

Communicating with Vingd

radomir@vingd.com, Vingd Inc.

March 12, 2013 (v0.1)

1 References

1. A good starting point is developers.vingd.com.
2. Source code for Vingd API Libraries (Python/PHP/C#) available on github.com/vingd:
 - [vingd-api-python](#), [vingd-api-php](#), [vingd-api-csharp](#).
3. Comprehensive documentation (reference manuals) available for python and php.
4. This document is here.

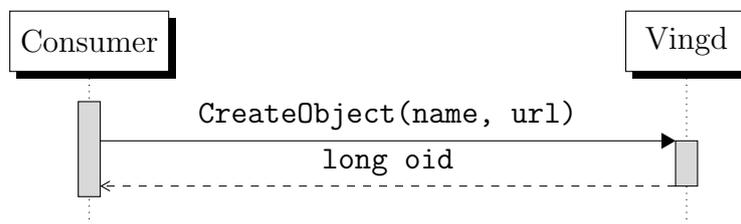
2 Purchase

The content or service of yours has an abstract representation on Vingd, called **Vingd Object**. Object enrollment (aka registration) is a one-time process, after which an object can be "sold" to Vingd users. Terms of purchase (namely, the price) are defined by you, the seller, in what is called *Purchase Order*. Orders can be viewed as bills you give to users. User pays that bill on *Vingd Frontend* and it gets object access rights assigned for the object you referenced in the order. Each time a user tries to access your object (the location you specified upon object enrollment and which you control), you should demand an access token from your user, issued by Vingd Broker. With the access token we shall vouch that the specific user has paid for the access, and you should allow him to view the object (content/service).

2.1 One-time Initialization: Registering an Object

In order to sell anything thru Vingd, you must enroll (i.e. register) an abstract representation of that item, as an object in our *Object Registry* (thru Vingd Broker). Objects are enrolled with `Vingd::CreateObject()` function. During object enrollment, a unique *Object ID* (`oid`) is assigned to that object. Store that `oid`, since you'll be using it as the object handle.

Object description is a dictionary (associative array) with only two mandatory keys: `name` and `url`. Make sure the `url` points to a valid location, because users will be redirected there upon object purchase (consider this URL to represent the object location, according to the REST paradigm). More precisely, the complete URL where user will be redirected is object's `url` with *Token ID* and *Object ID* glued as GET parameters (`tid` and `oid`).

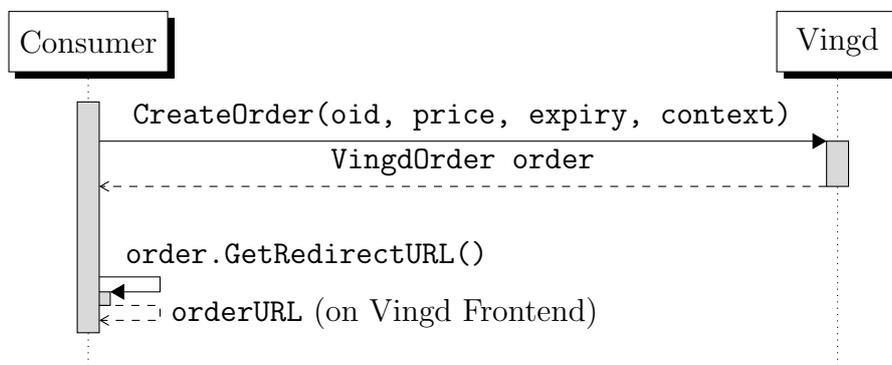


2.2 Selling an Object

To sell any of your enrolled objects, you must create an *Object Order* which shall encapsulate the terms under which the object is being sold to the customer (e.g. price). `Vingd::CreateOrder()` method facilitates order creation. Object and price are defined with `oid` and `price` in `vingd` (rounded to cents). Expiry date of the very order is defined with `expiry`.

It is important to note that orders are not bound to a specific user. That means you could (and preferably would) generate one order with expiry time longer than the default 15mins and offer that same order to more than one user (e.g. you could have special order for each "class" of your users). The advantage of having pre-generated orders is that you are cutting down the overhead of object purchase on your site - you don't have to generate a new order in background for every user's "buy-click", but simply direct the user to the order-purchase-link on Vingd (as returned upon the initial order making).

On a successful order creation, you shall be given an `VingdOrder` object. Call `GetRedirectURL()` to get an URL pointing to your order on Vingd Frontend. To enable an user to pay the access to your object, you should direct her exactly to that very URL (if you're using *Vingd Popup Library* you should use the URL returned with `GetPopupURL()`).

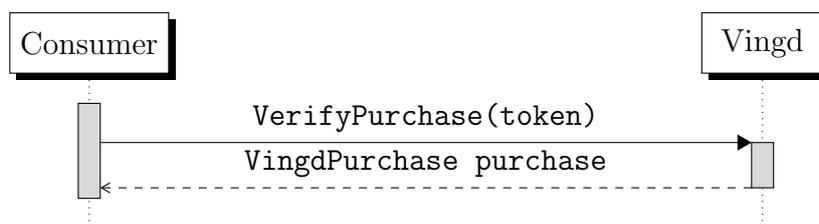


2.3 Verifying user access rights

Upon successful confirmation of an order (specified amount of vingd is reserved from user's account), a *Vingd Token* is issued by Vingd Broker. User is redirected to the (callback) *object url* with the Token ID (`tid`) and Object ID (`oid`) appended. This token is short-lived (typically 15 minutes) and during its lifespan it represents a handle that can be checked to see if the carrier of the (`tid`, `oid`) pair has rights for accessing object `oid`. For convenience, `tid` and `oid` are packed inside a JSON dictionary and returned as `token` GET parameter.

To verify the token, use `Vingd::VerifyPurchase()`. If user access is granted, token verification interface returns *Token Description*. The Token description will also hold the `context` stored during order creation.

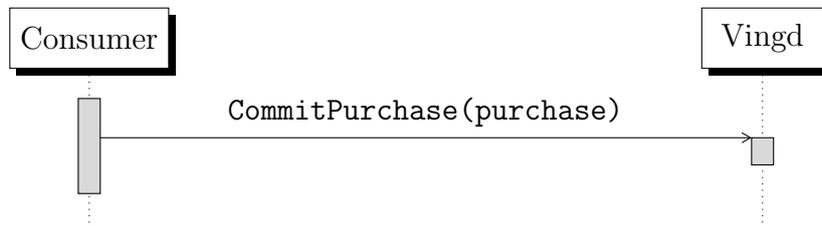
In a popup mode, you will also want to verify the token when user returns from Vingd Frontend (see Vingd Popup onSuccess() callback in Popup library docs).



2.4 Wrapping up: Committing the purchase

Once you have successfully served the content/service user has paid for, you should notify the Vingd Broker, referencing the *Order ID*, *Purchase ID* and *Transfer ID* (all of these are contained in the *Token Description* received during purchase verification). This completes the purchase.

If you don't do `Vingd::CommitPurchase()`, Vingd Broker assumes a seller (you) failed to deliver the content or service to the user and does an automatic refund.



2.5 Complete flow

