Developing Mobile Views

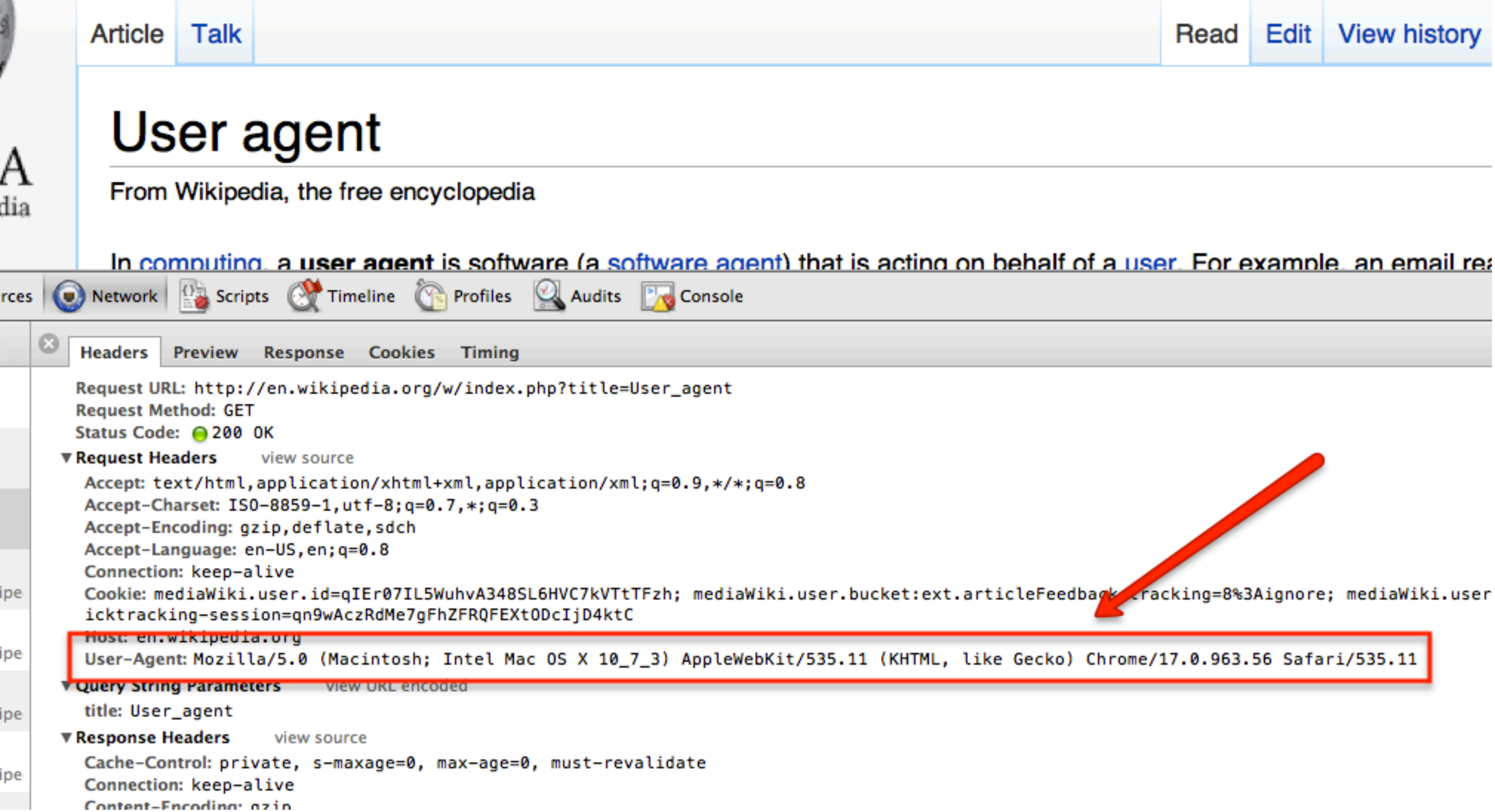
Approaches

- Detect devices
- Responsive Design
- Hybrid

User Agent

- Browsers send a User Agent string to the server, examples:
  - User-Agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/535.11 (KHTML, like Gecko) **Chrome**/17.0.963.56 Safari/535.11
  - Mozilla/5.0 (**iPhone**; CPU iPhone OS 5_0 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3
- Server examines the user agent and redirects the user to a mobile specific set of views

User Agent



User agent

From Wikipedia, the free encyclopedia

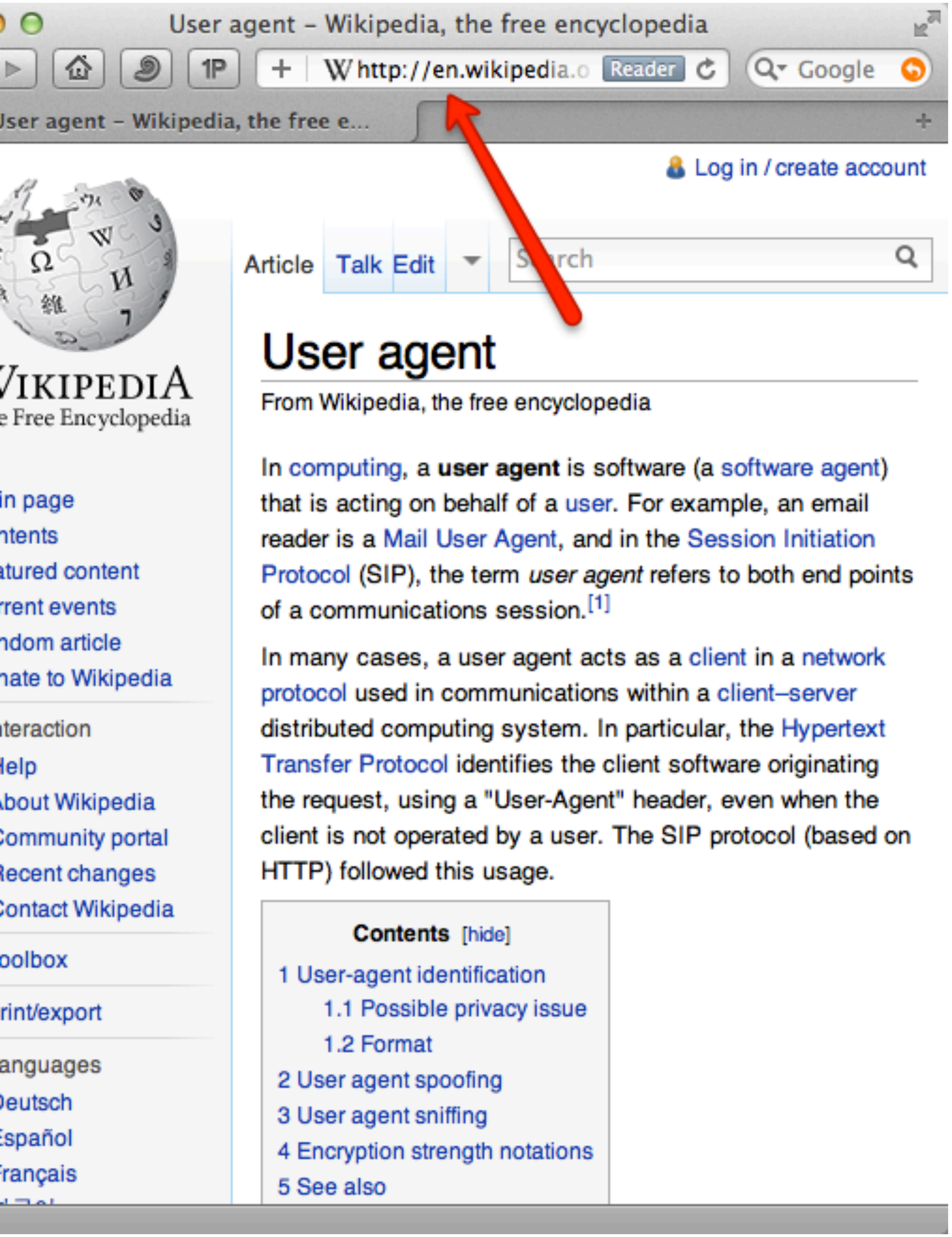
In computing, a **user agent** is software (a software agent) that is acting on behalf of a user. For example, an email rea

```
Request URL: http://en.wikipedia.org/w/index.php?title=User_agent
Request Method: GET
Status Code: ● 200 OK
▼Request Headers      view source
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
  Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
  Accept-Encoding: gzip,deflate,sdch
  Accept-Language: en-US,en;q=0.8
  Connection: keep-alive
  Cookie: mediaWiki.user.id=qIEr07IL5WuhvA348SL6HVC7kVTtTFzh; mediaWiki.user.bucket:ext.articleFeedback-tracking=8%3Aignore; mediaWiki.user
  icktracking-session=qn9wAczRdMe7gFhZFRQFEXtODcIjD4ktC
  Host: en.wikipedia.org
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.56 Safari/535.11
▼Query String Parameters      view URL encoded
  title: User_agent
▼Response Headers      view source
  Cache-Control: private, s-maxage=0, max-age=0, must-revalidate
  Connection: keep-alive
  Content-Encoding: gzip
```

Testing Device Detection

- Firefox - User Agent Switcher
- Chrome - Developer Tools > Settings > Override User Agent
- Safari - Develop Menu Bar
- IE - http://www.howtogeek.com/howto/18450/change-the-user-agent-string-in-internet-explorer-8/ (is there a better way?)

Safari



User agent

From Wikipedia, the free encyclopedia

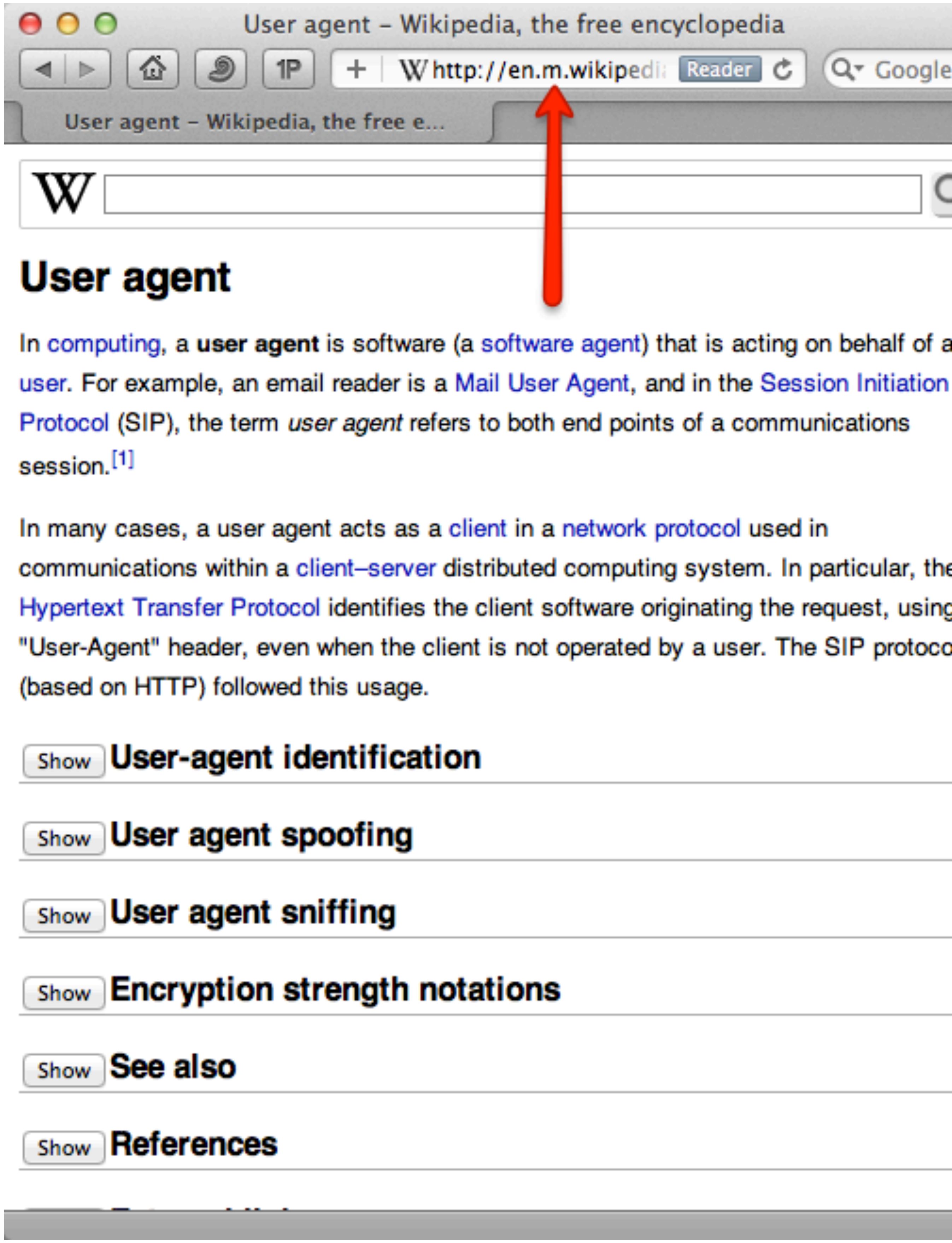
In computing, a **user agent** is software (a software agent) that is acting on behalf of a user. For example, an email reader is a Mail User Agent, and in the Session Initiation Protocol (SIP), the term *user agent* refers to both end points of a communications session.[1]

In many cases, a user agent acts as a client in a network protocol used in communications within a client–server distributed computing system. In particular, the Hypertext Transfer Protocol identifies the client software originating the request, using a "User-Agent" header, even when the client is not operated by a user. The SIP protocol (based on HTTP) followed this usage.

Contents [hide]

1 User-agent identification
1.1 Possible privacy issue
1.2 Format
2 User agent spoofing
3 User agent sniffing
4 Encryption strength notations
5 See also

iPhone


User agent – Wikipedia, the free encyclopedia
http://en.m.wikipedi
Reader
Google
User agent – Wikipedia, the free e...
W

User agent

In computing, a **user agent** is software (a software agent) that is acting on behalf of a user. For example, an email reader is a Mail User Agent, and in the Session Initiation Protocol (SIP), the term *user agent* refers to both end points of a communications session.[1]

In many cases, a user agent acts as a client in a network protocol used in communications within a client–server distributed computing system. In particular, the Hypertext Transfer Protocol identifies the client software originating the request, using "User-Agent" header, even when the client is not operated by a user. The SIP protoco (based on HTTP) followed this usage.

Show **User-agent identification**

Show **User agent spoofing**

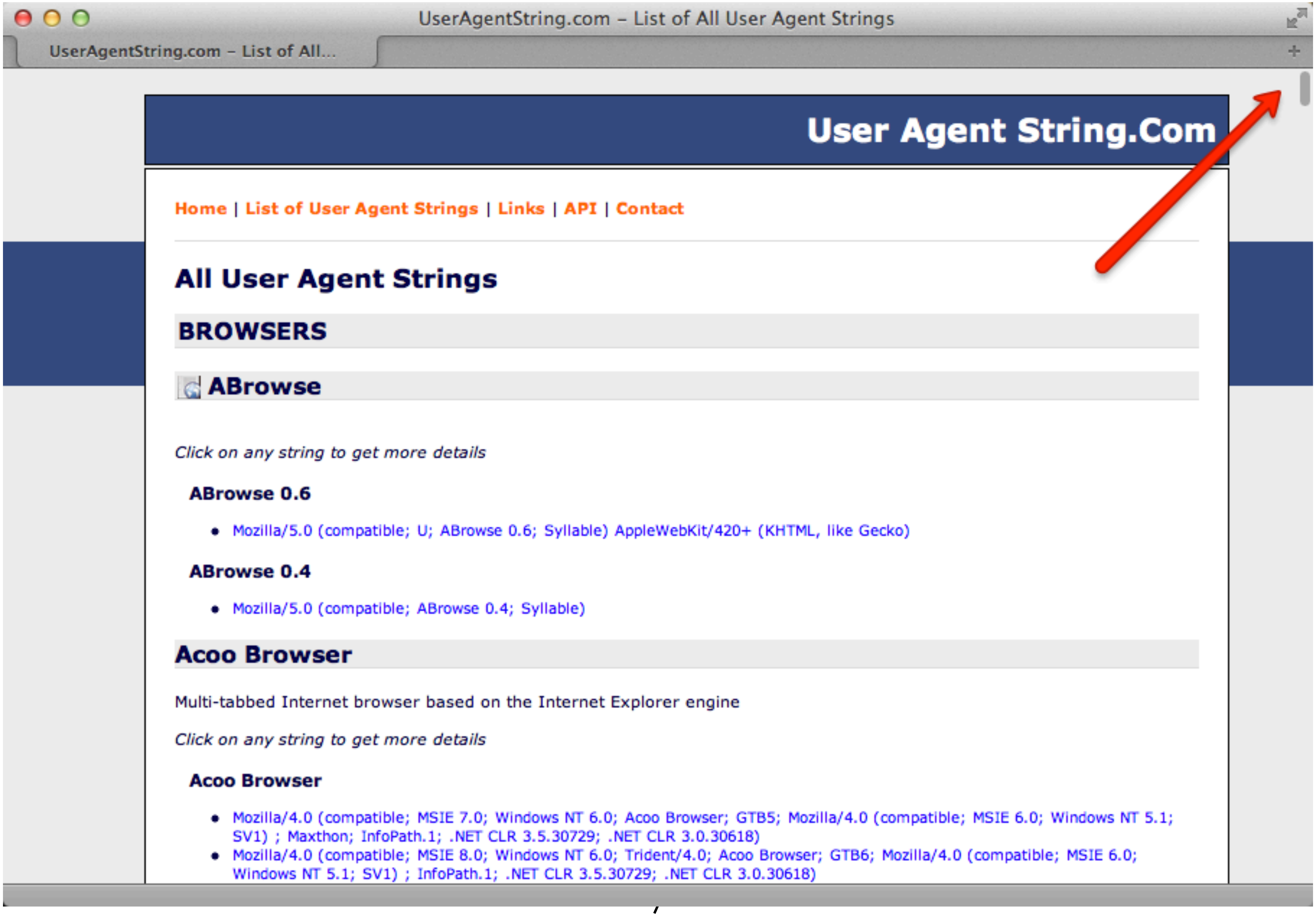
Show **User agent sniffing**

Show **Encryption strength notations**

Show **See also**

Show **References**

http://www.useragentstring.com/pages/All/



UserAgentString.com – List of All User Agent Strings
UserAgentString.com – List of All...
User Agent String.Com
Home | List of User Agent Strings | Links | API | Contact

All User Agent Strings

BROWSERS

ABrowse

Click on any string to get more details

ABrowse 0.6

- Mozilla/5.0 (compatible; U; ABrowse 0.6; Syllable) AppleWebKit/420+ (KHTML, like Gecko)

ABrowse 0.4

- Mozilla/5.0 (compatible; ABrowse 0.4; Syllable)

Acoo Browser

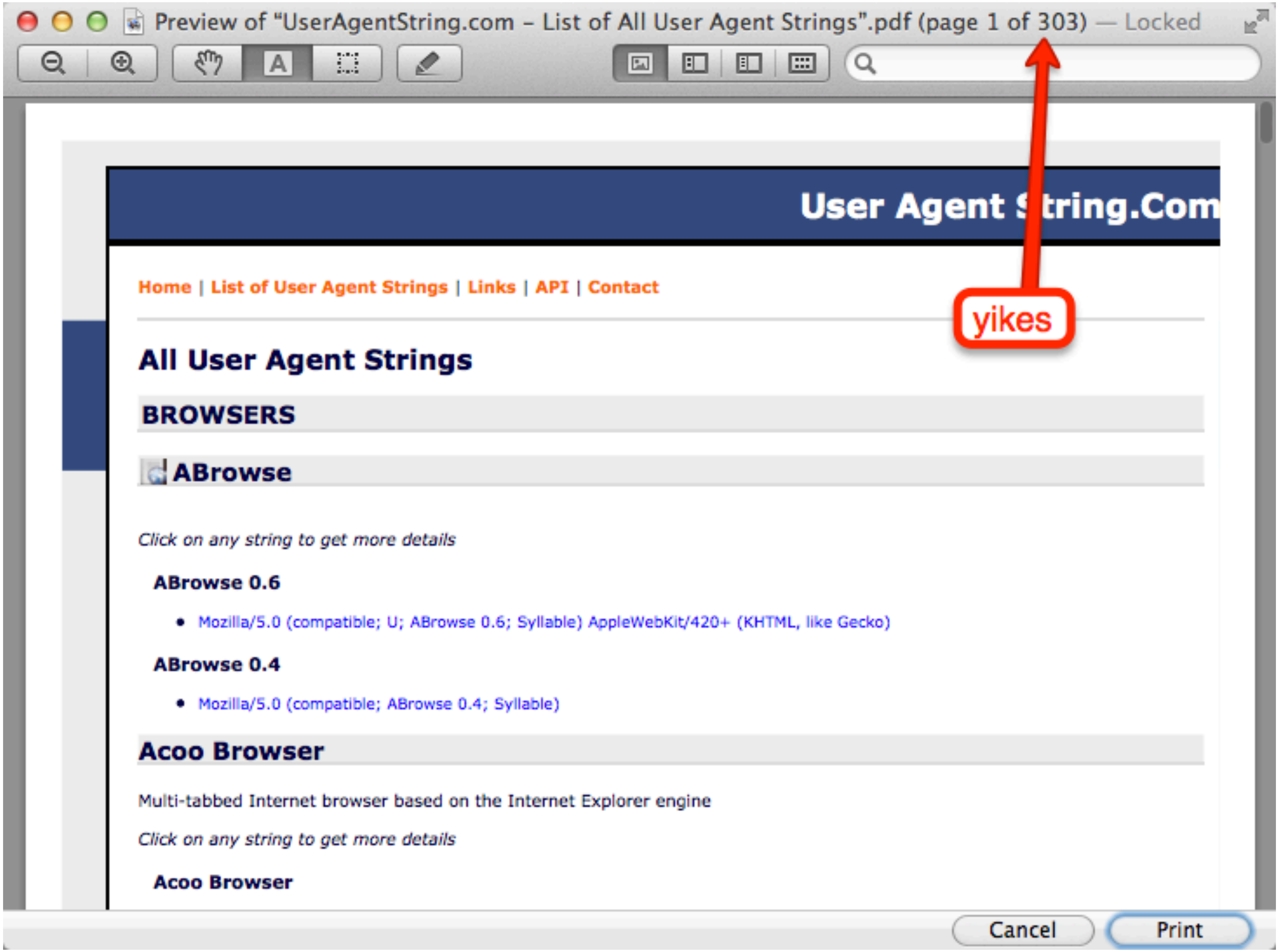
Multi-tabbed Internet browser based on the Internet Explorer engine

Click on any string to get more details

Acoo Browser

- Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; Acoo Browser; GTB5; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; Maxthon; InfoPath.1; .NET CLR 3.5.30729; .NET CLR 3.0.30618)
- Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0; Acoo Browser; GTB6; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; InfoPath.1; .NET CLR 3.5.30729; .NET CLR 3.0.30618)

http://www.useragentstring.com/pages/All/



Preview of "UserAgentString.com – List of All User Agent Strings".pdf (page 1 of 303) — Locked
yikes
Cancel
Print

User Agent String.Com

Home | List of User Agent Strings | Links | API | Contact

All User Agent Strings

BROWSERS

ABrowse

Click on any string to get more details

ABrowse 0.6

- Mozilla/5.0 (compatible; U; ABrowse 0.6; Syllable) AppleWebKit/420+ (KHTML, like Gecko)

ABrowse 0.4

- Mozilla/5.0 (compatible; ABrowse 0.4; Syllable)

Acoo Browser

Multi-tabbed Internet browser based on the Internet Explorer engine

Click on any string to get more details

Acoo Browser

User Agent Reliability

In the beginning there was NCSA Mosaic, and Mosaic called itself NCSA_Mosaic/2.0 (Windows 3.1), and Mosaic displayed pictures along with text, and there was much rejoicing

http://webaim.org/blog/user-agent-string-history/

User Agent Reliability

And behold, then came a new web browser known as “Mozilla”, being short for “Mosaic Killer,” but Mosaic was not amused, so the public name was changed to Netscape, and Netscape called itself Mozilla/1.0 (Win3.1), and there was more rejoicing.

http://webaim.org/blog/user-agent-string-history/

User Agent Reliability

And Netscape supported frames, and frames became popular among the people, but Mosaic did not support frames, and so came “user agent sniffing” and to “Mozilla” webmasters sent frames, but to other browsers they sent not frames.

http://webaim.org/blog/user-agent-string-history/

User Agent Reliability

And then Google built Chrome, and Chrome used Webkit, and it was like Safari, and wanted pages built for Safari, and so pretended to be Safari. And thus Chrome used WebKit, and pretended to be Safari, and WebKit pretended to be KHTML, and KHTML pretended to be Gecko, and all browsers pretended to be Mozilla, and Chrome called itself Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.13 (KHTML, like Gecko) Chrome/0.2.149.27 Safari/525.13, and the user agent string was a complete mess, and near useless, and everyone pretended to be everyone else, and confusion abounded.

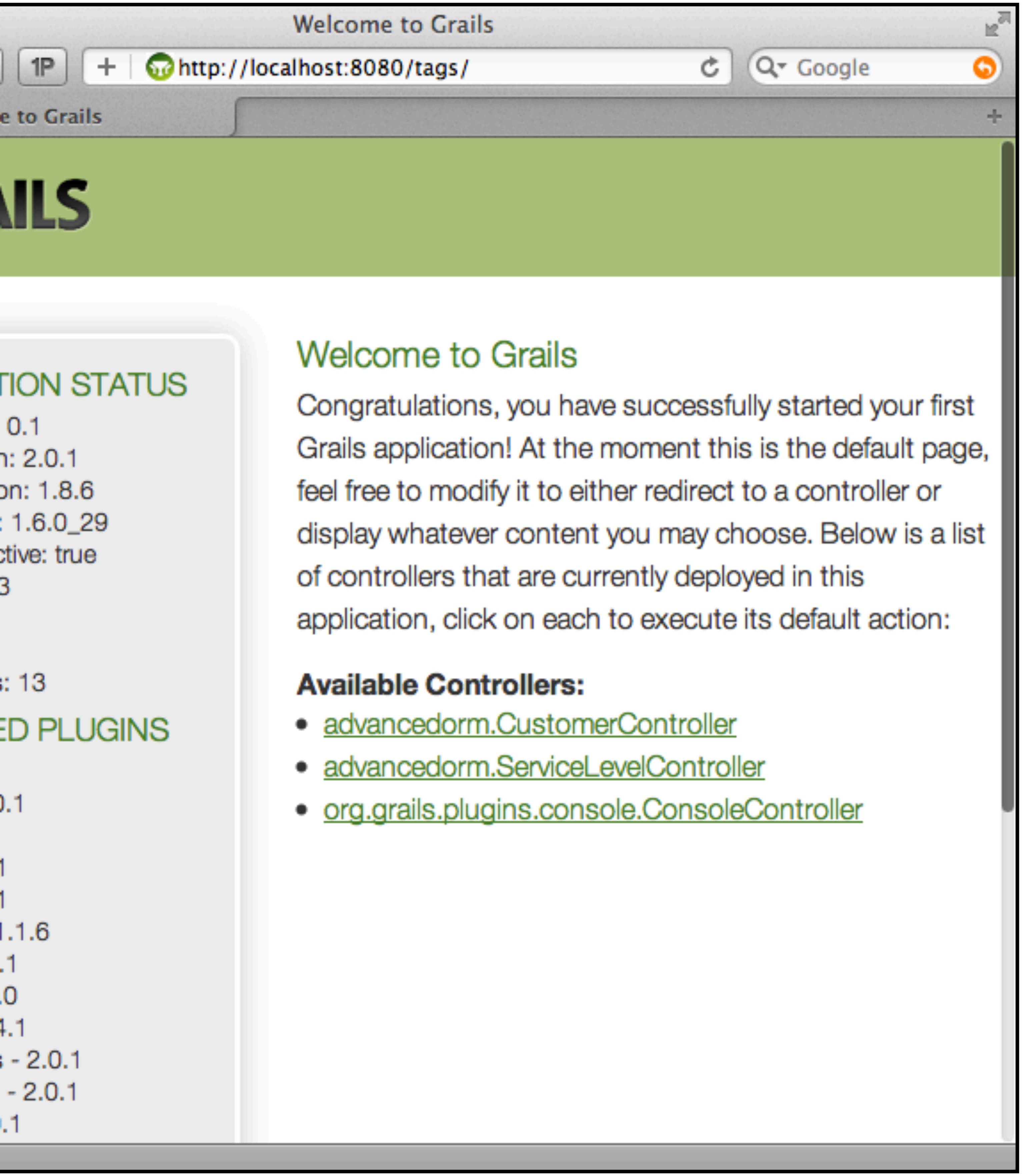
http://webaim.org/blog/user-agent-string-history/

User Agent Reliability

- Enter: Wireless Universal Resource File
  - http://wurfl.sourceforge.net/
  - AGPL license
- Abstraction layer that provides:
  - device name, make, model in a consistent fashion
  - device capabilities
  - device metadata (isTablet, isWireless, isMobile, etc.)

Responsive Design

- Use CSS3 media queries to adapt the presentation to fit the viewport
- Advantage - you only maintain one set of views
- Already built in to Grails scaffolding - just resize your browser
- Another example: Twitter Bootstrap:
  - http://twitter.github.com/bootstrap/



TION STATUS
0.1
n: 2.0.1
on: 1.8.6
: 1.6.0_29
ctive: true
3

s: 13

ED PLUGINS

0.1

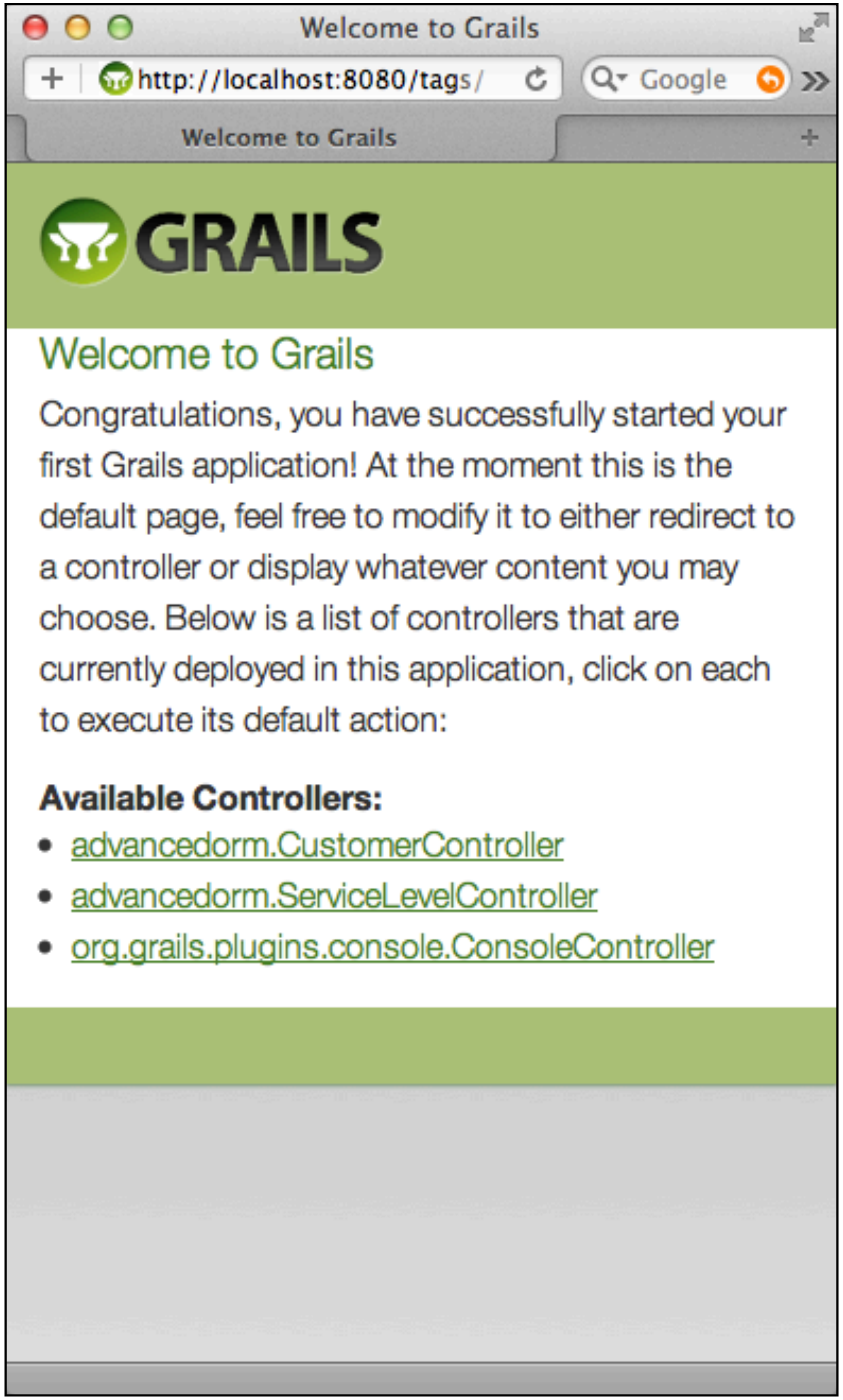
1
1
1.1.6
.1
.0
4.1
s - 2.0.1
- 2.0.1
.1

Welcome to Grails

Congratulations, you have successfully started your first Grails application! At the moment this is the default page, feel free to modify it to either redirect to a controller or display whatever content you may choose. Below is a list of controllers that are currently deployed in this application, click on each to execute its default action:

Available Controllers:

- advancedorm.CustomerController
- advancedorm.ServiceLevelController
- org.grails.plugins.console.ConsoleController

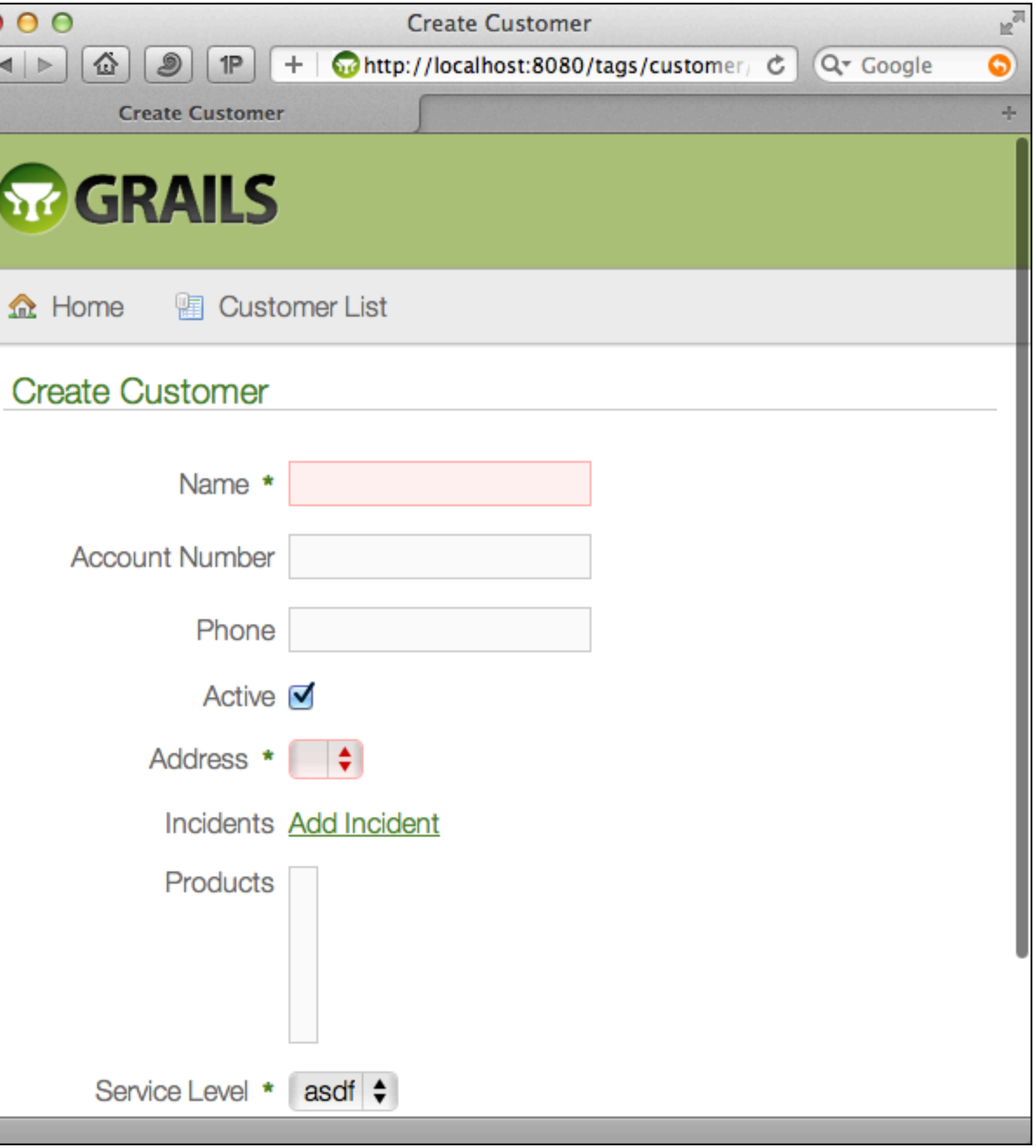


Welcome to Grails

Congratulations, you have successfully started your first Grails application! At the moment this is the default page, feel free to modify it to either redirect to a controller or display whatever content you may choose. Below is a list of controllers that are currently deployed in this application, click on each to execute its default action:

Available Controllers:

- advancedorm.CustomerController
- advancedorm.ServiceLevelController
- org.grails.plugins.console.ConsoleController



Create Customer

Name *

Account Number

Phone

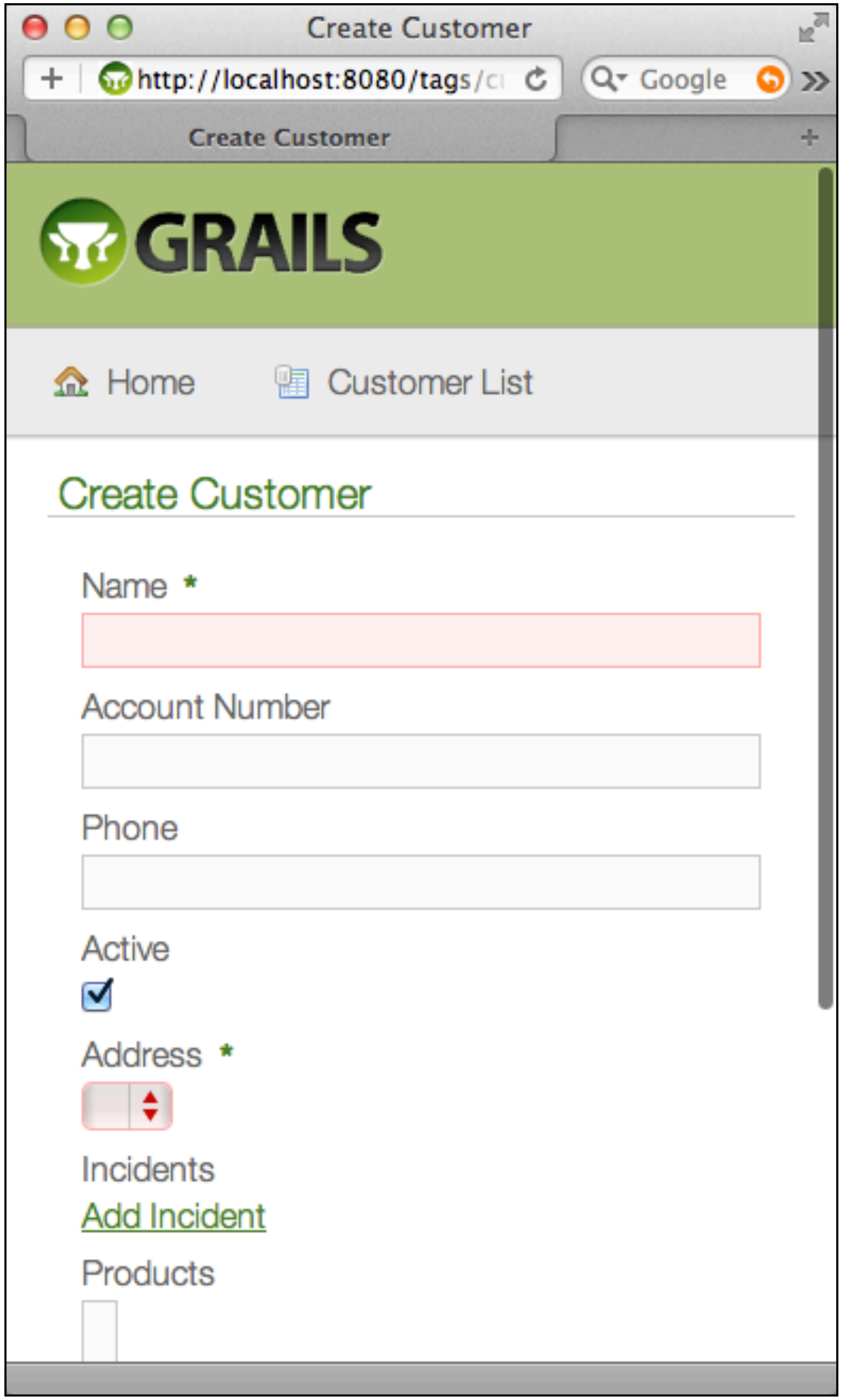
Active ☑

Address *

Incidents Add Incident

Products

Service Level * asdf



Create Customer

Name *

Account Number

Phone

Active

☑

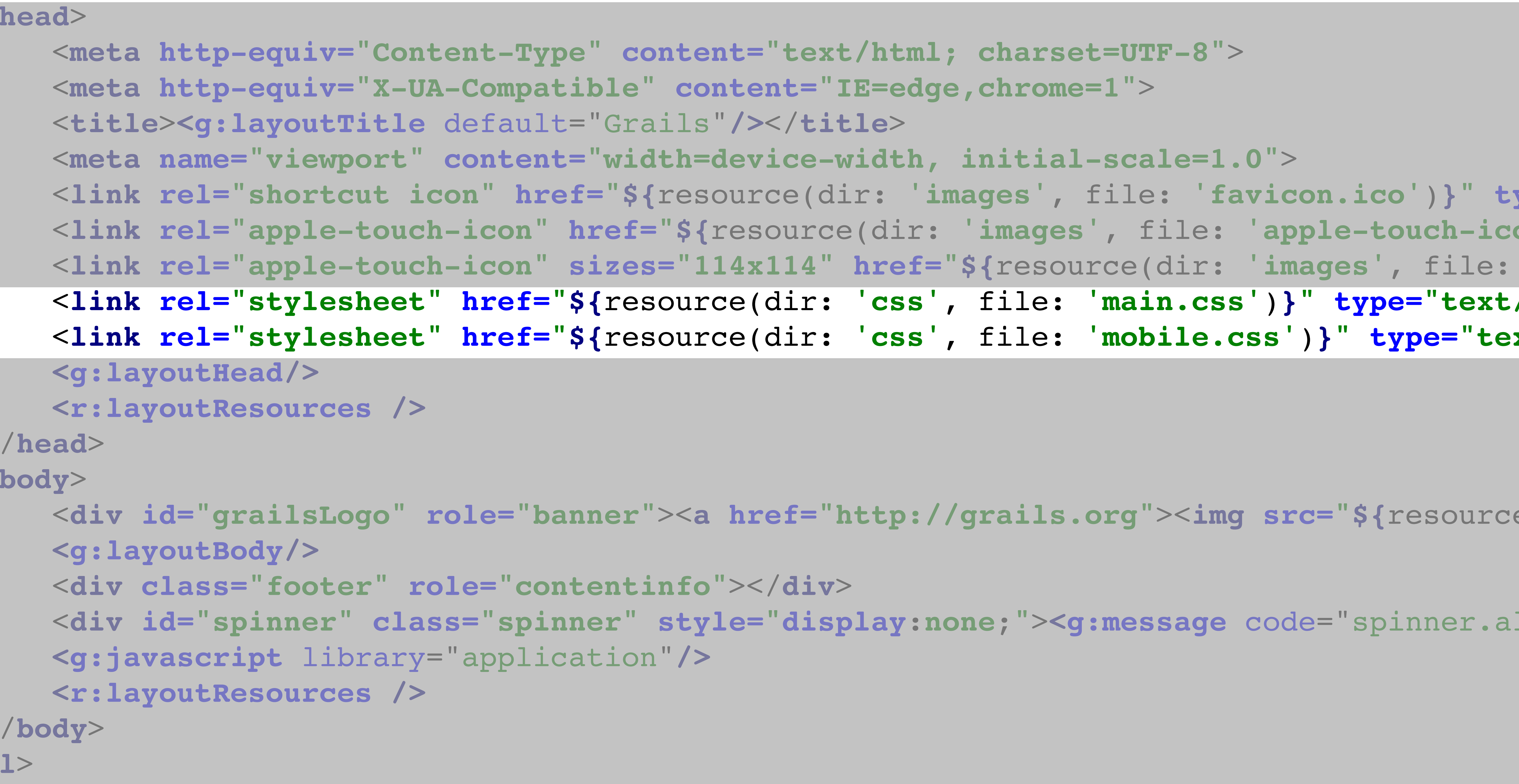
Address *

Incidents

Add Incident

Products

main.gsp



```
head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
   <title><g:layoutTitle default="Grails"/></title>
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="shortcut icon" href="${resource(dir: 'images', file: 'favicon.ico')}" ty
   <link rel="apple-touch-icon" href="${resource(dir: 'images', file: 'apple-touch-ico
   <link rel="apple-touch-icon" sizes="114x114" href="${resource(dir: 'images', file:
   <link rel="stylesheet" href="${resource(dir: 'css', file: 'main.css')}" type="text/
   <link rel="stylesheet" href="${resource(dir: 'css', file: 'mobile.css')}" type="tex
   <g:layoutHead/>
   <r:layoutResources />
/head>
body>
   <div id="grailsLogo" role="banner"><a href="http://grails.org"><img src="${resource
   <g:layoutBody/>
   <div class="footer" role="contentinfo"></div>
   <div id="spinner" class="spinner" style="display:none;"><g:message code="spinner.al
   <g:javascript library="application"/>
   <r:layoutResources />
/body>
l>
```

mobile.css

```css
/* Styles for mobile devices */

@media screen and (max-width: 480px) {
    .nav {
        padding: 0.5em;
    }

    .nav li {
        margin: 0 0.5em 0 0;
        padding: 0.25em;
    }
 ........
```

Responsive Design

- Good for content websites - news, blogs, etc.
- Disadvantages
  - Performance
  - Relies on newer CSS and JavaScript capabilities
  - Difficult if you don't already have a flexible layout

Hybrid Approach

- Use both User Agent sniffing and Responsive Design
  - use user agent to determine what size images to send to browser
  - add logic to views to only render specific content for certain types of devices (e.g. CSS `display:none` still downloads the resource, which uses bandwidth)

Grails User Agent Detection

- Utilize Spring Mobile (plugin) for basic mobile device detection
  - two resolvers, “lite” and WURFL

```
def list = {
  def view = "list"
  withMobileDevice {
        view = "mobileList"
  }
 render(view: view, model: [list: listInstance])
}
```

References

- http://www.slideshare.net/bryanrieger/rethinking-the-mobile-web-by-yiibu **(fun and interesting)**
- http://blog.mozilla.com/webdev/2011/05/04/approaches-to-mobile-web-development-part-1-what-is-mobile-friendliness/
- http://www.howtogeek.com/howto/18450/change-the-user-agent-string-in-internet-explorer-8/
- http://useragentstring.com/
- http://thinkvitamin.com/design/beginners-guide-to-responsive-web-design/