Programing guide
Web presentation checker

Adam Kugler

15. prosince 2014

Obsah

1 Introduction

This guide schould help programmers that want to continue working on this tool. This tool is build upon Java EE technology using Maven and uses MySQL database.

2 Development environment

As a team you schould have the most similar development environment as possible. It's up to you which tools you use, because you can't work with our team these days. But for inspiration we used **Netbeans** as our programing environment, **Git** as version control system and **Redmine** for task distribution and monitoring spent time.

2.1 Instaling DE

Because our project is using Maven, it's up to you which environment for Java you choose. We recommend you NetBeans or Eclipse. When you have succesfully installed this environment just open our project as maven project. You can also use Git on GitHub, where you also find how to work with Git.

3 Programing conventions

At first you schould read basic Java conventions.

- You schould write all class names, variable names, etc. in code and all coments in English.
- Use logger for important steps. Write short and meaningful messages.
- Remove logging messages that you used just for debugging.
- Don't use redundant imports.
- Format your code (press ALT+SHIFT+F in NetBeans).

3.1 JavaDoc

- Use the JavaDoc comments for every class and method except setters and getters. You schould comment also unprivate variables.
- It is better to comment definitions rather than implementations.

- Use links within the comments ({@link class#method}).

4 Used tools and libraries

4.1 Graphviz

Graphviz is open source graph visualization software. Graph visualization is a way of representing structural information as diagrams of abstract graphs and networks. We use it for drawing web presentation graphs.

4.2 CSSBox

CSSBox is a Java library which brings the ability of applying CSS rules on HTML DOM. It is strongly bonded to JStyleParser which is CSS rules parser. Without using it, it would be much more difficult to implement our famous CSS Redundancy Checker.

5 Tool extension

This tool was designed as easy extendable by using interfaces. Don't forget to add dependencies corectly if you are adding new functionality.

5.1 Control extension

If you want to add new control you must decide if it schould be "single page control"or "whole presentation control". Be sure that web crawler give you all files you need, because default web crawler gives you just CSS and HTML documents and is skipping images, etc. If you need to change web crawler read about Replacing part. Don't forget to allow control by implementing *AllowOptionService* interface.

5.1.1 Single page control

Create new package and implement *SinglePageChecker* interface.

5.1.2 Whole presentation control

Create new package and implement *WholePresentationChecker* interface.

5.2 Adding new graph

To add new graph you schould create new *GraphGenerator* implementation and *GraphResult* with unique graph type id. You can use GraphvizUtils to work with *Graphviz*.

5.3 Replacing part

If you want to replace some part of our tool (for example WebCrawler), you can do it by implementing given interface. But be sure, that your class do everything, what it schould do. Especially be very careful about sending asynchronous messages.