CaptureHandler takes screenshots

CaptureHandler takes screenshots of pages using either Chrome or PhantomJS.

- Chrome
- PhantomJS
- Screenshot service
- Encode URLs
- Screenshot library
- Client-side capture

Chrome

Chrome is the recommended engine from v1.23.
Add this to `gramex.yaml`:


```yaml
url:
  capture:
    pattern: /$YAMLURL/capture
    handler: CaptureHandler
    kwargs:
      engine: chrome
      pattern: ^http
```

When Gramex runs, it starts `node chromecapture.js --port 9900` running a node.js based web application (chromecapture.js) at port 9900.

v1.94 The `pattern: ^http` only allows URLs that start with `http`, disallowing `file://` and other such URLs. (Relative URLs like `../` are converted to absolute HTTP URLs before checking the pattern, so they will work fine.)

To only allow specific domains, e.g. `gramener.com` and `gramener.co`, use:

```
pattern: ^https?://(www\.)?(gramener\.com|gramener\.co)/
```

To change the port, use:

```
pattern: /$YAMLURL/capture
handler: CaptureHandler
kwargs:
  engine: chrome
  pattern: ^http
  port: 9901 # Use a different port
```

To use an existing instance of chromecapture.js running on a different port, use:

```
pattern: /$YAMLURL/capture
handler: CaptureHandler
kwargs:
  engine: chrome
  pattern: ^http
  url: http://server:port/capture/ # Use chromecapture.js from this URL
```

The default viewport size is 1200x768. To set a custom viewport for images or PPTX, use `?width=` and `?height=`. For example, `?width=1920&height=1080` changes the viewport to 1920x1080.

By default, requests timeout within 10 seconds. To change this, use `timeout:`.

```
pattern: /$YAMLURL/capture
handler: CaptureHandler
kwargs:
  pattern: ^http
  timeout: 20 # Wait for max 20 seconds for server to respond
```

The default chromecapture.js is at `$GRAMEXPATH/apps/capture/chromecapture.js`. To use your own chromecapture.js, run it using `cmd:` on any port and point `url:` to that port:

```
pattern: /$YAMLURL/capture
handler: CaptureHandler
kwargs:
  engine: chrome
  cmd: node /path/to/chromecapture.js --port=9902
  url: http://localhost:9902/
```

To use a HTTP proxy, set the `ALL_PROXY` environment variable. If your proxy IP is `10.20.30.40` on port `8000`, then set `ALL_PROXY` to `10.20.30.40:8000`. See how to set environment variables. (You can also use the `HTTPS_PROXY` or `HTTP_PROXY` environment variables. These supercede `ALL_PROXY`.)

NOTE: If you're running CaptureHandler with Chrome on a Docker instance (or any other headless Linux), you may get an `error while loading shared libraries`.

This is because Chrome needs additional dependencies.

On Ubuntu 20.04, you can run this command to fix it:

```
sudo apt-get -y install xvfb libnss3 libatk1.0-0 libatk-bridge2.0-0
libxcomposite1 libcups2 libxrandr2 libpangocairo-1.0-0 libgtk-3-0
```

For other systems, check this issue.

PhantomJS

PhantomJS is **deprecated** but is the default for backward compatibility. To use it, install PhantomJS and it to your PATH. Then add this to `gramex.yaml`:

```yaml
url:
  capture:
    engine: phantomjs # Optional.
    pattern: /$YAMLURL/capture
    handler: CaptureHandler
```

Screenshot service

You can add a link from any page to the `capture` page to take a screenshot.


```
<a href="capture?ext=pdf">PDF screenshot</a>
<a href="capture?ext=png">PNG screenshot</a>
<a href="capture?ext=jpg">JPG screenshot</a>
<a href="capture?ext=pptx">PPTX screenshot</a>
```

Try it here:

- PDF screenshot
- PNG screenshot
- JPEG screenshot
- PPTX screenshot

It accepts the following arguments:

- `?url=`: URL to take a screenshot of. This defaults to `Referer` header. So if you link to a `capture` page, the source page is generally used.
  Example: ?url=https://example.org/
- `?file=`: screenshot file name. Default: `screenshot`.
  Example: ?file=newfile
- `?ext=`: format of output. Can be pdf, png, jpg or pptx. Default: `pdf`.
  Example: ?ext=png. (`ext=pptx` available only in `engine: chrome` from **v1.23.1**)
- `?domain=`: set cookies on specified domain. Use this if sending CORS (cross-origin requests) and CaptureHandler must set cookie domains.
  Example: ?domain=.example.org passes cookies to *.example.org, over and above the `url`
- `?delay=`: wait for before taking a screenshot.
  - If this is a number, waits for this many milliseconds.
    Example: ?delay=1000 captures this timer page with a ~1000 ms delay
  - If `?delay=renderComplete`, waits until the JavaScript variable `window.renderComplete` is set to true - or 30 seconds.
  - If the delay is more than the `timeout:` in the `kwargs:` section, the page will time out.
- For PDF:

    - `?format=`: A3, A4, A5, Legal, Letter or Tabloid. Default: A4.
      Example: ?format=Tabloid
    - `?orientation=`: portrait or landscape. Default: portrait.
      Example: ?orientation=landscape
    - `media=`: `print` or `screen`. Default: `screen`. (Only in `engine: chrome`)
      Example: ?media=print
    - `header=`: a pipe-separated string that sets the page header. You can use `$pageNumber`, `$totalPages`, `$date`, `$title`, `$url` as variables.
      Example: ?header=Gramener: Left header “Gramener”
      Example: ?header=|$title|: Center header with page title
      Example: ?header=|$pageNumber: Right header with page number
      Example: ?header=©|Gramener|$pageNumber/$totalPages: Left, middle right headers.
    - `footer=`: similar to `header`
    - `headerTemplate=`: HTML template to add a custom header. Template cannot load external sources or run javascript, but can use inline css styles. See docs. Ensure that enough margin is provided for the header.
      Example: `?headerTemplate=<div style="border-bottom:1px solid black;display:flex;justify-content:space-between;width:100%"><span class="url"></span><span class="date"></span></div>`
    - `footerTemplate=`: similar to `headerTemplate`
    - `margins=`: comma-separated list of margins specifying top, right, bottom, left margins respectively. default margin is `1cm,1cm,1cm,1cm`.
      Example ?margins=2cm,,2cm, sets top and bottom margin to 2cm
- For images (PNG/JPG):
    - `?width=`: optional viewport width in pixels. Default: 1200
      Example: ?width=600
    - `?height=`: optional viewport height in pixels. Default: auto (full page)
      Example: ?height=600

    - `?scale=`: zooms the screen by a factor. scale=2 returns an image twice as large and sharp as scale=1. Default: 1.
      Example: ?scale=0.2 compared with ?scale=1
    - `?selector=`: Restrict screenshot to (optional) CSS selector in URL. Captures the entire element, even if it exceeds the viewport
      Example: ?selector=.content excludes the sidebar.
    - `?emulate=`: emulate full page on a device. Ignores `?width=`, `?height=` and `?scale=`. (Only in `engine: chrome` from **v1.56.0**)
      Example: ?emulate=iPhone 6. Device names can be iPhone 8, Nexus 10, Galaxy S5, etc.
- For PPTX (Only in `engine: chrome` from **v1.23.1**):
    - `?layout=`: A3, A4, Letter, 16x9, 16x10, 4x3. Default: `4x3`
      Example: ?layout=16x9
    - `?dpi=`: optional image resolution (dots per inch). Default: 96
      Example: ?dpi=192
    - `?width=`: optional viewport width in pixels. (Default: 1200px)
      Example: ?width=600&height=400
    - `?height=`: optional height to clip output to. Leave it blank for full page height
      Example: ?width=1200&height=900
    - `?selector=`: CSS selector to take a screenshot of
      Example: ?selector=.codehilite
    - `?title=`: optional slide title
      Example: ?title=First+example&selector=.codehilite
    - `?title_size=`: optional title font size in points. Defaults to 18pt
      Example: ?title=First+example&title_size=24&selector=.codehilite
    - `?x=`: optional x-position (left margin) in px. Centers by default
      Example: ?x=10&selector=.codehilite

    - `?y=`: optional y-position (leftop margin) in px. Centers by default
      Example: ?y=200&selector=.codehilite
    - For multiple slides, repeat `?selector=`, optionally with `?title=`, `?title_size=`, `?x=`, `?y=`, `?dpi=`.
      Example: ?selector=.toc&title=TOC&selector=.codehilite&title=Example
- `?debug=`: displays request / response log requests on the console.
    - `?debug=1` logs all responses and HTTP codes. It also logs browser console.log messages on the Gramex console
    - `?debug=2` additionally logs all requests

If the response HTTP status code is 200, the response is the screenshot. If the status code is 4xx or 5xx, the response text has the error message.

Authentication is implicit. The cookies passed to `capture` are passed to the `?url=` parameter. This is exactly as-if the user clicking the capture link were visiting the target page.

To try this, log in and then take a screenshot. The screenshot will show the same authentication information as you see below.


```
{
    "_t": 1734956763.4942372,
    "id": "+YTlPxGOk/gLL3ulhJyixqDiaMsmz6y9",
    "randkey": 597
}
```


You can override the user by explicitly passing a cookie string using `?cookie=`.

All HTTP headers are passed through by default. CaptureHandler sends them to Chrome (not PhantomJS), which passes it on to the target URL.

If `capture.js` was not started, or it terminated, you can restart it by adding `?start` to the URL. It is safe to add `?start` even if the server is running. It restarts `capture.js` only if required.

Encode URLs

When constructing the `?url=`, `?selector=`, `?title=` or any other parameter, ensure that the URL is encoded. So a selector `#item` does not become `?id=#item` – which makes it a URL hash – and instead becomes `?id=%23item`.

Use `urlencode` to encode URLs in templates or in Python:

```
from urllib.parse import urlencode

query = urlencode({'url': ..., 'selector': [..., ...]}, doseq=True)
```

Use `URLSearchParams` to encode URLs in JavaScript:

```
const query = (new URLSearchParams({url: '...', selector:
'...'})).toString()
const query = (new URLSearchParams(['url', '...'], ['selector', '...'],
['selector', '...'])).toString()

// Set a link HREF based on the query
document.querySelector('a.capture').setAttribute('href', `capture?
${query}`)
// Or add an event listener based on the query
```

```
document.querySelector('button.capture').on('click', function() {
    location.href = `capture?${query}`
})
```

Screenshot library

You can take screenshots from any Python program, using Gramex as a library.


```
from gramex.handlers import Capture          # Import the capture library
capture = Capture(engine='chrome')           # Run chromecapture.js at
port 9900

url = 'https://gramener.com/'                # Page to take a screenshot
of
with open('screenshot.pdf', 'wb') as f:      # Save screenshot as PDF
    f.write(capture.pdf(url, orientation='landscape'))
with open('screenshot.png', 'wb') as f:      # Save screenshot as PNG
    f.write(capture.png(url, width=1200, height=600, scale=0.8))
```

The Capture class has convenience methods called `.pdf()`, `.png()`, `.jpg()` that accept the same parameters as the handler.

Client-side capture

CaptureHandler reloads a page to take a screenshot. This can be slow. To avoid this, you can:

- Cache page results for longer

- OR: use client-side capturing using [html2canvas] and downloading via [FileSaver.js]

Add the libraries from the [UI component library]:

```
<script src="ui/html2canvas/dist/html2canvas.min.js"></script>
<script src="ui/file-saver/dist/FileSaver.min.js"></script>
```

Trigger the download as follows:

```
html2canvas(document.querySelector(".chart")) // Pick the element to download
  .then((canvas) => {
    canvas.toBlob((blob) => {
      saveAs(blob, "chart.png"); // Pick your filename
    });
  });
```

WARNING: This requires inline styles. Styles from classes (e.g. Bootstrap's `border`) are not applied. Add styles inline, via `style="..."`.

Source ↗

Client-side capture example

Gramener

Gramener is a design-led data science company that helps solve complex business problems with compelling data stories using insights and a low-code analytics platform.

Gramex on Social Media

Quick links

About us

News

Videos

Blog

Careers

Cookie Policy

Privacy Policy

Products & Offerings

Gramex Low-code Platform

SlideSense

Clusters

Data Explorer

ComicGen

Solutions

Narrative

Data Science Advisory

Contact Us

Request Demo

+1 609 454 3669

reachus@gramener.com

[redacted]