THE UNIVERSITY *of* EDINBURGH
informatics

# A User Interface for adding Machine Learning tools into GitHub

Rumyana Rafailova

# Motivation

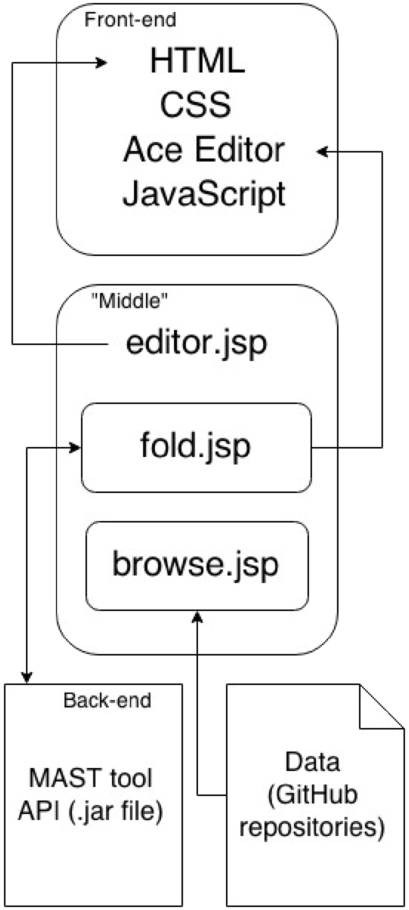
- Understanding a new codebase
  - evaluating open-source software:
    - Algorithms
    - Data Structures
    - Coding standards
  - understanding core functionality at a high level
- Common problem
- Automated tools can
  - Save a lot of time
  - Solve the “chicken-and-egg” problem

# Project Aim

- MAST tool – Machine Learning:
  - Takes some code
  - Produces a summarised version of it – boilerplate blocks folded away
- Project aim: Build a user interface for it

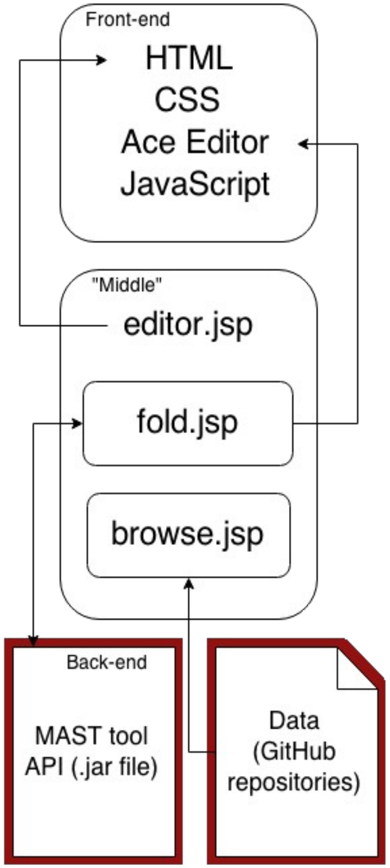
- Feature not currently available in IDEs
  - Except for some trivial decisions (Java imports)
- Manual folding not available while browsing GitHub

# System design - technologies used



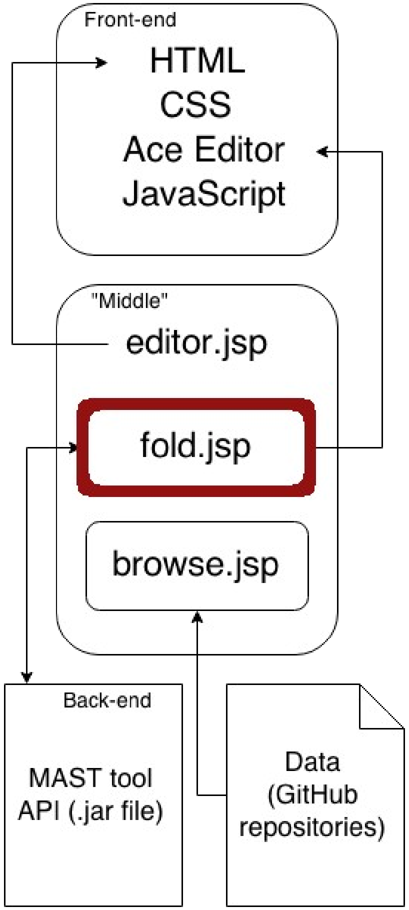
- Front end:
  - HTML/CSS/JavaScript
  - Ace Editor
- Core functionality: JSP/Tomcat
  - communicates with backend
  - builds HTML/JS for frontend
- Back end:
  - MAST tool (jar)
  - Git repositories on the server

# Back end



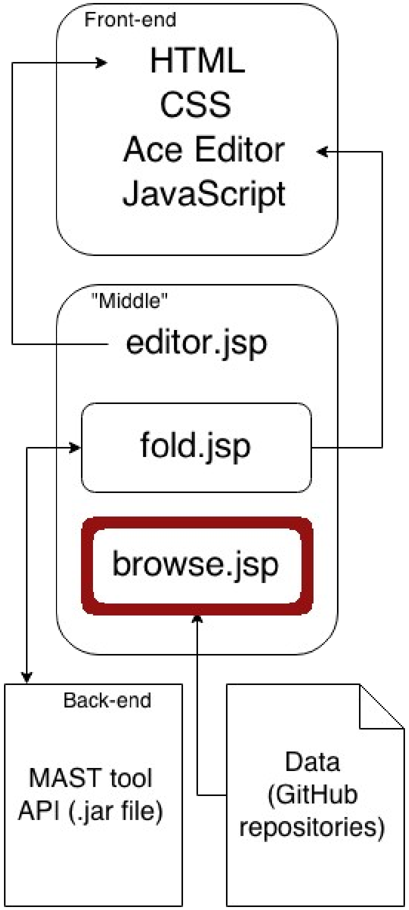
- Popular GitHub repositories cloned on local server
- MAST tool
  - Model
  - Jar file
    - Input: file to fold + compression ratio
    - Output: line numbers to be folded

# Core functionality



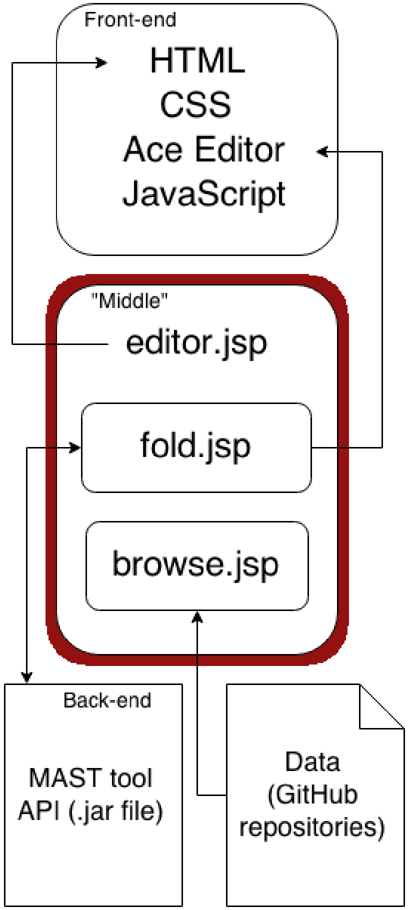
- Backend jar file called by JSP page
- fold.jsp: code editor functionality
  - generates HTML/JS for editor
  - can function separately

# Core functionality



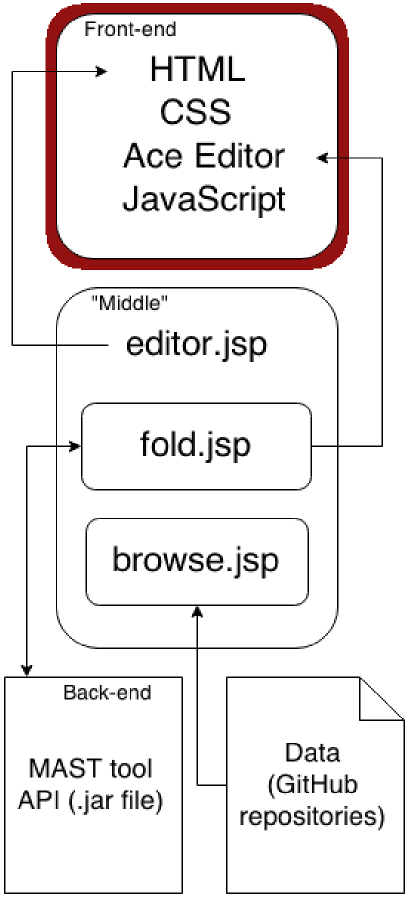
- browse.jsp: file browser functionality
  - recursively traverses server's file tree
  - generates HTML file tree
    - Indented
    - With each scope in its own div
  - custom methods needed because of JSP limitations:
    - Compatibility with Java libraries
    - Internal JSP objects

# Core functionality



- editor.jsp:
  - includes fold.jsp and editor.jsp
  - compiles and executes them as separate servlets
  - compression ratio slider functionality
  - general HTML skeleton

# Front end



- HTML4-compliant, HTML5 for ratio slider
  - with JS for compatibility with older browsers
  - chosen over heavier ways to achieve this
- Ace Editor: code display and folding
  - open-source
  - used by GitHub
  - allowed to extend initial project goals
  - some functionality adapted via JS:
    - feedback button – only front end implemented

# Evaluation

- Usability:
  - simple interface
  - URL rewriting (Tomcat configuration)
- Modularity:
  - browser/editor functionality independent
- Maintainability - possible to
  - extend to more than .java files
  - switch to only showing current project in code browser
  - change appearance
- Portability:
  - Tomcat configuration specific to application, described within its directory

# Evaluation

- Regression testing
  - Test case URLs generated with their expected HTML
  - Compared with the actual server output: wget
  - Tests performed on 3 popular GitHub repositories

- GUI testing
  - Manually throughout implementation
  - On multiple recent versions of Firefox and Opera
    - Testing on more browsers difficult: application isn't hosted

# Thanks!
# Questions?