



## Assignment 13 (HF, major subject)

---

*Due: no due date – for your own preparation only.*

### Goals

These exercises will

- Help you prepare for the exam
- Reactivate topics we discussed during the semester

Suggestion: Do them on paper to reenact an exam-like scenario.

### Task 1: HbbTV 2

Read this article: [HbbTV 2.0 spec released](#) by Chris Dziadul

Briefly answer the following questions:

- a) What are the most anticipated features of HbbTV 2.0?
- b) Is HbbTV capable of UltraHD? Which codec is mentioned in this context?
- c) Name 2 implications of “synchronization between TV, tablets and smartphones” for the users (not in the article).
- d) Is HbbTV 2.0 downward-compatible with older versions of HbbTV? Why is this important?

### Task 2: PHP

- a) You want to implement a “guess-my-number” game with PHP.  
Extend the code fragment on the next page to meet these requirements:
  - a. Create a variable that holds a secret number of your choosing.
  - b. If the form was submitted, check if the user entered the correct number. If the guess was correct, inform the user and reset the game. Otherwise, inform the user that the guessed number was incorrect.
  - c. The game counts how often the user submits the form. Display the number of guesses!



```
<!DOCTYPE html><html>
<head lang="en">
  <title>Guess a Number</title>
</head>
<body>
<form method="post">
  <input type="number" name="guess" />
  <input type="submit" value="Check!" />
</form>
```

- b) How are Sessions destroyed in PHP?
- c) How does PHP identify users to keep track of individual sessions?
- d) How do you transparently transmit form data to the server, i.e. without the user immediately seeing the data?



## Task 3: AJAX

Response from movi.es

```
[
  {
    "name" : "Inside Out",
    "year" : 2015,
    "cast" : ["Amy Poehler", "Bill Hader", "Lewis Black"]
  },
  {
    "name" : "The Revenant",
    "year" : 2015,
    "cast" : ["Leonardo DiCaprio", "Tom Hardy", "Will Poulter"]
  },
  {
    "name" : "Fight Club",
    "year" : 1999,
    "cast" : ["Brad Pitt", "Edward Norton", "Helena Bonham Carter"]
  }
]
```

- Explain the advantages and disadvantages of AJAX!
- Explain what the "Same Origin Policy" (SOP) is. Give a concrete example of an AJAX request that will result in a violation of the SOP.
- There is a REST-API that provides a list of movies at the fictional URL <http://movi.es>. It responds with a JSON array like the one above.

- Complete the function `requestMovies()`. It should send a request to the server. Pass `handleMoviesResponse` as success callback for the request.
- Complete the function `handleMoviesResponse(movies)`.
  - `movies` is the response from the server.
  - Generate a separate `<div>` for each movie.
  - Each `div` should have a heading `<h2>` that includes the name and year of the movie (see image on the right)
  - Each `<div>` should include an unordered list `<ul>` of the band's members.
- The script loads in the `<head>` of the HTML document, so potentially the DOM is not ready, when the script is executed. Add a jQuery event handler to make sure that the script is executed only when the DOM is ready.

Movies
<b>Inside Out (2015)</b> <ul style="list-style-type: none"><li>Amy Poehler</li><li>Bill Hader</li><li>Lewis Black</li></ul>
<b>The Revenant (2015)</b> <ul style="list-style-type: none"><li>Leonardo DiCaprio</li><li>Tom Hardy</li><li>Will Poulter</li></ul>
<b>Fight Club (1999)</b> <ul style="list-style-type: none"><li>Brad Pitt</li><li>Edward Norton</li><li>Helena Bonham Carter</li></ul>

For this task, you can choose whether to use jQuery or VanillaJS (we suggest to use jQuery). **Put your code into the code fragment on the next page!**



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Get Movies</title>
  <script src="https://code.jquery.com/jquery-2.1.3.min.js"></script>
  <script>

    var output = $('#output');

    function handleMoviesResponse(movies){

    }

    function requestMovies(){

    }

    requestMovies();

  </script>
</head>
<body>
<div id="output"><h1>Movies</h1></div>
</body></html>
```



- d) The same functionality from the jQuery/VanillaJS code above could be done with Polymer. A colleague of yours sent you an HTML file, which does not seem to work. There are four critical errors. Help him fix it by extending and correcting the following code fragment.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Get Movies</title>
  <script src=
    "bower_components/webcomponentsjs/webcomponents-lite.min.js">
  </script>

</head>
<body>
<div id="output">
  <h1>Movies</h1>

  <iron-ajax auto
    url="movies.json"
    lastResponse="{{moviesResponse}}"></iron-ajax>
  <template is="dom-repeat"
    items="[[moviesResponse]]"
    as="movie">
    <h2>movie.name (movie.year)</h2>
    <ul>
      <template is="dom-repeat"
        items="[[movie.cast]]"
        as="actor">
        <li>[[actor]]</li>
      </template>
    </ul>
  </template>

</div>
</body></html>
```



## Task 4: Node JS

- Provide the npm command to add the body-parser package as a dependency.
- NodeJS has an event-driven I/O. Explain what this means in terms of serving clients.
- The app.js looks is provided in the following. It currently responds to GET <http://localhost:1337/year> with a JSON that contains the current year. Extend the app such that it responds to <http://localhost:1337/time?hours=1&minutes=1> with a JSON like this:

```
{  
  "date": "2016-02-01T10:48:22.658Z",  
  "hours": 11,  
  "minutes": 48  
}
```

The app would respond like this, if it's requested with fewer GET parameters.

<http://localhost:1337/time?hours=1> would give you:

```
{  
  "date": "2016-02-01T14:00:57.301Z",  
  "hours": 11  
}
```

Your task is to write the endpoint and the middleware that handles the requests on this route (similar to the '/year' endpoint). Extend the code fragment below.

```
var express = require('express');  
var bodyParser = require('body-parser');  
  
var app = express();  
app.use(bodyParser.json());  
app.use(bodyParser.urlencoded({ extended: false }));  
  
app.get('/year', function(req, res){  
  res.json({  
    year : new Date().getFullYear()  
  });  
});  
});
```

```
app.listen(1337);
```



## Task 5: Metadata & Multimedia Content Description

Imagine you work at a broadcasting company that manages a large set of multimedia files (mainly videos).

- a) The company uses software to manage the video catalogue. Next week, there will be a short documentary on the presidential election in the U.S. Please name 3 types of criteria that can be used to find related video clips on the topic. Indicate if this data is high- or low-level metadata.

Search / Filter Criterion:	Type of metadata:

- b) So far, the company relies on its employees to manually create and aggregate metadata about any video clip. Explain how this process could be sped up. Include at least one concrete technology in your answer.
- c) The company asks you about your opinion to switch to an MPEG-7 compatible software. Provide one argument both in favor and against MPEG-7.

## Submission

This is a repetition sheet and not submitted to UniWorX.