now.

Prerequisite:

User:
- The only equipment needed is a PC, operating system, Internet connection and web browser. There is nothing to download for users.
- Multi-platform: runs on Windows or Linux using Internet Explorer or Firefox.
- The training is free. It is the documentation.

Developer:
- Designed by XHTML, CSS, JavaScript, PHP and SQL (CRUD).
- Requires MySQL, phpMyAdmin, POP3 for mail, FTP and a web browser.
- Has been tested and works with Apache (> = 1.3.33), MySQL (> = 4.1.9), PHP (> = 4.3.10) with the cURL extension installed and PhpMyAdmin (> = 2.6.1).

Download

The software is based on a policy of transparency and sustainable development. Licenses chosen is the GNU GPL. It is free software. Thus, it has been made available free on SourceForge.net to be downloaded and installed on servers online.

http://sourceforge.net/projects/paybook/
http://paybook.sourceforge.net/

Software Installation

1. Buy a domain name (myonlinetradingplatformsample.com) with a Registrar.

2. Getting accommodation containing enough space (approx 60 Mb) and sufficient bandwidth (several Giga) depending on the number of users expected. (and POP3, FTP, and MySQL included)

3. De-compress the files previously downloaded. (as above)

4. Edit the file 'scripts sql tables & champs.sql':
- Replace email in the table freepay_titre
- Replace customer information in the table freepay_client

5. Create a database "Paybook" (without the double quotes) in your control administration (usually the URL http://myonlinetradingplatformsample.com/phpmyadmin/)
- Create a user and give access rights to the database for reading and writing. (if this is not done automatically).

6. Click the SQL tab, paste the data file 'scripts sql tables & champs.sql' in the textbox and click Run. No error message should appear.

7. Change the default values by those who have been provided by the host in the file 'params.php' (without the single quotes) host, user, password, database.

8. On the web server, copy and paste the modified source (with default settings) in 7z and zip. Also create a folder /paybook/. Upload files via FTP (ex FireFTP, a Firefox extension) with the parameters of the host ('params.php') in the directory previously created.

9. Launch the browser http://myonlinetradingplatformsample.com/paybook/. The index page should be displayed without an error message. Sources (7z and zip) should be downloadable from a tab 'documentation' or 'download'.

10. Suggest your site on search engines (eg http://www.google.com/addurl/?continue=/addurl)

11. Generate an XML sitemap and put it in the root (eg http://www.xml-sitemaps.com/)

12. Optimize your website (eg with Google Webmaster Tools https://www.google.com/webmasters/tools)

The process explains how to switch from "Give1Get2" to the next version "Paybook":

Mashup Give1Get2 without Moneybookers (for architecture) + FreePay (for additions) + Paypal Express Checkout (for connection) = Paybook ATS

http://sourceforge.net/projects/freepay/
http://sourceforge.net/projects/give1get2/
https://developer.paypal.com/
http://developer.paypal-portal.com/pdn/board?board.id=fr

- find a test server
  - delete the existing db test to avoid slip-ups.
  - create basic user rights
- update software ftp
  - create ftp user
  - delete ftp test
- create a subdomain.
  - Internal change the logo (not changed the other)
  - Install scripts. + Db
- Delete the default page if necessary
- rename the installation directory
- configure with the settings (be careful with prefixes)

Interface:

- creation of account (email, pass, API_username, API_password, API_signature)
  - import script "client_ajouter.php" and its interface FreePay
  - nav top to change the link to "client_ajouter.php"
  - add to the interface API_username, API_password, API_signature "
  - change the treatment on "API_username, API_password, API_signature"
  - modify the db on "API_username, API_password, API_signature"
  - add disclaimer
  - information account creation api verified by paypal (trying a command)
  - change url absolute and relative current directory
  - any error messages
  - recovery post on error
  - PayPal API error display well formatted
  - API function paypal town client_ajouter.php and client_modifier.php
  - optimize the verification ACK
- login (email, password)
  - add link "open account"
  - change default directory
  - check login and pass FreePay
- disconnection
  - change default directory
- changing the account (mdp, api)

    - import "client_modifier.php"
    - create interface in separate file
    - Modified treatment
    - Info modified api verified account with paypal (trying a command)
    - the person changes its api credentials and it still works (by paypal, we can not modify)
    - optimize the verification ACK
    - PayPal API error display well formatted
- menu
  - change "mb_paiement_en_attente.php" in psp_ordre_afficher.php
  - change the menu link
- JavaScript Core Curriculum:
  - JavaScript validation registration (EN + FR) (longer)
  - JavaScript validation connection (EN + FR) (same for short)
  - Warning message javascript i18n
  - define the encoding for file European UTF-8
- follow the guidance of visual integration
    - change the PSP logo
    - follow the guidelines
    - change the meta tags (keywords)
    - logo change trading platform

- Treatment:
  - create account paypal sandbox
    - first create accounts with API
    - be connected to PayPal sandbox for testing
  - GetExpressCheckout
    - edit table "freepay_achat_en_attente" with token
    - edit integration with token (the seller)
    - retrieve API_username, API_password, API_signature from the token (the seller)
  - treatment (status_report.php)
    - If> =:
      - money transfer?
      - Change of Ownership?
      - history?
      - new tariff?
    - if <:
      - money transfer?
      - deduction under former guard and former owner
      - create new title and new owner
      - histo new title

- db:
  - table "Freepay_titre" substitute "goods / id" APIs by info (username, password, signature)
  - edit table fields
  - modify request

Redirecting to Paypal

CP: reading directories for documentation (faster than typing by hand)
Official documentation move the PSP in the "paypal" Project
rename files for conventions of writing
also include the source document cited in the external doc internal compliance with copyright
photo paper at the end of poverty (too heavy (8 MB), abandoned)

delete:
- mb_afficher.php
- nav_transaction.php
- Services / ISIN / ISIN_verif.php
- support / plan.php
- support / moneybookers.php

check: no errors should appear when you type the name of the page in the browser
- paiement/titre/titre_acheter_interface2.php
  - Payment / title / titre_acheter_interface.php
- support / customer / client_modifier_interface.php
  - support / customer / client_modifier.php
- support / customer / client_ajouter_interface.php
  - support / customer / client_ajouter.php
- support / customer / connexion_interface.php
  - support / customer / connexion.php
- Pay / status_report.php
  - title / titre_consulter.php

test:
- success>
- success =
- success <

write documentation Give1Get2 (Dev, User, Slides)
- User's Guide: Making screenshots
  - IN
  - EN
- Dev Guide
- Slides

- Change the name of the project for compliance
- What happens if a person paying on the psp is not authenticated on FSX? A receives the money. B pays. B receives nothing. C does not receive the claim and pays nothing.
- User guide with screenshot in English
- Respect the conventions of graphic presentation of documentation (User Guide FR and EN)

- Remove unused images
- Automate the names of files for download
- Change the disclaimer at the connection
- Respect the naming conventions for files
- Load images for better quality

Architecture

The architecture is three tier (data, business logic and presentation). The architecture is based on Give1Get2 the project. (http://sourceforge.net/projects/give1get2/) To deepen the Give1Get2 documentation is available. (http://give1get2.sourceforge.net/give1get2/nav_telechargement.php?option=documentation)

The source code is in French. The source code comments are in French too. Except for the financial standards that are in English. The project is oriented towards internationalization (I18N).

All images are in a specific folder (/ images).
All CSS (Stylesheets cascading) are in a specific folder (/ style).
All that was attractive to user support is in the / support.
The documentation is in the / support / docs.
Everything concerning the internationalization is in the folder named "services/i18n.

The programming style is procedural: the methods are called in a specific order.

The visible part is composed of the main page of the history and documentation.

The existence of a claim can be verified through history.

The availability of a claim can be verified by applying the filter with the ISIN number as parameter.

As a web project, the human-machine interface is based on an architecture is client / server. And the server to a 3-tier architecture (database, processing, presentation).

The database

It is composed of 4 tables per financial institution: ("scripts sql tables & champs.sql" present at the root)

- Freepay_achat_en_attente: list of attempted transactions. (num_transaction `,` num_titre `,` valeur_titre `,` profit_titre `,` date_attente `,` mail_acheteur `,` mail_vendeur `,` status `,` token `)
- Freepay_client: Information Client API (mailcli `,` mdpcli, API_username ``, `API_password`, `API_signature`, `` datecreationcompte)
- Freepay_titre: list of securities for sale in their current states. (Numtitre ``, `datetitre`,

`valeurtitre`, `` mailcli)
- Freepay_titre_transferer: list of orders made. (DateT ``, `numtitre`, `mailcli`, `achattitre`, `` tauxprofit)

The stages of payment for a user

1. The user creates his account with the PSP, has supplied, has outperformed his account and retrieve the information API.

2. The user creates an account. (Link on the first page top right.)

3. The user goes to the purchase page and select an ISIN number (defined as the value and profit if needed).

The purchase order is saved and accessible via the menu of the same name. It summarizes the state of
transaction (Active / Standby, Fails / Canceled or Finished).

4. The user is redirected to the PSP. She did all the steps required for payment. And returned on the trading platform automatically.

5. The money was transferred by the PSP and the equivalent request for payment also. (Transfer possession of securities is based on new figures sent by the API to prevent any attempt of fraud.) The pending transaction "pending" changes to "done" with the transaction number recovered from the PSP. For each transaction, money is saved in the accounts of the financial institution. In a crisis (such as unavailability of the platform title), the money, it is always available.

6. The user can see his tracks and refresh the page (F5). It can also use the emailling to expedite the payment process, then:

- The seller receives an email notification informing the FSX the transfer of title and receipt of money. (Transmitter + money + currency)

- The buyer receives an email notification from the financial institution. (+ money + currency + product code)

The balance of the user is higher than it was before without the use of this application. This was to be demonstrated.

Each user can repeat the process as many times as he wants without restrictions.

7. The user disconnects from the trading platform (FSX) and PSP.

The stages of payment for a developer

The diagram of treatment processes the merchant side of the financial institution has been copied in an image software. Express Checkout flow.png "(" PayPal Sandbox User Guide (in English) p41: "Testing PayPal NVP APIs") version 2.3

Express Checkout is PayPal Name-Value Pair (NVP) API cURL is used.

Basically there are 3 functions (put in the same file name):
- SetExpressCheckout: Define payment information
- GetExpressCheckout: Get information on the buyer
- DoExpressCheckoutPayment: Execute payment and get confirmation of the execution of payment

1. The buyer selects a command. Information Form (amount + currency + language) and the information of the seller (API_username, API_password, API_signature) are injected into the SetExpressCheckout bound platform of payment (cURL).

2. The ACK SetExpressCheckout is a success or a failure.

3. If the ACK SetExpressCheckout is successful, the buyer is redirected to the payment platform (by specifying the token of the ACK in the url) via an HTTP redirect (JavaScript or HTML). The buyer logs in and confirms the order. (The buyer pays on the PSP)

4. The buyer is then redirected (using a return address provided in the SetExpressCheckout) on the ecommerce platform. The URL provided by the SPP contains a token and PayerID.

5. The token can be used to retrieve information from the transactions defined in the SetExpressCheckout. The information is then injected into a GetExpressCheckout,

6. then in a DoExpressCheckoutPayment.

7. If the ACK DoExpressCheckoutPayment is successful, the treatment of vendor run.

The status / states of a transaction: 2: validated (Processed), 1: test 0: Pending (Pending), -1: Canceled (Canceled) -2: Failed (Failed) -3: Backspace (Chargeback)

Optional:
- Record responses of the PSP in a table. (KISS: Keep it Simple, Stupid)
- To change the currency: set the CurrencyCode in SetExpressCheckout and DoExpressCheckoutPayment
- The local language is defined only in the SetExpressCheckout. (AU, DE, FR, GB, IT, ES, JP, U.S. accepted by the PSP)

The script takes into account that the fields by way of simplification. Regarding the optional fields: refer to official documentation.

Views:

The architecture views is common FreePay (header and footer together on all pages).

Menu: Home, History of trade, Online Payment, Transaction History, My claims, Contact, Documents

Controllers:

These are the same as those of FreePay. PHP and Javascript for client and server side respectively.

Transactions that fail after 1 day were classified as having failed. (status to -2).

Security

Sessions: Authenticated users can access the pages. Those unauthenticated are requested to do so via a redirection of url.

Authentication: The user data is verified with the PSP before they give their permission. I strongly advise against using the same password on PayPal. (because of phishing) passwords users are md5 encrypted in the database. In a change of user information, authorization is requested from the PSP for validation. (Data Security)

Filtering external data: There is a validation of incoming data (GET and POST). The files concerned are: the registration form, the login form, the form of changing preferences, the change form when purchasing an order, while the internal search engines, and finally the contact form . Services (PSP) are also filtered.

The application has been tested with CAL9000 (OWASP) to be protected against certain attacks like Cross Site Scripting (XSS). This type of attack is in the Top 10 vulnerabilities in 2007 by the Open Web Application Security Project (OWASP).

Test and Verify:

Web Performance Best Practices :
- Optimize images
- Serve resources from a consistent URL
- Avoid CSS expressions
- Combine external CSS
- Specify image dimensions
- Minimize redirects
- Put CSS in the document head
- Use efficient CSS selectors

Compliance with W3C standards: has been validated XHTML 1.0 Transitional and CSS 2.0 in Mozilla Firefox, Internet Explorer and Safari.

The test accounts are opened at the initiative of customers. Funding test are given free by the PSP.

Pass the following test series: Requires minimum 2 users. (Alice and Bob)

Preparation: Note the financial position of existing users: "Balance in Euro" and "ISIN Balance" for each.

Make a purchase and for 3 cases (less than, equal, higher), check:
- The balance of the buyer (Alice) has he fallen?
- The value of the security of the buyer (Alice) has she grown?
- The balance of the receiver (Bob) has he grown?
- The titles of the receiver (Bob) have decreased?
- The receiver (Bob) has it been notified by email?

Marketing:

- Search engine optimization
- (On Change:) Generate PAD file (Portable Application Description). Submitting the PAD file.
- PayPal Training Library
- Answer questions on forums
- Sign guru.com

Next development platform:

FaceBook

Legality:

I think my system is legal because I am doing research in this direction before putting in free. I believe it is in line with the principles of the European Union. (http://europa.eu/scadplus/european_convention/objectives_fr.htm). I have attached the file "Support"> "docs" reference documents concerning the legislative, legal and regulation which might be pertinent.

The website of the European Commission is very instructive on this issue. (http://ec.europa.eu/internal_market/top_layer/index_24_fr.htm) Category: Commission European> Market> Single Market for Services> Financial Services. I am not completely agree on the choice of this category since the trading platform offers a free service (without compensation) and does not manage money (only confirmation that money has been transferred).

Fund investments> Alternative investments: there is a Draft Directive on fund managers known as "alternative. The draft guideline can still be changed, and the final version will not necessarily apply to specific cases. http://ec.europa.eu/internal_market/investment/alternative_investments_fr.htm

Payment services> E-Money: The trading platform does not change. So, this concerns only the PSP. http://ec.europa.eu/internal_market/payments/emoney/index_fr.htm

Payment services> e-Invoicing: PSP is Moneybookers which manages billing (it can disable) Only one copy is kept for archive purposes by the trading platform (or anything depending on the setting)
http://ec.europa.eu/internal_market/payments/einvoicing/index_fr.htm

Financial conglomerates: Depending on the size of the trading system Paybook and structure, it can enter or not enter this category. It is the selection of the contractors as opportunities for merger / acquisitions.
http://ec.europa.eu/internal_market/financialconglomerates/index_fr.htm

Electronic Business: It depends on what is done by customers in return for the money sent.
http://ec.europa.eu/internal_market/e-commerce/directive_fr.htm

Copyright:

My copyright is protected by the GNU General Public License. http://www.gnu.org/licenses/gpl.html

My creations are protected at European level by Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs. (http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31991L0250:FR:HTML)

My designs are protected internationally by the Berne Convention for the Protection of Literary and Artistic Works (currently managed by the World Intellectual Property Organization (WIPO), specialized agency within the UN). (source: http://www.wipo.int/treaties/fr/ip/berne/trtdocs_wo001.html)

Disclaimer:

Using this application value of acceptance of the disclaimer Next: The author assumes no responsibility for any consequences that may result from using this application.

Script comes with no warranty.

The Organization

The Indefinite Lifespan Foundation is a charitable organization of nonprofit, nongovernmental, dedicated to reducing human mortality, to promote the extension of life expectancy and the pursuit of happiness using mainly preventive medicine, Public health and scientific computing. (Currently being created)

Contact:

As a developer, I always look for a way to produce more at lower cost. My motivation is based on the fact that the software works and it is useful. While I agree that the type of communication is more efficient face to face, I acknowledge that I am not always available and so I put up with written documentation. Under a policy of transparency, I also put my resume attached in order to learn who wants my identity and my professional skills (which leaves several ways to contact me). His reading is optional.

Also, I'm open to suggestions for improving the software. If there are bugs, so I can correct them, I must have accurate knowledge. From my experience, to improve a system requires that users can contact the author for improvement of the platform is through positive feedback loops at the initiative of users. This will return to the basic architecture of the next version (which will contain the existing + corrections).

Finally, if you have a problem of a financial nature relating to your PayPal account, you can contact the customer service of PayPal. (https://www.paypal.com/fr/cgi-bin/webscr?cmd=_contact-phone)