ReproZip Web Documentation

Release 0.1

Brian Hoffman, Remi Rampin, Fernando Chirigati, and Vicky Steev

Mar 20, 2019

Contents

Welcome to ReproZip Web's documentation! This tool is a prototype that leverages ReproZip and Webrecorder to archive web applications and allows users to replay these apps with little to no effort.

CHAPTER 1

Installation Instructions

Python 2.7.3 or greater, or 3.3 or greater is required. If you don't have Python on your machine, you can get it from python.org. You will also need the pip installer and Docker.

For Debian and Ubuntu, you can get most of the required dependencies using APT:

```
apt-get install python python-dev python-pip
```

For Fedora and CentOS, you can get most of the dependencies using the Yum packaging manager:

```
yum install python python-devel
```

For macOS, be sure to upgrade *setuptools*:

```
$ pip install -U setuptools
```

After installing these required dependencies, clone the repository and cd into it:

```
$ git clone https://github.com/reprozip-news-apps/reprozip-web
$ cd reprozip-web
```

Now install all the dependencies and the prototype:

```
$ pip install -r requirements.txt
$ pip install -e .
```

CHAPTER 2

Archiving and Replaying a Web App

2.1 Step 1: Package a Web site using ReproZip

Skip to step 2 if you already have an `.rpz` package. Otherwise, follow the ReproZip's documentation to package a web app using ReproZip.

2.2 Step 2: Record the Web site assets from the package using Webrecorder

Make sure that you have Docker installed and running. Given an *.rpz* package from a web app, you can run the following command:

```
reprounzip dj record <package> <target> --port <port>
```

where `<package>` is the `.rpz` file, `<target>` is the target directory for ReproZip, and `<port>` is the port number where the web app run. For instance, a Rails app will likely run on port `3000`, while a NodeJS app will likely run on port `8000`.

Note that, while recording, the Chromium Web browser will be used to open the web app. When the recording is done, Chromium will automatically close.

You should be able to see the `WARC_DATA` directory in the package now:

```
$ tar -t -f <package>
-rw-------  0 root   root 729415801 Mar  9  2017 DATA.tar.gz
-rw-------  0 root   root        19 Mar  9  2017 METADATA/version
-rw-r--r--  0 root   root   5912576 Mar  9  2017 METADATA/trace.sqlite3
-rw-------  0 root   root    293142 Mar  9  2017 METADATA/config.yml
-rw-r--r--  0 hoffman staff   807498 Jan 11 09:16 WARC_DATA/rec-20190111141622981410-
↪anything.local.warc.gz
-rw-r--r--  0 hoffman staff    37089 Jan 11 09:16 WARC_DATA/autoindex.cdxj
```

The following flags can also be used when running the `reprounzip dj record` application:

- `--quiet`: hides terminal messages.
- `--keep-browser`: keeps the Web browser open for manual recording.
- `--skip-record`: writes `WARC` data from `<target>` directory without recording the web app again.
- `--skip-setup`: skips the `reprounzip setup` step. This option can only be used if the web app was already unpacked by ReproZip.
- `--skip-run`: skips the `reprounzip run` step. This option can only be used if the web app was already unpacked by ReproZip.
- `--skip-destroy`: does not destroy the Docker container and `<target>` directory after recording the web app.

2.3 Step 3: Replay the web app

To replay the package web app, run the following command:

```
$ reprounzip dj playback <package> <target> --port <port>
```

The Chromium Web browser will automatically open, and you can turn off your Wi-Fi and hit reload to explore the web app. Press Enter in your terminal session to shut everything down.

The following flags can also be used when running the `reprounzip dj playback` application:

- `--quiet`: hides terminal messages.
- `--standalone`: runs the archived web app as a wayback collection you can share over the web. Does not launch a browser.
- `--hostname`: sets the hostname used by the proxy server and displayed in the browser's location bar.
- `--skip-setup`: skips the `reprounzip setup` step. This option can only be used if the web app was already unpacked by ReproZip.
- `--skip-run`: skips the `reprounzip run` step. This option can only be used if the web app was already unpacked by ReproZip.
- `--skip-destroy`: does not destroy the Docker container and `<target>` directory after replaying the web app.

2.4 Skipping removal of container

When you finish recording, or exit a playback session, the unpacked container will be automatically destroyed. You can prevent this from happening by using the `--skip-destroy` flag:

```
$ reprounzip dj playback <package> <target> --port <port> --skip-destroy
```

Then you can reuse the container on another playback session:

```
$ reprounzip dj playback <package> <target> --port <port> --skip-setup --skip-run
```

2.5 Packing and Recording Simultaneously

You can run `reprozip trace` and `reprounzip dj record` at the same time, using two different terminals: both on the site host, or one on the site host and one on a different host.

Terminal 1:

```
$ cd /path/to/your/project
$ reprozip trace <application>
```

Terminal 2:

```
$ mkdir /path/to/target
$ reprounzip dj live-record <localhost-link> <target>
```

where `<localhost-link>` is the local link to the web app (e.g.: `http://localhost:3000`). Wait for the recorder to finish, then go back to Terminal 1 and press `CTRL-C`.

Terminal 1:

```
$ reprozip pack reprozip-package.rpz
```

The final step is to merge the recorded data into the ReproZip package:

```
$ reprounzip dj record reprozip-package.rpz <target> --skip-record
```