Coding Choices

Octopeer for Bitbucket

In this document some of the choices we've made during the course of the project are clarified.

Global code

The trackers we use are injected into the webpage, without a single line of global code, only classes and functions will get injected, but won't get used. We need a few lines of global code to start using the logic contained inside the classes. The global code works like an ignition in a car, "starting up" the code inside the classes and functions.

Global mocks

In order to test properly we need to mock several functions, including some chrome functionality. We discovered that global mocks in separate files affect each other, as this lead to failing tests. Since there is also some code duplication we decided to group all global mocks in a single file.

Keylogger

When logging keyboard events there are multiple events to choose from. *keydown*, *keypress*, or *keyup*. We chose *keyup* event in our keylogger. The benefit of choosing the *keyup* event over the *keypress* is that the event is also triggered when a special key (e.g. Page Up, Control) is pressed. There is also a small drawback, the *keyup* event handles all characters as capitals, regardless of Caps Lock or Shift usage. We thought this was less of a handicap than leaving special keys out though.

Mouse position throttling

When adding a *mouse mouse* event listener it will fire a lot. If data is send to the database every time an event occurs, the database might be flooded. Because of this we decided to throttle the amount of times data is send to a maximum of one time per second. At this points any superfluous data is discarded. We will look into implementing a strategy design pattern for the throttling, so that we may swap it out for a different solution (such as bulk sending messages) easily in the future.