



Overview

Page Size

Client-Side

Server-Side

Backoffice

# GCLC: Green Code Lab Challenge

*Image SlideShow*

▀ Team 27: GreenTSP



Overview

## Team 27: GreenTSP

### ➤ Members:

- *Alexis Mousset*
- *François Monniot*
- *Clément Berthou*
- *Thomas Lecourt*
- *Moroine Bentefrit*

### ➤ Where

- *Telecom SudParis - EVRY*

Page Size

Client-Side

Server-Side

Backoffice

Overview

Page Size

Client-Side

Server-Side

Backoffice

## ➤ Gather files

- Gather javascript files to limit the number of files served
- Interest: Each request introduces latency due to response time
- Use sprites to gather images files

## ➤ Use cache

- Enable cache user and proxy on static resources
- Use cache-control to keep website up-to-date
- Use public cache to enable proxy cache

## ➤ Avoid the use of unnecessary libraries

- Remove jquery library save up-to 242 KB.
- Remove unused style rules in bootstrap

Overview

Page Size

Client-Side

Server-Side

Backoffice

## ➤ Resources' minimization

- Interest: Reduce page size by removing unnecessary or redundant data without affecting how the resource is processed by the browser
- Content-specific optimizations can significantly reduce the size of delivered resources.
- How ?
  - Remove unnecessary content like spaces, line-break ...
  - Compress data by renaming variables and functions with smaller text.

## ➤ Enable Gzip compression for text files

- Enabling gzip compression can reduce the size of the transferred response by up to 90%
- Reduce the amount of time to download the resource
- Improve the time of the first render

Overview

Page Size

Client-Side

Server-Side

Backoffice

## ➤ Optimize images

- Remove metadatas
- Preprocess images by resizing and cropping. In order to serve them in the right format (Here we have two available formats, small and large)
- Use utilities: Jpegoptim & Optipng
- Improve the user experience by using progressive images

## ➤ Load Images Asynchronously

- Reduce page rendering
- Load images only if it's necessary

## ➤ Reduce Twitter widget impact

- Reduce synchronization timer to 30 seconds
- Reduce image size

Overview

Page Size

Client-Side

Server-Side

Backoffice

- CSS Animations rather than Js Animations
  - Use GPU if hardware acceleration is enabled
  - 2 transitions available: fade & translate
- Avoid memory loss
  - Delete unused variables
- Information in Json configuration file
  - Available images
  - Images' descriptions
  - Images' path
  - Images' transition type



Overview

Page Size

Client-Side

Server-Side

Backoffice

- Avoid serving dynamic content

- Serve static index

- Image preprocessing

- Check new, modified and removed images
  - Create the preprocessed image (resize + crop)
  - Update Json information file

- Avoid parsing folders for each request: NodeJs CRON

- Preprocess new, modified and deleted images in gclcimages folder
  - Parse property files in gclcimages folder
  - Create Json configuration file

Overview

Page Size

Client-Side

Server-Side

Backoffice

## ➤ Technologies

### ➤ Remove unnecessary functions/packages

- Mail server (exim)
- No logs (rsyslog disabled)

### ➤ Nginx vs Apache2

- Require less resources (RAM & CPU)
- Event driven server: efficient for static resources
- Configuration with 1 worker, open\_file\_cache enabled and a reduced keepalive timeout to be as efficient as possible

### ➤ NodeJs for preprocessing images

- Efficient



Overview

Page Size

Client-Side

Server-Side

Backoffice

- Backoffice: upload images

- url: <http://92.222.110.219/backoffice/upload>

- Upload images for the slideshow

- Preprocess image and update configuration file (call nodeJs preprocessor)

- FileUpload in php

- Use less memory in waiting state than Node process

- Use php-fpm on ondemand mode to spawn php processes only when needed



# Find our project on GitHub

<https://github.com/greenTSP/greenTSP-slideshow>