# Installation

| Key | Value |
| --- | --- |
| Key | **Value** |
| NodeJS YSlow  Setup | The following has to be done to make the NodeJS server working:   * Do the basic setup as described on the "YSlow Docu" page.   + <http://yslow.org/node-server/> * The package "JSDom" had to be installed separately for some unknown reason:   + npm install jsdom * The "yslow-server.js" downloaded from the above "YSlow Docu" page had to be patched like this at the beginning of the code:   'use strict';  var jsdom = require('jsdom');  const { JSDOM } = jsdom;  const { document } = (new JSDOM('')).window;  global.document = document; var doc = document;  var express = require('express'),     YSLOW = require('yslow').YSLOW,     /\* doc = require('jsdom').jsdom(), \*/     http = require('http'),     url = require('url'),     fs = require('fs'),    * When installing express, install the really old version 3.5.0. Else you get an error that "Bundles are not provided by Express anymore", installing these bundles did not make it work either:   + npm install express@3.5.0 * The file "./custom\_yslow/custom\_yslow.js" is a customized yslow.js. To deploy the file it has to be copied to NodeJS in the following location: "<NODEJS\_ROOT>\node\_modules\yslow\lib\yslow.js * Run NodeJS: node yslow-server.js <port> |
| Page Analyzer Server | Deploy the file “.dist/pageanalyzer.war” in your preferred web server. |
| Configure JVM properties | Use the following java properties for configuration:  # The host of the YSlow nodeJS server (Default: 'localhost')  -Dpageanalyzer.nodejs.hostname=127.0.0.1    # The port of the YSlow nodeJS server (Default: '9999')  -Dpageanalyzer.nodejs.port=9999  # The support details as semicolon separated string included on the docu page (Default: null)  -Dpageanalyzer.support.details="Phone: +4100 000 00 00;Email: foo@bar.com" |
| Configure server properties | The following is an example for JBoss web server, this was added to standalone.xml: <server>   ...      <system-properties>          <property name="pageanalyzer.nodejs.hostname" value="127.0.0.1"/>          <property name="pageanalyzer.nodejs.port" value="9999"/>          <property name="pageanalyzer.support.details" value=" Phone: +4100 000 00 00;Email: foo@bar.com” />      </system-properties>  ... |

# Access Page Analyzer

To access the application use the following URL:

* http://<yourservername>:<port>/pageanalyzer/harupload
* There is as well a REST API, which returns the results as JSON. Use the following URL to access the docu through a browser
  + http://<yourservername>:<port>/pageanalyzer/api

# Custom Ruleset: pageanalyzer

Page analyzer uses its own ruleset, as there where several issues with the existing rules and yslow implementations in general:

|  |  |
| --- | --- |
| Key | Value |
| Key | **Value** |
| cset.getComponentsByType() Issue | Components(~Requests) are selected by for the rule evaluation with the method getComponentsByType(). Most of the YSlow rules ignore components completely when they are requested after the onload method was executed. Therefore most of the rules did not a sufficient analysis, by adding the parameter "true" to the method this gets fixed. |
| Messages | Messages set by the rules where often not returned by the nodejs server.  This was fixed by putting the messages in the "components" part of the results. Also some messages where missing important details. |
| Not checking if rule applicable | Most rules did not check if it is applicable to the analyzed page. For example if a page did not do any AJAX requests it still got 100% for this rule instead of "N/A". That increased the overall score of the page, what gave a wrong picture in the end. |
| Not working without pages | It was not working if no pages and only entries where defined in the .har-File (e.g. Fiddler export). This issue was fixed by iterating over all entries and ignore pages. |
| Only First Page Analyzed | Only the first page in the .har-File was analyzed with the out of the box yslow.js, it was changed to analyze all pages. |
| pacacheajax | improved version of "yxhr". |
| paexpires | * Improved version of "yexpires", the rule was not working properly as it took the current date and time when the yslow analysis was exectued, instead of using the time the page was actually loaded, what lead to false positives if you analyzed a .har-File which was created some time ago. * The yexpires rule did also not consider when there was not expiration was set at all and did not blame in suhc cases. * Also some details where added to the output(expiration timestamp and relative time). |
| paexternalcss  paexternaljavascript | Rule "yexternal" was split up to get more distinction. |
| pafavicon | Improved version of "yfavicon", the rule was not working correctly as it extracted the favicon paths from the header(in most cases relative) and compared it to the components urls (often absolute), so it was not able to identify the icons. |
| pagetforajax | improved version of "yxhrmethod". |
| paimgnoscale | Improved version of "yimgnoscale". |
| pajsbottom | * Improved version of "yjsbottom". the rule only checked javascripts in the <head>-Tag and not in hte <body>-Tag, therefore scripts might have not have been analyzed(also it did not check anything that came after on load). * The yjsbottom rule did not consider inline scripts. * Added some details to the results. |
| pamergejs  pamergecss | Custom rules, check if it would make sense to merge javascript or css files. |
| pamincookie | Improved version of "ymincookie", the rule was only checking cookies of the main document and ignored all other cookies. |
| paminifycss  paminifyjavascript  paminifyjson  paminifyxml  paminifyxhr | Rule "yminify" was split up to get more distinction between css and javascript. The rules for other formats are custom rules. Also the YSlow method"isMinified" was improved to get better results. |
| panohttp4xx | improved version of "yno404". Will check for any HTTP 4xx status and will blame for it instead of only |