Table of Contents

Sometimes it's useful to know the browser name which runs the Vaadin application. It could be useful for showing custom controls or creating responsive layouts.

We have different options to get the name of the browser (user agent). The first one is the recommended option. Simply use your launcher to get the information:

```
getLauncher().getParameter(VaadinUI.PROP_USERAGENT);
```

or

```
getLauncher().getParameter("Web.useragent");
```

If you need more details about the browser, e.g., if it runs on a mobile device or on a table, simply call

```
((VaadinUI)getApplication().getLauncher().getResource()).getAgentInfo()
```

The object will be an instance of `com.handinteractive.mobile.UAgentInfo`. The class was copied from MobileESP and integrated in our UI because we tried to reduce dependencies and it was only one single file.

The class has methods like *detectIpad*, *detectIphone*, *detectSmartphone* but also *detectSonyPlayStation* or *detectXbox*.

If you want to know if the application runs on a tabled or smartphone, simply check

```
getLauncher().getParameter(VaadinUI.PROP_PHONE);
```

or

```
getLauncher().getParameter("Web.mobile.phone");
```

of

```
getLauncher().getParameter(VaadinUI.PROP_TABLET);
```

or

```
getLauncher().getParameter("Web.mobile.tablet");
```

If you need access to the original HttpRequest, simply use HttpContext to get access:

```
HttpContext hctxt = HttpContext.getCurrentInstance();

if (hctxt != null)
{
    Object request = hctxt.getRequest();

    if (request instanceof HttpServletRequest)
    {
        ((HttpServletRequest)request).getHeader("User-Agent");
```

```
    }
}
```

Be careful, it's also possible that the original request was a PortletRequest.

From:
http://doc.sibvisions.com/ - **Documentation**

Permanent link:
http://doc.sibvisions.com/vaadin/useragent

Last update: **2020/07/03 17:06**