Developer resources > Getting started > Authentication > OAuth

Admin API for Embedded App

- API key `{api_key}`
- Password `{password}`
- URL {api_url}
  - e.g. `https://{api_key}:{password}@{hostname}/admin/api/{version}/{resource}.json`
- Shared Secret `{secret}`

Verify Integrity of Request (Query String) from Shopify

Developer resources > Getting started > Authentication > OAuth > Verification

example from How to Generate a Shopify Access Token | Shopify Partners

`http://localhost/generate_token.php?code=6a94694acf0339e9eb8068d8f4718eea&hmac=710205c27e7f780b0cd7ee58146388094be1b9e4762e3752840d1de21deeac5d&shop=johns-apparel.myshopify.com&timestamp=1516399193`

NOTE Query String is everything after `?`

1. Remove the HMAC query parameter (`&hmac={hmac}`) from the query string
2. Process through an HMAC-SHA256 hash function

```
import Crypto.Hash.Algorithms (SHA256)
import Crypto.MAC.HMAC (hmac, hmacGetDigest)
import Data.ByteArray.Encoding (convertToBase, Base(Base64))

-- Data.ByteArray ( instance ByteArrayAccess String )
-- Data.ByteArray ( instance ByteArrayAccess ByteString )
hmac :: (ByteArrayAccess key, ByteArrayAccess message, HashAlgorithm a)
     => key     -- ^ Secret key
     -> message -- ^ Message to MAC
     -> HMAC a

-- Crypto.Hash ( instance ByteArrayAccess (Digest a) )
hmacGetDigest :: HMAC a -> Digest a

-- Data.ByteArray ( instance ByteArray ByteString )
convertToBase :: (ByteArrayAccess bin, ByteArray bout) => Base -> bin -> bout

-- | Example
secret :: ByteString
  = "..."
message :: ByteString
```

```
  = "..."
hmacStr :: ByteString
  = "..."

digested :: ByteString
  = convertToBase Base64 (hmacGetDigest (hmac secret message) :: Digest SHA256)

validity :: Bool
  = hmacStr == digested
```

Step 1: Add App

example: Product Reviews

1. Add App `https://productreviews.shopifyapps.com/login?shop=wynntest.myshopify.com` Shopify redirects User to `app-server/add-app` previously whitelisted endpoint
2. App Server generate unique identifier `{nonce}` Prepare redirect url with required query parameters filled in `https://wynntest.myshopify.com/admin/oauth/request_grant?client_id=60fca9c7f3400ddd43004e94b1355691&redirect_uri=https%3A%2F%2Fproductreviews.shopifyapps.com%2Fauth%2Fshopify%2Fcallback&scope=read_orders%2Cwrite_products%2Cwrite_script_tags%2Cwrite_themes&state=c3dc925c23b77c6999541222135a5c12407eedb625aef614` respond with Header

- status code `201`
- `Location` = url

3. Install `https://productreviews.shopifyapps.com/?hmac=afe96b14ccfbaab2ffe27a4fd5c6c17f11ee6d371311b251976c1a61faf362d1&shop=wynntest.myshopify.com&timestamp=1559669232` Shopify redirects User to `app-server/install-app` previously whitelisted endpoint
4. App Server fetch OAuth `{access_token}` rediret User to home page of embeded app

`https://wynntest.myshopify.com/admin/apps/product-reviews/?hmac=53733f438def68727ad4faf4432d35fee50e7fb1a12421f2e9ddd5124449c17b&locale=en-US&protocol=https%3A%2F%2F&shop=wynntest.myshopify.com&timestamp=1559669294`

Step 2: Ask for permission

`https://{shop}.myshopify.com/admin/oauth/authorize?client_id={api_key}&scope={scopes}&redirect_uri={redirect_uri}&state={nonce}&grant_options[]={access_mode}`

- `{shop} :: String`
  - The name of the user's shop.
- `{api_key} :: String`
  - The app's API Key.
- `{scopes} :: [Scope]`
  - A comma-separated list of scopes. For example, to write orders and read customers, use `scope=write_orders,read_customers`. Any permission to write a resource includes the permission

    to read it.

- `{redirect_uri} :: String`
    - The URL to which a user is redirected after authorizing the client. The complete URL specified here must be added to your app as a whitelisted redirection URL, as defined in the Partner Dashboard.
    - NOTE In older apps, this parameter was optional and redirected to the application callback URL, defined in the Partner Dashboard, when no other value was specified.
- `{nonce} :: String`
    - A randomly selected value provided by your app that is unique for each authorization request. During the OAuth callback, your app must check that this value matches the one you provided during authorization. This mechanism is important for the security of your app.
- `{access_mode} :: Maybe String`
    - `null | "per-user"`
        - `null` offline access mode
        - `"per-user"` online access mode
    - Sets access mode. Defaults to offline access mode if left blank or omitted. Set to per-user for online access mode.

example: Shopify Flow `https://wynntest.myshopify.com/admin/oauth/request_grant?client_id=15100ebca4d221b650a7671125cd1444&redirect_uri=https%3A%2F%2Fflow.shopifycloud.com%2Fauth%2Fshopify%2Fcallback&scope=write_orders%2Cwrite_customers%2Cwrite_products%2Cread_locations%2Cread_notifications%2Cread_shipping%2Cwrite_draft_orders%2Cread_fulfillments%2Cread_gift_cards%2Cwrite_admin_notifications%2Cwrite_channels%2Cread_users%2Cwrite_inventory%2Cread_all_orders%2Cread_apps&state=65d2d3ee0dfa2b946db1cd7e5ee26640c09e24c707fb2d20`

- base url `https://wynntest.myshopify.com/admin/oauth/request_grant`
    - `{shop}` = `wynntest`
    - NOTE `authorize` <- `request_grant` (old version?)
- query params
    - `{api_key}` = `15100ebca4d221b650a7671125cd1444`
    - `{scopes}` = `write_orders,write_customers,write_products,read_locations,read_notifications,read_shipping,write_draft_orders,read_fulfillments,read_gift_cards,write_admin_notifications,write_channels,read_users,write_inventory,read_all_orders,read_apps`
    - `{redirect_uri}` = `https://flow.shopifycloud.com/auth/shopify/callback`
    - `{nonce}` = `65d2d3ee0dfa2b946db1cd7e5ee26640c09e24c707fb2d20`

Step 3: Confirm installation

3.1 Request from Shopify when Install is clicked

`{redirect_uri}?code={authorization_code}&hmac={hmac}&timestamp={timestamp}&state={nonce}&shop={hostname}`

- `{redirect_uri} :: String`
    - from Developer in the last step

- `{authorization_code} :: String`
    - from Shopify
    - e.g. `6a94694acf0339e9eb8068d8f4718eea`
    - The `code` parameter is your authorization code that you will use for the part of the OAuth process.
- `{hmac} :: String`
    - from Shopify
    - e.g. `710205c27e7f780b0cd7ee58146388094be1b9e4762e3752840d1de21deeac5d`
    - The `hmac` is valid. The HMAC is signed by Shopify as explained below, in Verification.
- `{timestamp} :: Int`
    - from Shopify
    - e.g. `1516399193`
- `{nonce} :: String`
    - from Developer in the last step
    - The `nonce` is the same one that your app provided to Shopify during step two.
- `{hostname} :: String`
    - e.g. `johns-apparel.myshopify.com`
    - The `hostname` parameter is a valid hostname
        - NOTE ends with `myshopify.com`
        - does not contain characters other than
            - letters (`a`-`z`)
            - numbers (`0`-`9`)
            - dots
            - hyphens

3.2 Fetch Access Token from Shopify OAuth Endpoint

`POST https://{shop}.myshopify.com/admin/oauth/access_token`

- `{shop}`
    - The name of the user's shop.
- query body :: JSON
    - `client_id` = `{api_key}`
        - The API key for the app, as defined in the Partner Dashboard.
    - `client_secret` = `{secret}`
        - The API secret key for the app, as defined in the Partner Dashboard.
    - `code` = `{authorization_code}`
        - The authorization code provided in the redirect.
- response body :: JSON
    - offline mode
        - `access_token` = `{access_token}`
            - e.g. `f85632530bf277ec9ac6f649fc327f17`

      - An API access token that can be used to access the shop's data as long as the client is installed. Clients should store the token somewhere to make authenticated requests for a shop's data.
    - `scope` = `{scopes}`
      - e.g. `write_orders,read_customers`
      - The list of access scopes that were granted to the application and are associated with the access token. Due to the nature of OAuth, it's always possible for a merchant to change the requested scope in the URL during the authorize phase, so the application should ensure that all required scopes are granted before using the access token. If you requested both the read and write access scopes for a resource, then check only for the write access scope. The read access scope is omitted because it's implied by the write access scope. For example, if your request included `scope=read_orders,write_orders`, then check only for the `write_orders` scope.
  - online mode
    - `access_token` = `{access_token}`
    - `scope` = `{scopes}`
    - `expires_in`
      - e.g. `86399`
      - The number of seconds until the access token expires.
    - `associated_user_scope`
      - e.g. `write_orders`
      - The list of access scopes that were granted to the app and are available for this access token, given the user's permissions.
    - `associated_user`
      - type

        ```
        data AssociatedUser = AssociatedUser
          { id :: Word32
          , first_name :: ByteString
          , last_name :: ByteString
          , email :: ByteString
          , email_verified :: Boolean
          , account_owner :: Boolean
          , locale :: ByteString
          , collaborator :: Boolean
          }
        ```

      - Information about the user who completed the OAuth authorization flow.
      - e.g.

```json
{
  "id": 902541635,
  "first_name": "John",
  "last_name": "Smith",
  "email": "john@example.com",
  "email_verified": true,
  "account_owner": true,
  "locale": "en",
  "collaborator": false
}
```

Step 4: Making authenticated requests