



Assignment 9 (NF, minor subject)

Due: Friday 13.01.2017; 14:00h (3 Weeks)

Goals

After doing the exercises,

- You have a better understanding of different watermarking techniques
- You can create watermarks “on the fly” to protect your online content.
- You can create mash-ups and become an Ajax-Superhero/-ine.

Task 1: Watermark Characteristics

Difficulty: Medium

Please characterize the following watermarks regarding visibility, universality, detectability, robustness, capacity, security, and efficiency:

- a) watermark on a banknote
- b) Image rendered on top of another image (what we did in the tutorials)
- c) A Digimarc Watermark
- d) EXIF Information in images
- e) The copy protection of a BluRay disc

Create a table like this with a row for each watermark described in a) to e):

| | Visible? | Universal? | Detectable? | Robust? | High Capacity? | Secure? | Efficient? |
|----------|----------------------------|---------------------------------|-------------|---------|----------------|---------|------------|
| Banknote | Yes, if held against light | Yes, the same for all banknotes | Yes... | | | | |

Submit the solution as “task1.pdf” or “task1.xls”

Task 2: Watermark-Page

Difficulty: Easy

We created a webpage that allows a user to select a file from their computer and add a watermark to it. We already implemented a large part of the functionality, but there are four small TODO items in the code skeleton that you find in the material.

Watch a demo of the functionality here: <https://youtu.be/JsRI6R94dTI>

Complete the four TODOs and submit your code.



Task 3: Mash Up

Difficulty: Hard

This task is time-consuming, but it's a cool project.

A Mash-Up is a web application that fetches content from many different sources, most of the times there is a specific topic.

Create a Mash-Up web-page with a topic of your choosing.

Our idea would be to create a mash-up about music artists.

Possible Features / Use Case Scenario

The user enters the name of an artist. Then the web page fetches...

- images and allows the user to browse through them in a slide-show.
- tracks and displays them as a list with their album-cover image. When they are clicked, the song starts playing a free preview
- a list of events, where the artist is playing. The user can click them and is taken to the event details
- videos about the artist.

You can do all this by looking at tutorials 06 – 09. There are a couple of links on the next page that help you get started.

Submit all your code files. We will create a mash-up, too and update this assignment to include a link to a demo video.



Service Providers / APIs

Here is a small list of APIs that you can use for your mash-up.

- **Spotify.** You should already be familiar with their API, if not, check out the tutorial slides and previous assignments.
Search URL: <https://api.spotify.com/v1/search>
Parameters: q (query), type (artist, track, album, playlist)
API Key required: No
Documentation: <https://developer.spotify.com/web-api/search-item/>
- **Songkick.** Songkick delivers events (tourdates) of specific artists.
Search URL: http://api.songkick.com/api/3.0/artists/{artist_id}/calendar.json (replace artist_id with real artist id)
Parameters: apiKey, all others see documentation
API Key required: Yes (free, but takes several days to receive it.)
Documentation: <http://www.songkick.com/developer/artist-search>,
<http://www.songkick.com/developer/upcoming-events-for-artist>
- **Flickr.** This API delivers images.
Search URL: <https://www.flickr.com/services/api/flickr.photos.search.html>
Parameters:
API Key required: Yes (Free)
Documentation: <https://www.flickr.com/services/api/flickr.photos.search.html>
- **YouTube.** Delivers the URLs of YouTube Videos (can be embedded)
Parameters: q (String, query) part (String, use “snippet”)
API Key required: Yes (Free)
Documentation: <https://developers.google.com/youtube/v3/docs/search/list>
- **Getty Images.** Another image API.
Search URL: <https://api.gettyimages.com/v3/search/images>
Parameters: phrase (String)
Headers: Api-Key (to use headers in jQuery, see here: <http://api.jquery.com/jquery.ajax/>)
API key required: Yes (Free)
Documentation: <http://developers.gettyimages.com/api/docs/v3/search/images/get/>

Useful tutorial links:

- <http://www.programmableweb.com/>
- <https://www.codecademy.com/en/tracks/youtube>



Happy Holidays!

We hope you have a good time during the break. Happy holidays from the MMN team!

Submission

Please turn in your solution via UniWorX as a ZIP archive. You can form groups of up to four people.

We encourage you to sign up for our Slack Team! All you need is a CIP account and an email address that ends in “@cip.ifi.lmu.de”. Ask us if you don’t know how to get them.

If you have questions or comments before the submission, please contact one of the tutors. They are on Slack: [@tobi.seitz](#), [@peterjuras](#) and [@thomas-weber](#). Remember, that they also want to enjoy their weekends ☺

It also makes a lot of sense to ask questions directly in our [#mmn-ws1617](#) channel. Maybe fellow students can offer help or benefit from the answers, too!

CodeLabs

As always, everyone is invited to participate in the CodeLabs on Wednesday between 6 and 8 pm. We encourage you to go there if you struggle with installation or nasty bugs that just don’t seem to go away. The times are flexible, so you can go there sometime during the CodeLab.