|  |
| --- |
| Blue Husky Programming |
| Wulfkonto Implementation Spec |
| How to implement Wulfkonto in any high-level programming language |

|  |
| --- |
| Kyli Rouge  12-5-2014 |

Table of Contents

[2 The Short of It 3](#_Toc405634931)

[3 Definitions 3](#_Toc405634932)

[3.1 “Characters” 3](#_Toc405634933)

[4 The Server 3](#_Toc405634934)

[4.1 Commands 3](#_Toc405634935)

[4.2 Story Creation 3](#_Toc405634936)

[4.3 Responses 3](#_Toc405634937)

[4.4 File Formats 3](#_Toc405634938)

# The Short of It

Wulfkonto is a non-AI application which algorithmically generates a story, given as many or few inputs as the user cares to give. This document will pontificate on how to create the backend. The front end’s design is of another concern, and can be designed in any way so long as it properly hooks in to the backend.

# Definitions

These are definitions essential to understand before implementing Wulfkonto

## “Characters”

From here on in, “character” will refer to a story character (e.g. Huck Finn), and not a text character (e.g. U+0021 EXCLAMATION MARK (!))

More definitions to come

# The Server

Whether local or on a network, the story-creator should run on a multithreaded server. This gives the local advantage of a well-separated front- and back-end, and thus better perceived performance and easier updating. On a network, this gives the advantage of controlled and guaranteed results, little to no user-end workload, equally easy updating, and several other such features.

## Commands

To be considered

## Story Creation

To be considered

## Responses

To be considered

## File Formats

To be considered