ImmobilienScout 24 Technical Challenge

The objective is to build a web application that allows a user to conduct some analysis of an HTML web page.

We need the following simple interaction:

- The user is presented with a page containing a text field into which a HTTP(s) URL can be entered
- The form is accompanied by a button to submit the form to the server
- The server conducts the analysis (described below) and the results are displayed below the form, preferably in a tabular form.

For a valid URL that returns HTML and which is reachable from the server, the server should extract the following information:

- HTML version of the document
- Page title, if any.
- Number of headings grouped by heading level
- Number of hypermedia links in the document, grouped into "internal" links to the same domain and "external" links to other domains
- Does the page contain a login form? You will need to provide some plausible logic for detecting such a form in at least some cases given the multiple ways (and human languages) that might be used to construct login forms. We expect the login forms on https://www.spiegel.de/meinspiegel/login.html and https://github.com/login to be detectable.

Optional: Having collected the links on an HTML page, provide validation that each linked resource is available via HTTP(S). Consider the effect of redirection. Collect the results of attempting to reach all the links (either true or false). In the case of an unreachable link, provide some information on what went wrong. Display these results in the web page. For this part performance is relevant, so please document your thoughts on implementing it in an efficient way.

Please zip your solution and include the following:

- The full sources, including tests, comments, etc.
- A build file. You are strongly encouraged to use appropriate libraries – we are not expecting you to build a HTML parser from hand.
- A README that documents:
    - How to build and run your solution locally.
    - The assumptions you made, design decisions you took and the known constraints or limitations in your solution.

Your solution will be reviewed by two of our engineers. In order to help them do a fair job, we request that you submit anonymous code. Please try to avoid using your own domain in packages, sending us your git repository and generated code with your full name.

The server solution should run either on the JVM or in node.js. Please note the following:

	JVM	**Node.js**
Language	One of Java 8, Scala	Javascript
Runtime	JDK 8 on Unix (Linux, Mac)	Latest release of Node.js on Unix (Linux, Mac). npm
Build	One of: Maven, Gradle, SBT	One of: NPM, Grunt, Gulp
HTML Parser	Consider JSoup	Consider Cheerio

Finally – keep your solution simple, but not naïve. Provide robustness, but don't over-engineer. And your code should be presentable, but not gold-plated.

Good luck! We're looking forward to your submission.