AJAX

Maher Khan

Recitation 9
Date: 9th November, 2018
slides and codes: https://github.com/maher460/Pitt-CS1520-Recitations
Office hours today from 6:00 PM to 8:00 PM

Table of Contents

Assignment 3

Assignment 3 (Talk Python to Me) comments

- Do not use global variables inside classes
    - Only use class attributes (class wide or object wide)
- slugs(), abbr, and before_year() should be part of Media or Movie class
    - You can set them as static methods (show example)
- private vs public attributes and methods in classes
- abstract classes and methods in python

Assignment 4

Assignment 4 (Treat Yo Self) Tips

- Focus more on the functionalities, features and communications between client (browser) and server (flask)
  - You are not gonna be graded on how “pretty” your app looks, but everything including interactions should be clear and fluid
  - all css and js should be in separate static files which you have to load through flask
- As you are developing and testing your app, check if your functionalities, features and communications are behaving just as you would expect in a real application
- You should have a single login page for all three user types - owner, stylist and patron

Assignment 4 (Treat Yo Self) Tips

- Make sure your initdb and bootstrap cli commands are working:
  - initdb should delete and then create all tables in the database
  - bootstrap should populate the database with dummy data (owner, several stylists, several patrons and a few appointments)
  - Do not expect me to create all owner, patrons, stylists and appointments (I intend to only create a few just for testing)

Assignment 4 (Treat Yo Self) Tips

- Think about space and performance of your flask app, database system and client app (in the following order):

    1. minimize number of queries to database systems
    2. minimize space allocation in database systems
    3. minimize amount of computations in server controllers
    4. minimize communications between client and server
    5. minimize client-side computations

Assignment 4 (Treat Yo Self) Tips

- Think about edge cases that may break your app, e.g:
  - What if you try to access the URL of the stylist page, but you are not logged in?
  - What if a patron tries to access the profile page of another patron, should s/he have access to the other patron's profile page? (What if an owner/stylist visits a patron's profile page?)
  - Even though we are only going to look for basic authentication and authorization, you should still consider hashing your password and generate random session keys

Ajax

What is Ajax?

– Shorthand for Asynchronous JavaScript and XML

– Is a technique of designing web apps / single-page apps

– Makes use of • scripted http requests • (aka web remoting)

– In essence:
  - Without replacing the current page
  - Make a new request to a server
  - Process the response
  - Update the display

Ajax is the core of all single-page apps today:

– Gmail

– Piazza

– Tumblr

– Many many others …

Some Ajax patterns

- Dynamically update content from remote source
- Friendly interactive interface to remote services
- Responsive first-person interaction
  - Instead of click-and-wait
- Collaborative interaction
- Streaming within a web-page context
- Submission throttling
  - E.g. Google docs (saves)
- Partial completion
  - E.g. Google Suggest

Browser-Server communication

- Key to the Ajax style communication is communication between a browser and a server within the context of a page
  - i.e. without a new page load

- XMLHttpRequest is the primary means of facilitating this communication

- All modern browsers have a built-in XMLHttpRequest object.

XMLHttpRequest Overview

1. Get a new XMLHttpRequest object

2. Open it
   - Non-blocking (or blocking (less common))

3. Send a request

4. When complete, process the response

Step 1: Get a new XMLHttpRequest object

- var xhr = new XMLHttpRequest();

Step 2: Open

```
xhr.open("GET", "data.txt", true);
```

- 1st : http request method (text string)
  - i.e. get, post (most often)
  - Also head, put , delete, options

- 2nd: URL of HTTP request (text string)
  - Is relative to the URL of the document containing the script that is calling open.
  - Typically restricted to be on the same protocol://server:port

- 3rd: boolean
  - is the request asynchronous
  - True means a send will not block (asynchronous)
  - AVOID: False means a send will block (synchronous)

Step 3: Send

- GET, HEAD, PUT, DELETE, OPTIONS
  - Put parameters in the open URL query string
  - Then send(null)
  - E.g.:

```
open(“GET","http://localhost/foo/test?param1=x&m2=y", true);
send();
```

- POST – Put parameters as arguments to send
  - E.g.:

```
open("POST", "http://localhost/foo/test", true);
send(“param1=x&m2=y");
```

- Send() initiates communication with the server

Step 4: Handle Response

- Define a callback function to process the response from the server.
- The browser calls an event handler for each change in the state of the XMLHttpRequest object:
- Therefore you must define a callback function

```
xhr.onreadystatechange = function() { …
```

Optional: add headers to the request

- You can add headers to the request
  - E.g. xhr.setRequestHeader("Content-Type", "text/plain");

- Cookies are handled automatically

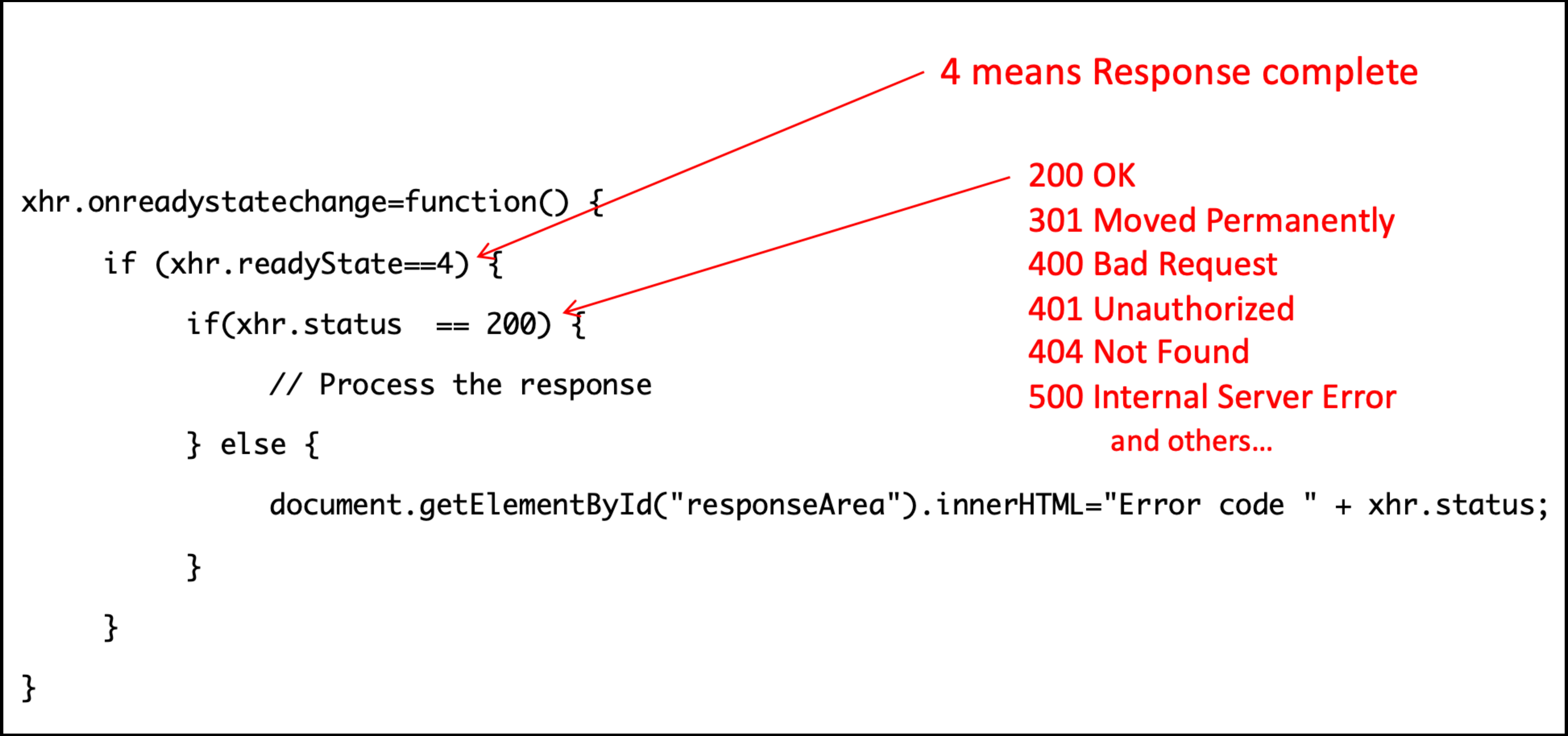
- Some headers are added automatically, for example:
  - referer
  - user-agent

XMLHttpRequest readyState

- xhr.readyState values:
  0. Object has been constructed
  1. Open method successful
     - headers can now be set
     - send can now be done
  2. Response headers have been received from the server
  3. Response body is being received (in progress)
  4. Response complete (or something went wrong)

- The onreadystatechange event fires each time the value of readyState changes.

Response Handler Example



```
xhr.onreadystatechange=function() {
     if (xhr.readyState==4) {
          if(xhr.status  == 200) {
               // Process the response
          } else {
               document.getElementById("responseArea").innerHTML="Error code " + xhr.status;
          }
     }
}
```

4 means Response complete

200 OK
301 Moved Permanently
400 Bad Request
401 Unauthorized
404 Not Found
500 Internal Server Error
and others…

Use the results

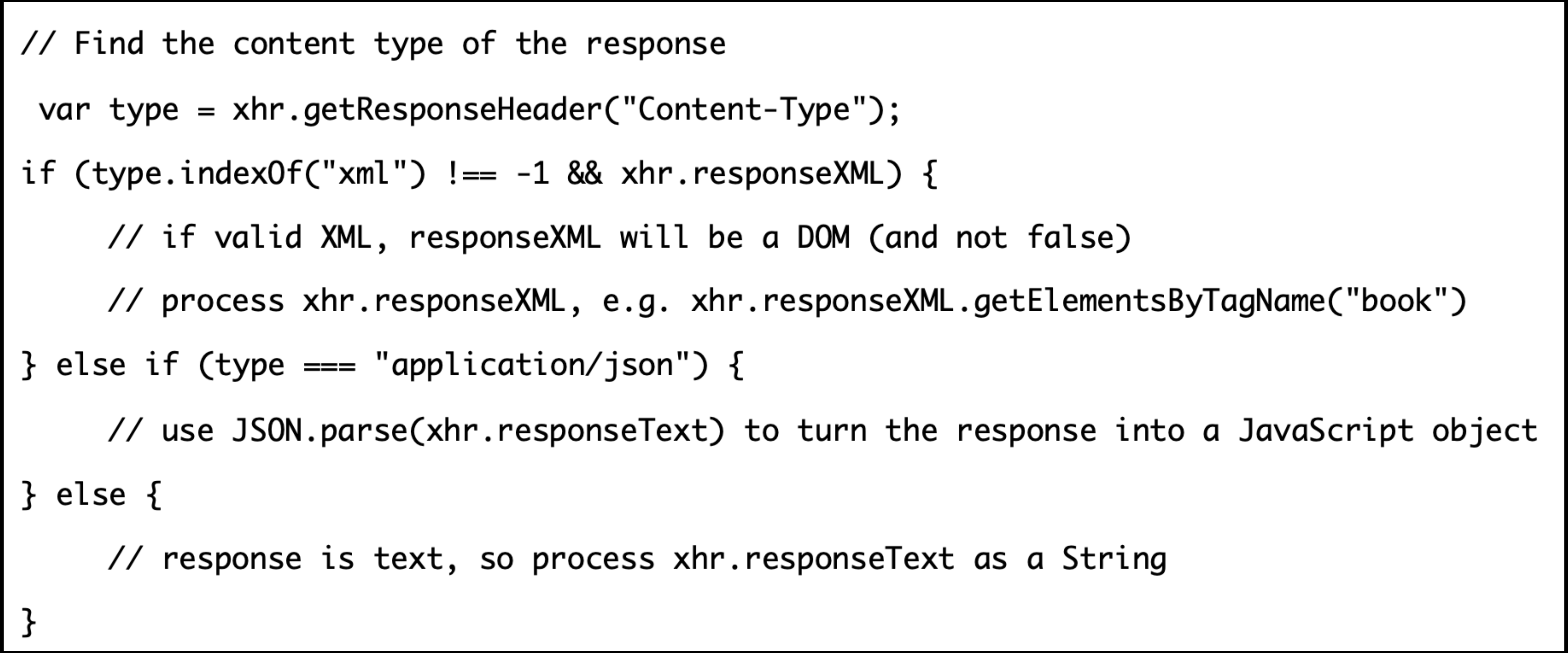
- XML
  - Iff the response is valid XML, it can be treated as such
  - responseXML will be a DOM document object
    - e.g. xhr.responseXML
- JSON
  - Can be parsed into a JavaScript object using JSON.parse()
- Text
  - The results can be treated as String
  - responseText
    - e.g. xhr.responseText
    - Can be manipulated as any String object

What is JSON?

- JavaScript Object Notation
- Lightweight data format
- Essentially, JavaScript literal format
- For details, see http://json.org

Processing the response content

- Most of the time you will know the content
- This demonstrates how to deal with each kind of content.
- And what to do if you are not sure.



```
// Find the content type of the response

 var type = xhr.getResponseHeader("Content-Type");

if (type.indexOf("xml") !== -1 && xhr.responseXML) {

     // if valid XML, responseXML will be a DOM (and not false)

     // process xhr.responseXML, e.g. xhr.responseXML.getElementsByTagName("book")

} else if (type === "application/json") {

     // use JSON.parse(xhr.responseText) to turn the response into a JavaScript object

} else {

     // response is text, so process xhr.responseText as a String

}
```