OVERVIEW

The ShrinkTheWeb API is designed such that anyone can code a hook into our system remotely. It is possible to code for making REST calls, .NET calls, AJAX, etc. Currently, however, ShrinkTheWeb supplies and maintains only PHP4 & PHP5 sample code.

Other code may available through the community with development and sharing strongly encouraged. As time permits, we will add to our library of supported code. If you have code to contribute, please let us know and we will give appropriate credit in the community for your efforts!

STW API RESPONSE CODES

Advanced Method

<table>
<thead>
<tr>
<th>CASES</th>
<th>EXISTS</th>
<th>VERIFIED</th>
<th>ACTION</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thumbnail New</td>
<td>false</td>
<td>false</td>
<td>noexist</td>
<td>Successfully Queued. Check back later.</td>
</tr>
<tr>
<td>New Request</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thumbnail Queued</td>
<td>false</td>
<td>false</td>
<td>queued</td>
<td>Still in Queue. Check back later.</td>
</tr>
<tr>
<td>Pending</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thumbnail Captured</td>
<td>true</td>
<td>false</td>
<td>delivered</td>
<td>Captured but not verified.</td>
</tr>
<tr>
<td>Captured</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thumbnail Verified</td>
<td>true</td>
<td>true</td>
<td>delivered</td>
<td>Status verified.</td>
</tr>
<tr>
<td>Finalized</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error External</td>
<td>false</td>
<td>true</td>
<td>noretry</td>
<td>No need to try again.</td>
</tr>
<tr>
<td>Bad Request</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General Error</td>
<td>false</td>
<td>false</td>
<td>noretry</td>
<td>Change your request before trying again.</td>
</tr>
<tr>
<td>STW Offline</td>
<td>false</td>
<td>false</td>
<td>offline</td>
<td>STW temporarily offline for maintenance.</td>
</tr>
<tr>
<td>ANY Access Denied</td>
<td>false</td>
<td>false</td>
<td>fix_and_retry</td>
<td>Change your request before trying again.</td>
</tr>
<tr>
<td>Refresh Queued</td>
<td>false</td>
<td>false</td>
<td>queued</td>
<td>Successfully Queued. Check back later.</td>
</tr>
<tr>
<td>Exists in STW cache</td>
<td>true</td>
<td>false</td>
<td>delivered</td>
<td>Existing capture exists &amp; delivered.</td>
</tr>
</tbody>
</table>

Basically, if the thumbnail exists and we delivered, you'll get "delivered"
If the PagePix service is temporarily offline for maintenance, the thumbnail action will be "offline"
If there is a problem where the domain/request is broken, the thumbnail action will be "noretry"
If there is a problem with your request, the thumbnail action will be “fix_and_retry”
If there is a problem with our capture, the thumbnail action will be "fix_and_retry" and no charge for refreshing
Example XML Response (ready to download)

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <stw:Response>
    <stw:ThumbnailResult>
      <stw:Thumbnail Exists="true">http://images.shrinktheweb.com/xian.php?stwaccesskeyid=&amp;stwu=&amp;subref=&amp;stwq=95&amp;stwxmax=320&amp;hash=</stw:Thumbnail>
       <stw:Thumbnail Verified="true">delivered</stw:Thumbnail>
    </stw:ThumbnailResult>
    <stw:ResponseStatus>
      <stw:StatusCode>Success</stw:StatusCode>
    </stw:ResponseStatus>
    <stw:ResponseTimestamp>
      <stw:StatusCode>1426457959</stw:StatusCode>
    </stw:ResponseTimestamp>
    <stw:ResponseCode>
    </stw:ResponseCode>
    <stw:CategoryCode>
      <stw:StatusCode></stw:StatusCode>
    </stw:CategoryCode>
    <stw:Quota_Remaining>
      <stw:StatusCode>162</stw:StatusCode>
    </stw:Quota_Remaining>
    <stw:Bandwidth_Remaining>
      <stw:StatusCode>999673</stw:StatusCode>
    </stw:Bandwidth_Remaining>
  </stw:Response>
</stw:ThumbnailResponse>
```

Example XML Response (successfully queued for capture)

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <stw:Response>
    <stw:ThumbnailResult>
      <stw:Thumbnail Verified="false">noexist</stw:Thumbnail>
    </stw:ThumbnailResult>
    <stw:ResponseStatus>
      <stw:StatusCode>Success</stw:StatusCode>
    </stw:ResponseStatus>
    <stw:ResponseTimestamp>
      <stw:StatusCode></stw:StatusCode>
    </stw:ResponseTimestamp>
    <stw:ResponseCode>
      <stw:StatusCode></stw:StatusCode>
    </stw:ResponseCode>
    <stw:CategoryCode>
      <stw:StatusCode></stw:StatusCode>
    </stw:CategoryCode>
  </stw:Response>
</stw:ThumbnailResponse>
```
<stw:Quota_Remaining>
  <stw:StatusCode>162</stw:StatusCode>
</stw:Quota_Remaining>

<stw:Bandwidth_Remaining>
  <stw:StatusCode>999672</stw:StatusCode>
</stw:Bandwidth_Remaining>

Example XML Response (invalid request - BLANK_DETECTED)

<?xml version="1.0" encoding="UTF-8"?>
  <stw:Response>
    <stw:ThumbnailResult>
      <stw:Thumbnail Verified="false">noretry</stw:Thumbnail>
    </stw:ThumbnailResult>
    <stw:ResponseStatus>
      <stw:StatusCode>BLANK_DETECTED</stw:StatusCode>
    </stw:ResponseStatus>
    <stw:ResponseTimestamp>
      <stw:StatusCode>1426469587</stw:StatusCode>
    </stw:ResponseTimestamp>
    <stw:ResponseCode>
      <stw:StatusCode>INVALID_REQUEST</stw:StatusCode>
    </stw:ResponseCode>
    <stw:CategoryCode>
      <stw:StatusCode></stw:StatusCode>
    </stw:CategoryCode>
    <stw:Quota_Remaining>
      <stw:StatusCode>161</stw:StatusCode>
    </stw:Quota_Remaining>
    <stw:Bandwidth_Remaining>
      <stw:StatusCode>999671</stw:StatusCode>
    </stw:Bandwidth_Remaining>
  </stw:Response>
</stw:ThumbnailResponse>
Example JSON Response (ready to download)


Example JSON Response (successfully queued for capture)


Example JSON Response (invalid request - BLANK_DETECTED)


Definitions

- **Fetch**: A request that we already have captured and stored in cache (i.e. not yet deleted by retention)
- **New Request**: A request that we must capture and store
- **Retention**: We store website thumbnails on our storage but delete eventually, if not requested again
- **Rolling Refresh**: Our term for the automatic monthly (30-45 days) refresh of popular requests

---

Page 4
### Simple Method

<table>
<thead>
<tr>
<th>REQUEST PARAMETERS</th>
<th>VALUES</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>stwembed</td>
<td>&lt;IMG SRC&gt;</td>
<td>1</td>
</tr>
<tr>
<td>stwaccesskeyid</td>
<td>Credentials</td>
<td>[KeyId]</td>
</tr>
<tr>
<td>stwsizes</td>
<td>75x56</td>
<td>mcr</td>
</tr>
<tr>
<td></td>
<td>90x68</td>
<td>tiny</td>
</tr>
<tr>
<td></td>
<td>100x75</td>
<td>vsm</td>
</tr>
<tr>
<td></td>
<td>120x90</td>
<td>sm</td>
</tr>
<tr>
<td></td>
<td>200x150</td>
<td>lg</td>
</tr>
<tr>
<td></td>
<td>320x240</td>
<td>xlg</td>
</tr>
<tr>
<td>stwadv</td>
<td>{format}</td>
<td>xml</td>
</tr>
<tr>
<td></td>
<td></td>
<td>json</td>
</tr>
<tr>
<td>stwxmax</td>
<td>{number}</td>
<td>[123]</td>
</tr>
<tr>
<td>stwurl</td>
<td>URL</td>
<td>[URL]</td>
</tr>
</tbody>
</table>

*It is not "Best Practices" to use &stwembed=1 for scripts or applications. Use the "Advanced Method" API instead. If you try to cache locally with the "Embedded Method", you must check the headers for "404 Not Ready".*

**Supports with or without “HTTP”; Supports “HTTPS”; Supports with or without “WWW”**


For PRO Features, please refer to the PRO Feature Documentation PDF for the latest usage guidelines.

Free, Basic service allows homepage and subdomain. Inside pages require the "Inside Page" PRO Feature upgrade.

DO NOT send &stwu (Secret Key) with "Simple Method" embedded requests. It will be blocked and return an error.

**Tip:** Paste your request into a browser with stwembed=0 and &stwu (secret key) in order to get raw XML response. Pass &stwadv=json, BEFORE &stwurl, in order to receive a JSON response instead of the XML default.

*Note: You must have your IP on your allowed referrers list, but it will be added automatically with this type of request.*