

Logging+Debugging

Multimedia Engineering II WS 13/14

Sven Spielvogel

Inhalt

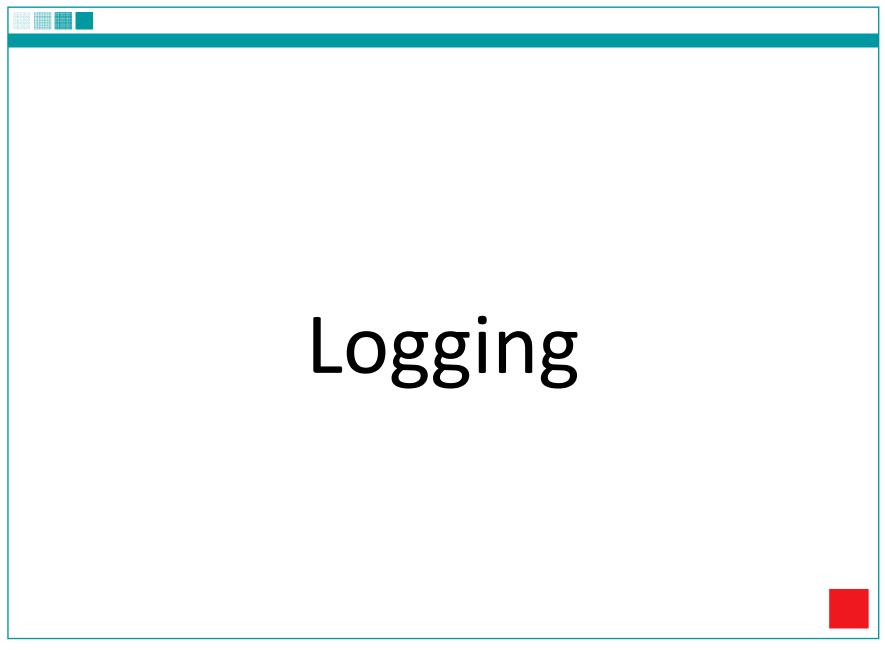
- Logging und Debugging allgemein
- Logging
- Debugging
- LiveEdit



Logging und Debugging allgemein

- Was ist der Unterschied zwischen Logging und Debugging?
- Was macht gutes Logging, bzw. Debugging aus?
- Welche Probleme können dabei auftreten?





Logging

Unter **Logging** versteht man in der Informatik generell das (automatische) Speichern von Prozessen oder Datenänderungen. Diese werden in sogenannten Logdateien hinterlegt bzw. gespeichert.

Wikipedia



Logging

- Logging kann mehrere Ziele haben:
 - Konsole
 - Log-Datei
 - Datenbank
 - Service
 - On-Screen
 - Alert
 - Gesonderter Container

Logging

- Logging hängt von der Laufzeitumgebung ab
 - → Browser
 - Problem: Nicht alle Browser bieten die gleichen Operationen
 - Lösung: Alternative Logging-Wege
 - → Logging-Framework / Klasse

```
// Avoid `console` errors in browsers that lack a console.
(function() {
    var method;
    var noop = function () {};
    var methods = [
        'assert', 'clear', 'count', 'debug', 'dir', 'dirxml', 'error',
        'exception', 'group', 'groupCollapsed', 'groupEnd', 'info', 'log',
        'markTimeline', 'profile', 'profileEnd', 'table', 'time', 'timeEnd',
        'timeStamp', 'trace', 'warn'
    ];
    var length = methods.length;
    var console = (window.console = window.console | | {});
    while (length--) {
        method = methods[length];
        // Only stub undefined methods.
        if (!console[method]) {
            console[method] = noop;
}());
```

- Konsole
 - Ausgabe normal:

```
console.log(object)
```

Ausgabe mehrerer Objekte

```
console.log([object,object,...])
```

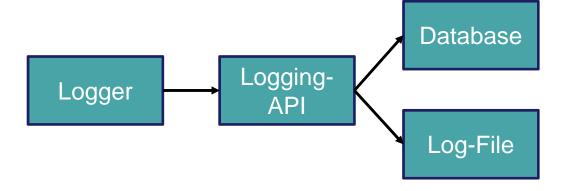
- On-Screen
 - \$("#log").append("...")
- Datei, Datenbank über Service
 - Übermittlung an Serverscript
 - Serverscript speichert Log
 - → Guter Ansatz für mobile Applikationen

JS – Übertragung an Server

```
function(txt) {
  var xhrobj = new XMLHttpRequest();
  var url = 'log.php';
   if(xhrobj) {
       xhrobj.onreadystatechange = function() {;};
       xhrobj.open("POST", url, true);
      params = "log=" + Date.now() + ' ' + txt;
       xhrobj.setRequestHeader("Content-type",
                 "application/x-www-form-urlencoded");
       xhrobj.send(params);
```

■ JS – Übertragung an Server

```
function(txt) {
    $.post('log.php',{data : txt});
};
```



 Der Inhalt der geloggten Information muss passend zum Szenario ausgewählt werden

proto : Array[0]

Initializing Login View

- Im Normalfall gibt uns die Konsole mehrere Informationen
 - Dateiname
 - Zeilennummer
 - Inhalt (String, Objekte, Array)

Achtung! Sollten die Dateien im Anschluss minified/komprimiert werden, gehen Dateiname und Zeilennummer verloren!



main.js:70

login.js:4

- Faustregel: Soviel wie nötig, so wenig wie möglich.
 - Ggf. Log-Level
 - Klassenname
 - Funktionsname
 - Aktion
 - Ggf. Wert(e)
- Beispiel

```
INFO::Main::init::fetched User::id:XYZ
```

ERROR::Main::init::API not found

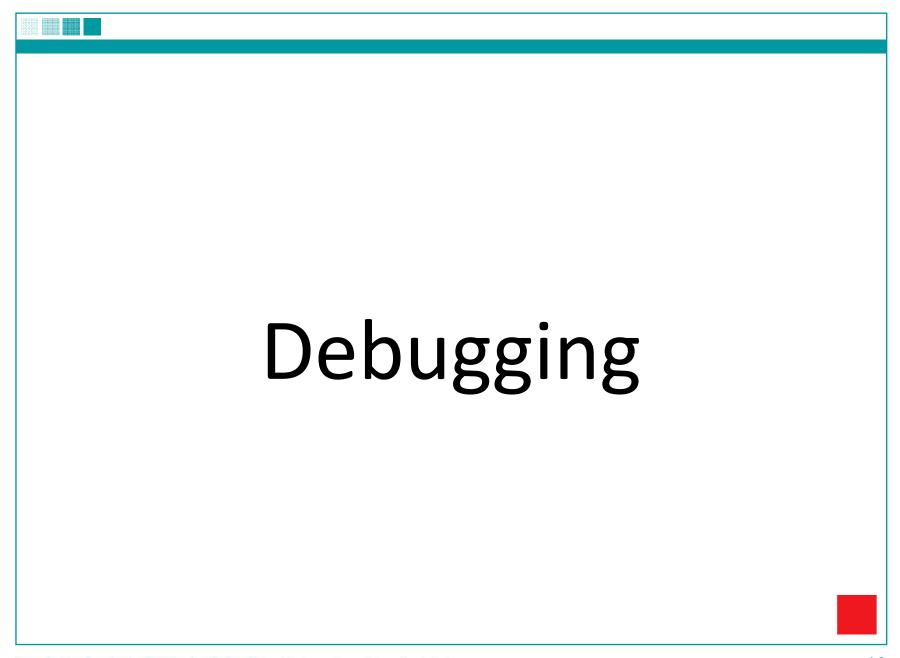
Gerade, wenn man keine Konsole hat (Service Logging), benötigt man so viel Informationen wie möglich.

Logging – nochmal mit Server Logging

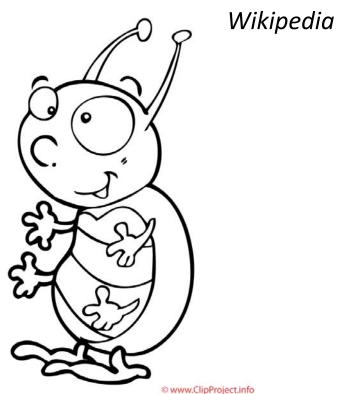
Übertragung an Server

Logging – Zusammenfassung

- Logging ermöglicht das nachträgliche Einsehen der Aktionen/Fehler
- Umfang und Menge liegt in der Hand des Entwicklers
- Abstufung ist wichtig! Zu viele Informationen könne hinderlich sein
- Zahlreiche Frameworks:
 - http://ajaxpatterns.org/Javascript Logging Framewo rks
- Eigenes Framework?



Ein Debugger (von engl. bug im Sinne von Programmfehler) ist ein Werkzeug zum Diagnostizieren und Auffinden von Fehlern in Computersystemen, dabei vor allem in Programmen, aber auch in der für die Ausführung benötigten Hardware.



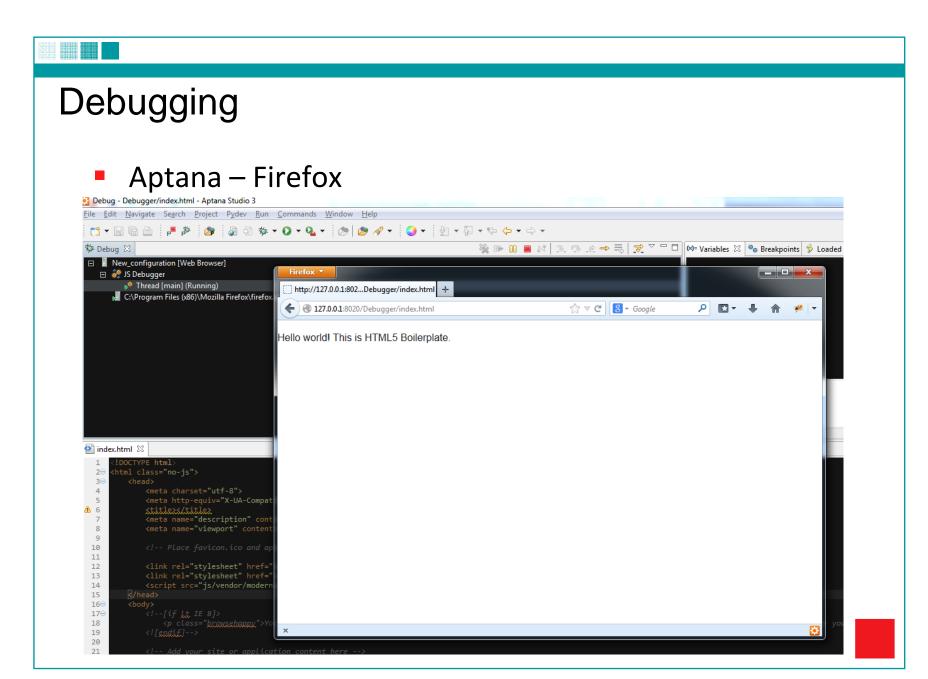
- Aptana
 - Firefox
- WebStorm/PHPStorm
 - Chrome

- Aptana Firefox
 - Debugger Extension installieren
 - Wird von Aptana abgefragt



Aptana – Firefox





Breakpoints

```
console.log("hello");
console.log("i'm not running yet...");
4
5
```

Steuerung

```
ation [Web Browser]

er

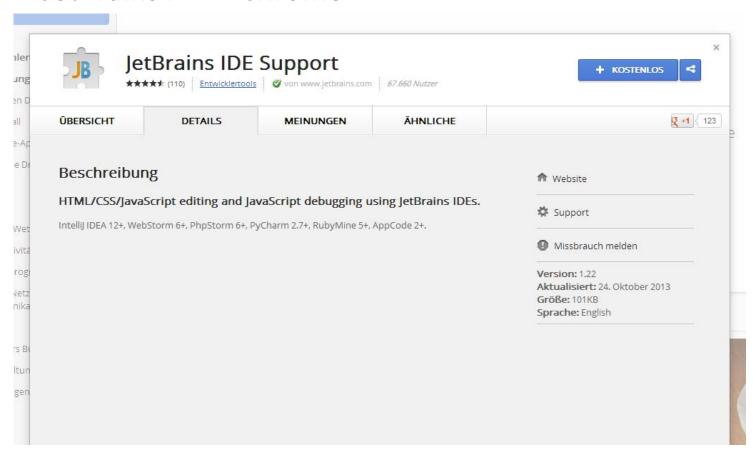
I [main] (suspending) (Running)
```

- JetBrains IDE Chrome
 - Fehlermeldung, wenn Erweiterung nicht gefunden wurde





JetBrains IDE – Chrome



JetBrains IDE – Chrome

```
<script src="js/vendor/modernizr-2.6.2.min.js</pre>
        1005 jquery-1, 10.2,mi
                                You are using an <st
<![endif]-->
                                 <!-- Add your site or application content here
                                    Hello world! This is HTML5 Boilerplate.
   404.html
   apple-touch-icon-precom 27
                                    <div id="pluginTarget">Nothing yet!</div>
   CHANGELOG.md
                                   <script src="//ajax.googleapis.com/ajax/libs/j</pre>
bug 🛜 meinPlugin
 Debugger Console - Elements
     Invalid CSS media query: (-o-min-device-pixel-ratio: 5/4)
      Consider using 'dppx' units instead of 'dpi', as in CSS 'dpi' means dots-per-CS
      [object Object] 18\mein. 18:3
```

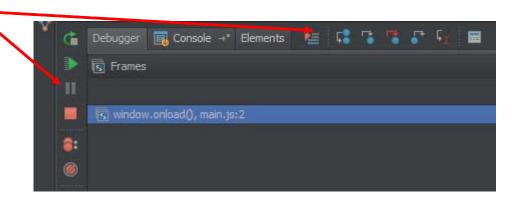
'JetBrains IDE Support" führt eine Problembehebung für diesen Tab durch.

Hello world! This is HTML5 Boilerplate.

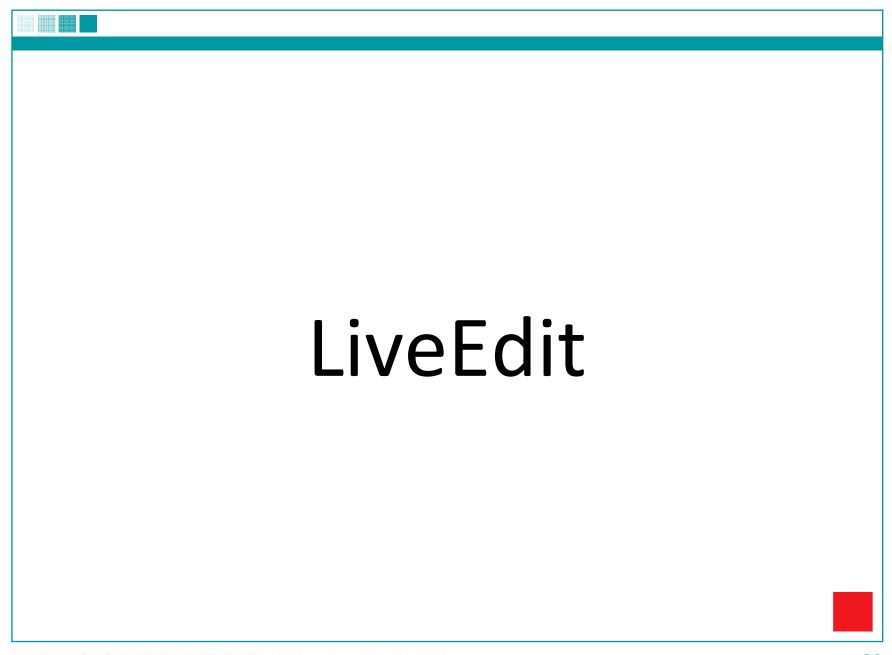
MME2 ist cool!

Breakpoints

Steuerung



- Vorteile
 - Einfaches nachvollziehen der Abläufe zur Laufzeit
 - Arbeiten direkt am Code bzw. in der IDE
 - Bessere Möglichkeit, Fehler nachzuvollziehen
 - Integrierter Web-Server
- Probleme
 - Step-by-Step in Closures, bzw. in verschachtelten Funktionen nicht möglich
 - Integrierter Web-Server unterstützt nicht alle Funktionen. Ggf. muss zusätzlich ein weiterer Server eingesetzt werden.
 - → kann mit komplett eigenem Server umgangen werden



LiveEdit – JetBrains IDE

- Ermöglicht das Editieren von CSS, HTML und JavaScript zur Laufzeit
- Änderungen werden gleich im Browser angezeigt
- Änderungen werden besonders hervorgehoben



→Live-Demo

Vielen Dank für die Aufmerksamkeit!