Columbia OpenAcademy:

ORION

Jonathan Balsano, Kaitlin Huben, Melody Ju
Final Presentation
May 14, 2014

OPEN ACADEMY

What is Orion?

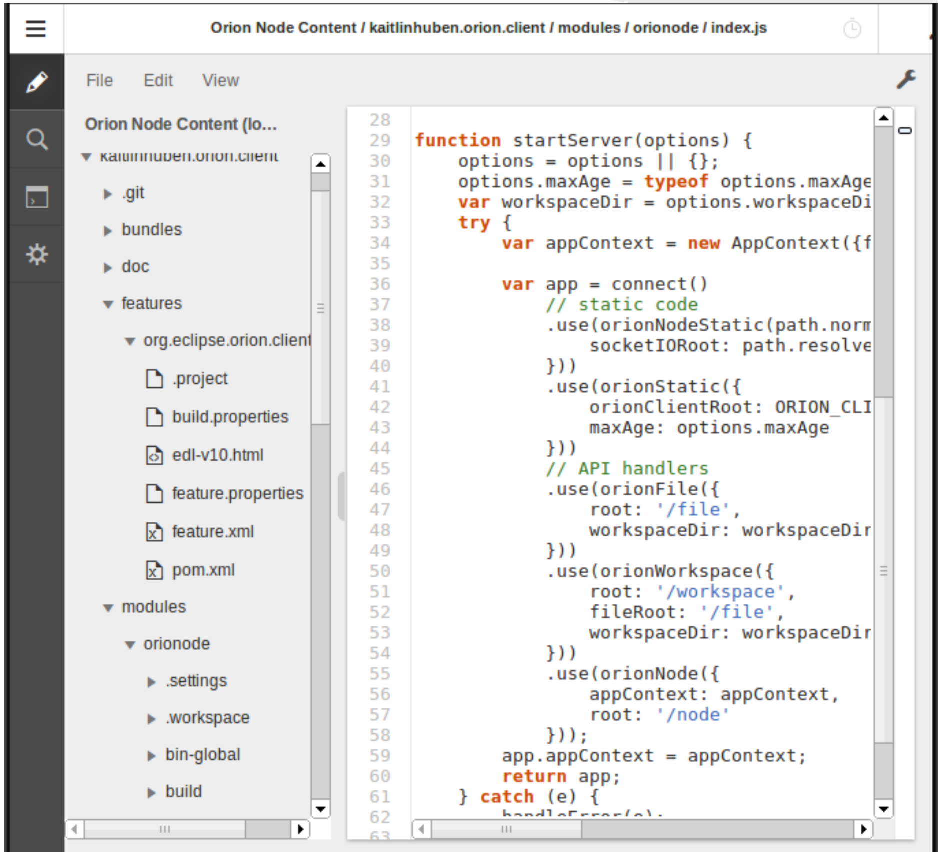
An open source platform for cloud-based development - but not Eclipse in a browser

“Cloud IDEs are like pot. People are interested… curious to be sure… unsure of the side effects… try one for your first time ;-)”

@kwalker during EclipseCon

Overview of Orion Node

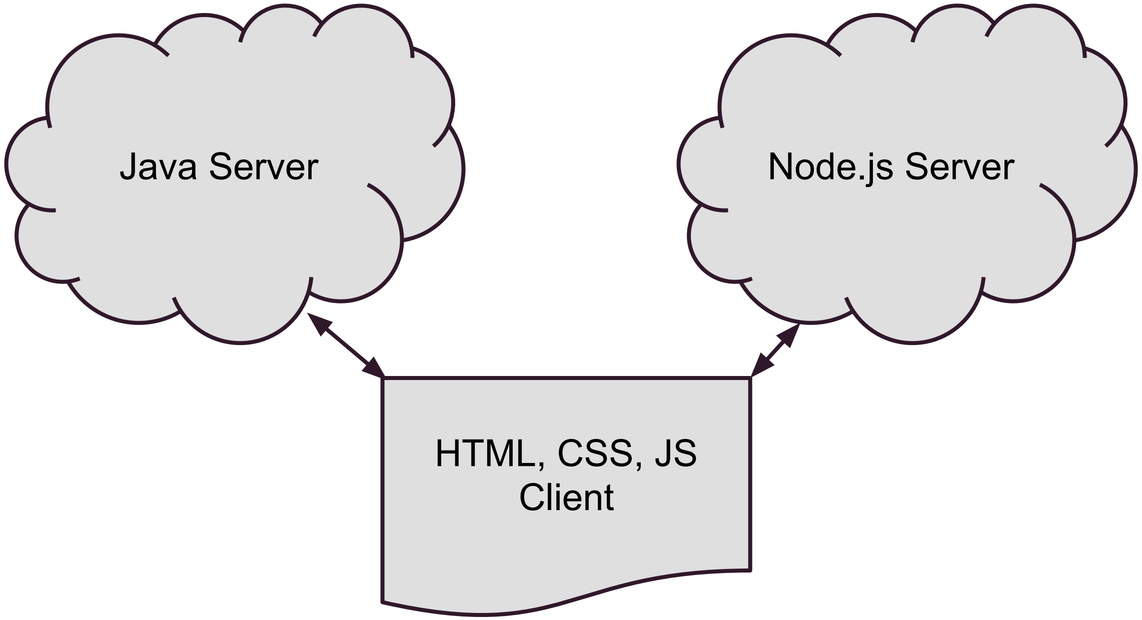
- UI to interact with files on filesystem
- Text editor
  - Syntax Highlighting
  - Vi Mode
  - Auto-save
- Multiple tab support
- “Shell”: node testing and debugging
- Support for managing git repos



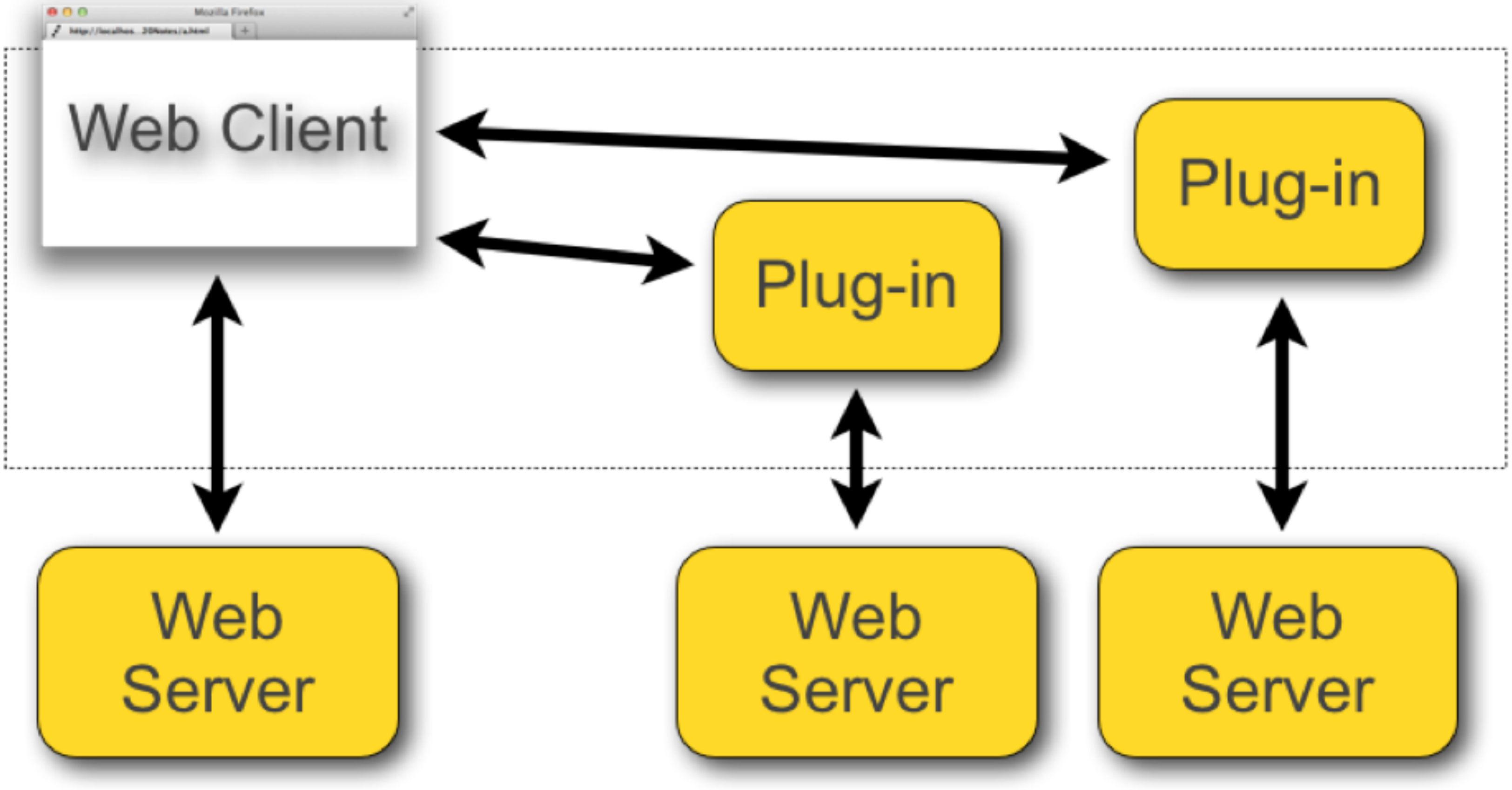
```
function startServer(options) {
    options = options || {};
    options.maxAge = typeof options.maxAge
    var workspaceDir = options.workspaceDi
    try {
        var appContext = new AppContext({f

        var app = connect()
            // static code
            .use(orionNodeStatic(path.norm
                socketIORoot: path.resolve
            }))
            .use(orionStatic({
                orionClientRoot: ORION_CLI
                maxAge: options.maxAge
            }))
            // API handlers
            .use(orionFile({
                root: '/file',
                workspaceDir: workspaceDir
            }))
            .use(orionWorkspace({
                root: '/workspace',
                fileRoot: '/file',
                workspaceDir: workspaceDir
            }))
            .use(orionNode({
                appContext: appContext,
                root: '/node'
            }));
        app.appContext = appContext;
        return app;
    } catch (e) {
        handleError(e);
```

Architecture

- Basic WebApp - Client and Server side
- Original, multi-user server in Java
- Embedded single-user server in Node.js
- A tale of two servers...and two repos



Plugins Architecture



Source: http://www.eclipse.org/community/eclipse_newsletter/2013/june/article2.php

Original goals

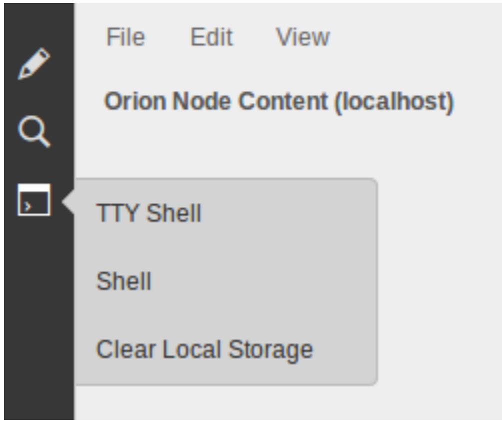
- Strip the old “shell”
  - Provided only commands manually implemented
  - Used “plugin” interface to add commands
  - `cd, clear, edit, help, ls, node, npm, plugins, pwd`, and `service` only
- Replace with browser shell enabling fuller functionality
  - Use OS tty interface to access a real shell
  - bash, zsh, sh

Demo

ORION

Clear Local Storage Plugin

- At Code Sprint, realized changes weren’t showing up
- Needed to clear the HTML5 Local Storage
  - Had to go through Console or browser: `javascript:localStorage.clear()`
- Developed a plugin to facilitate development



Console Architecture

Three Components

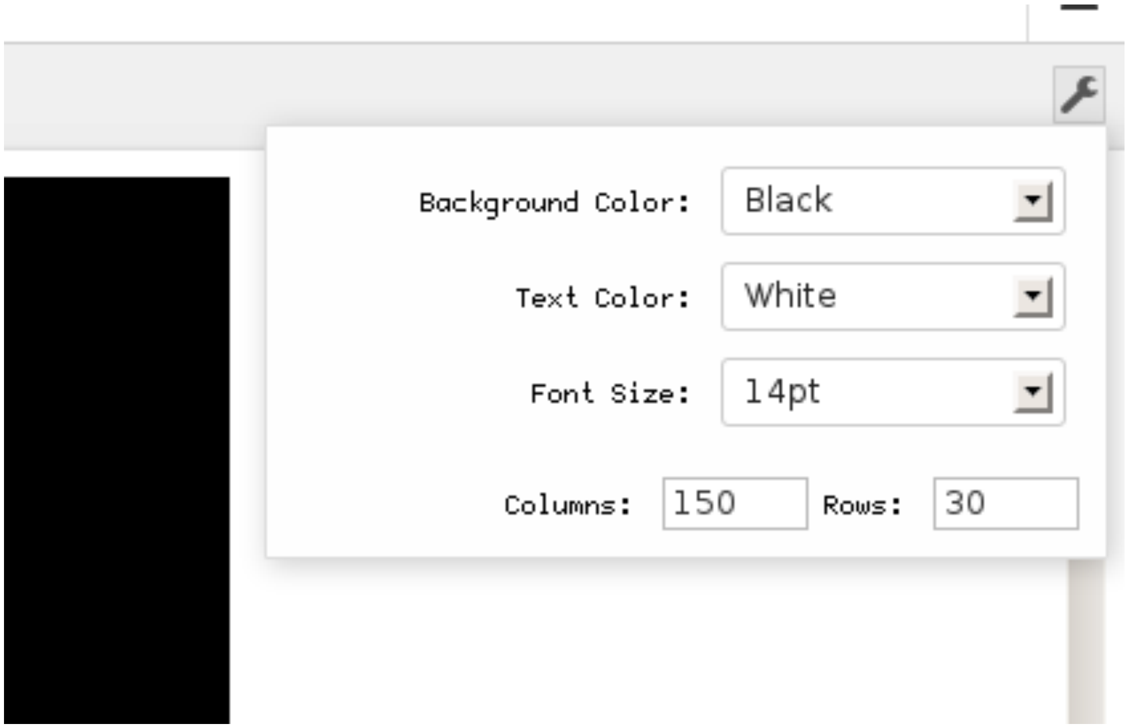
- Plugin is independent
  - Adds links to side bar and drop down menus
- Client side pages
  - Same location as plugin
  - Provide the interactive web page
- Server Handlers
  - Integrated into the server
  - Uses event based loops to respond to requests

Console Implementation

- Client side framework
  - term.js
  - require.js
- Server side framework
  - Socket.IO
  - term.js middleware
  - pty_fork()
- Additional Functionality:
  - Resizing and communicating rows/columns
  - Socket IO namespaces
  - Bringing over the Current Working Directory (Everything’s a Plugin!)

Console User Interface

- Precursor to color scheme personalization
- Consistent UI with the rest of Orion



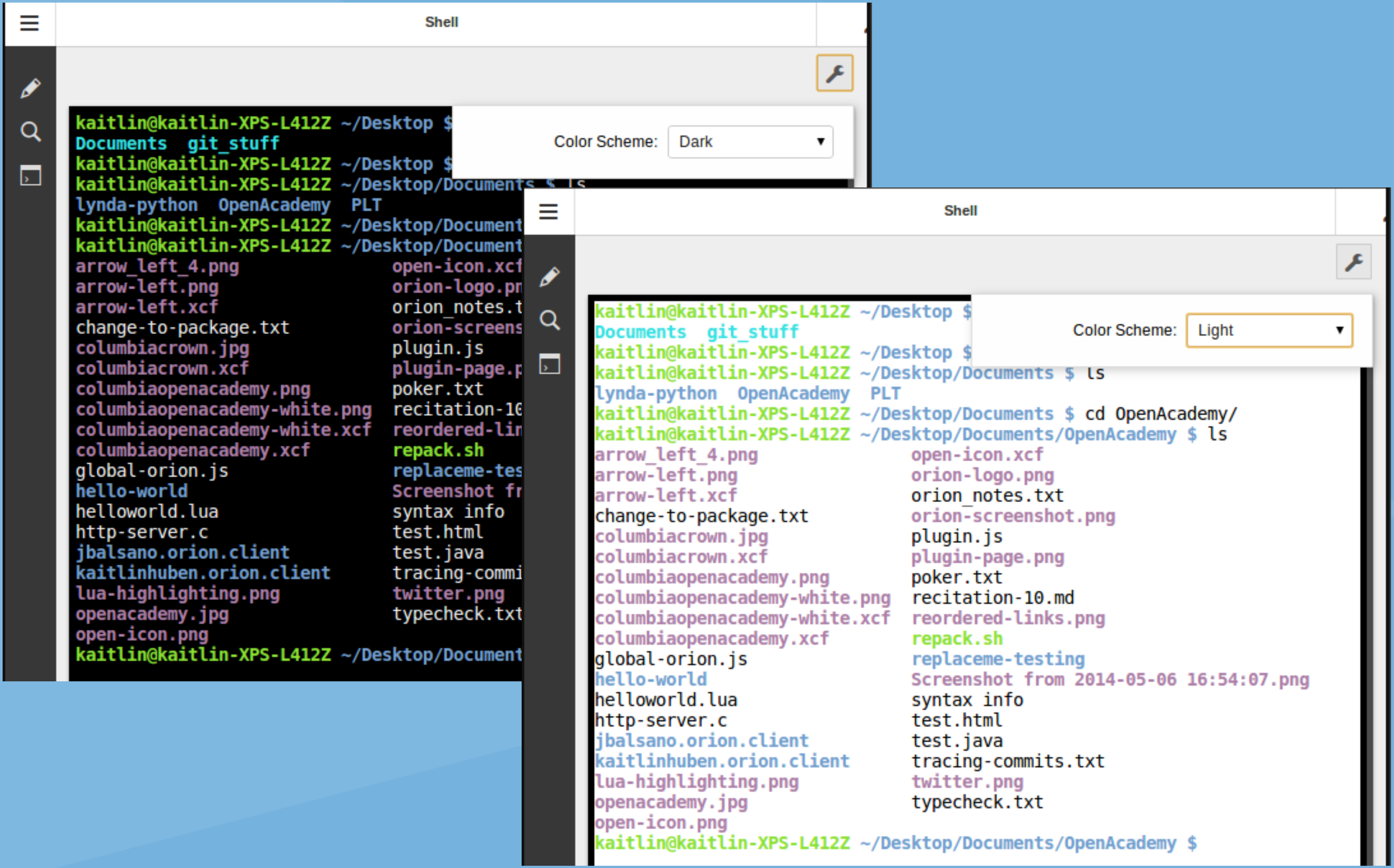
Background Color: Black

Text Color: White

Font Size: 14pt

Columns: 150 Rows: 30

Console Color Schemes



Color Scheme: Dark

```
kaitlin@kaitlin-XPS-L412Z ~/Desktop $
Documents  git_stuff
kaitlin@kaitlin-XPS-L412Z ~/Desktop $
kaitlin@kaitlin-XPS-L412Z ~/Desktop/Documents $ ls
lynda-python  OpenAcademy  PLT
kaitlin@kaitlin-XPS-L412Z ~/Desktop/Document
kaitlin@kaitlin-XPS-L412Z ~/Desktop/Document
arrow_left_4.png               open-icon.xcf
arrow-left.png                 orion-logo.pn
arrow-left.xcf                 orion_notes.t
change-to-package.txt          orion-screens
columbiacrown.jpg              plugin.js
columbiacrown.xcf              plugin-page.p
columbiaopenacademy.png        poker.txt
columbiaopenacademy-white.png  recitation-10
columbiaopenacademy-white.xcf  reordered-lin
columbiaopenacademy.xcf        repack.sh
global-orion.js                replaceme-tes
hello-world                    Screenshot fr
helloworld.lua                 syntax info
http-server.c                  test.html
jbalsano.orion.client          test.java
kaitlinhuben.orion.client      tracing-commi
lua-highlighting.png           twitter.png
openacademy.jpg                typecheck.txt
open-icon.png
kaitlin@kaitlin-XPS-L412Z ~/Desktop/Document
```

Color Scheme: Light

```
kaitlin@kaitlin-XPS-L412Z ~/Desktop $
Documents  git_stuff
kaitlin@kaitlin-XPS-L412Z ~/Desktop $
kaitlin@kaitlin-XPS-L412Z ~/Desktop/Documents $ ls
lynda-python  OpenAcademy  PLT
kaitlin@kaitlin-XPS-L412Z ~/Desktop/Documents $ cd OpenAcademy/
kaitlin@kaitlin-XPS-L412Z ~/Desktop/Documents/OpenAcademy $ ls
arrow_left_4.png               open-icon.xcf
arrow-left.png                 orion-logo.png
arrow-left.xcf                 orion_notes.txt
change-to-package.txt          orion-screenshot.png
columbiacrown.jpg              plugin.js
columbiacrown.xcf              plugin-page.png
columbiaopenacademy.png        poker.txt
columbiaopenacademy-white.png  recitation-10.md
columbiaopenacademy-white.xcf  reordered-links.png
columbiaopenacademy.xcf        repack.sh
global-orion.js                replaceme-testing
hello-world                    Screenshot from 2014-05-06 16:54:07.png
helloworld.lua                 syntax info
http-server.c                  test.html
jbalsano.orion.client          test.java
kaitlinhuben.orion.client      tracing-commits.txt
lua-highlighting.png           twitter.png
openacademy.jpg                typecheck.txt
open-icon.png
kaitlin@kaitlin-XPS-L412Z ~/Desktop/Documents/OpenAcademy $
```

Console Demo

Global install

- Had to change orion.conf to point to the directory you wanted to open, e.g. `workspace=/path/to/workspace`
  - Always had to start server from modules/orionode
- Wanted to implement a global installation
  - `npm install -g orion`
  - Then from any directory, could call:

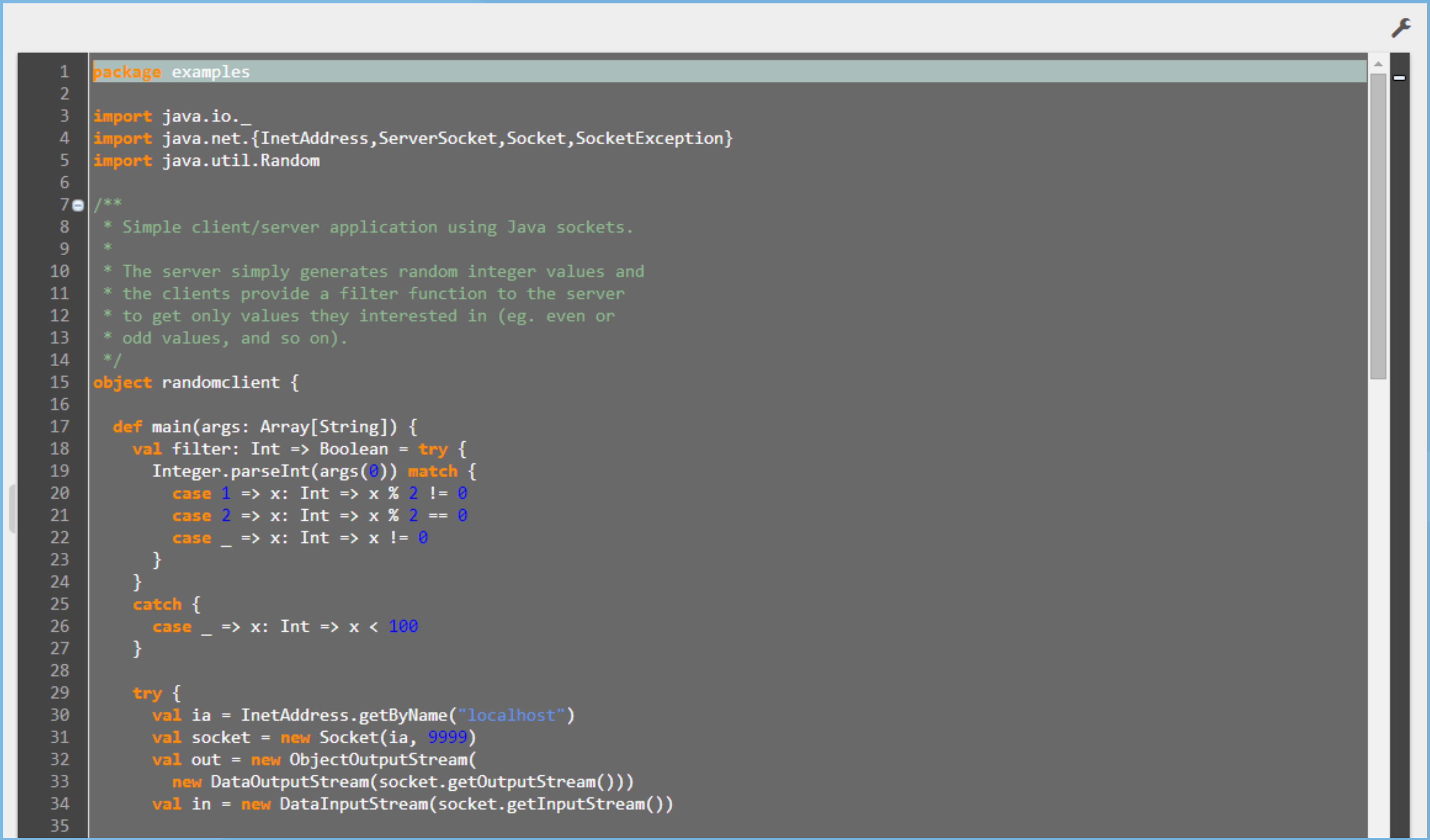
    ```
    $ orion
    $ orion ./
    $ orion this/one/instead
    $ orion ../../how/about/this/one
    ```

Syntax Highlighting

Three steps, three files:

1. Define the language’s grammar
    a. Give the grammar a unique identifier
    b. Register content types that will use this styling
    c. Tell the editor which language constructs to recognize
2. Add the grammar to Orion...by writing a(nother) plugin
    a. Register content type
    b. Register syntax styling
3. Add plugin to defaults

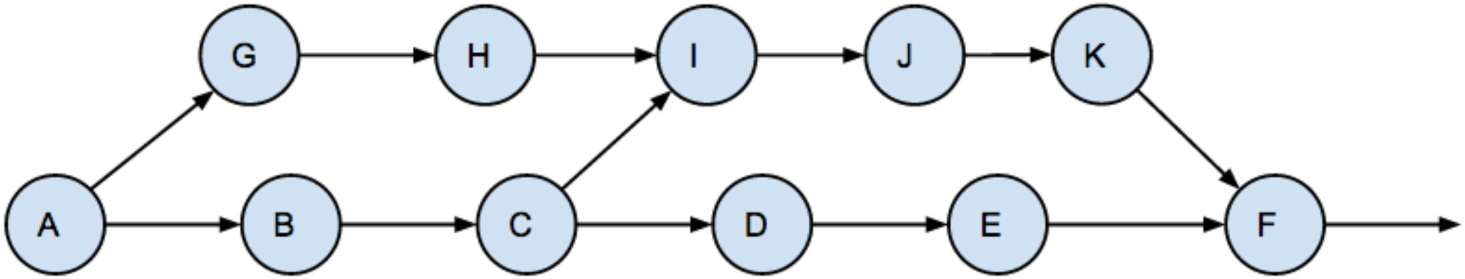
Syntax Highlighting



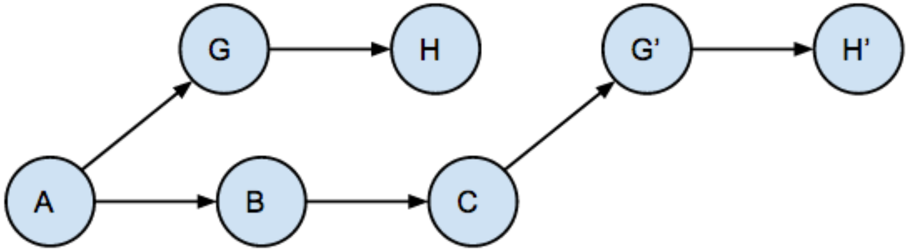
```
package examples

import java.io._
import java.net.{InetAddress,ServerSocket,Socket,SocketException}
import java.util.Random

/**
 * Simple client/server application using Java sockets.
 *
 * The server simply generates random integer values and
 * the clients provide a filter function to the server
 * to get only values they interested in (eg. even or
 * odd values, and so on).
 */
object randomclient {

  def main(args: Array[String]) {
    val filter: Int => Boolean = try {
      Integer.parseInt(args(0)) match {
        case 1 => x: Int => x % 2 != 0
        case 2 => x: Int => x % 2 == 0
        case _ => x: Int => x != 0
      }
    }
    catch {
      case _ => x: Int => x < 100
    }

    try {
      val ia = InetAddress.getByName("localhost")
      val socket = new Socket(ia, 9999)
      val out = new ObjectOutputStream(
        new DataOutputStream(socket.getOutputStream()))
      val in = new DataInputStream(socket.getInputStream())

```

Git: Rebasing

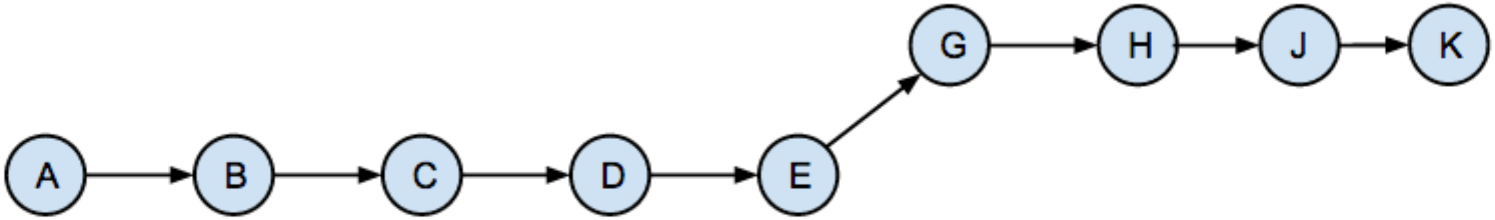
Branch



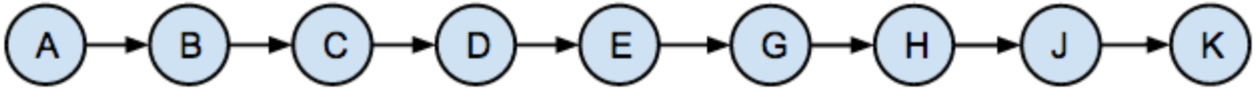
Rebasing



Rebased



Re-aligned



Git: Rebasing

- Re-applies each commit (changeset)
- Creates a new commit
  - Github doesn’t like this
- Requires force pushing
  - Coworkers don’t like this
- Provides opportunity for squashing
  - Eclipse likes this...

Git: Scripting

- Eclipse Foundation requires CLA verification
- All commits must follow format:
  - Bug #XXX - Full terminal-based console
  - Actual commit message
  - Signed-off-by: Your Name <yourCLA@email.com>

Enter git filter-branch

Git: Scripting

```
git filter-branch -f --msg-filter '
   if [ "$GIT_AUTHOR_NAME" = "$AUTHOR" ]; then
     echo "$BUG" && echo && cat && echo && echo \
           "Signed-off-by: $AUTHOR <$CLA_EMAIL>"
   else
     cat
   fi' \
   $EXCL_OLDEST...$NEWEST
```

Check us out!

Repos:

- ➔ Eclipse: https://github.com/eclipse/orion.client
- ➔ Jon (TTY shell, color schemes, syntax highlighting): https://github.com/jrbalsano/orion.client
- ➔ Kaitlin (Global installation, Lua highlighting): https://github.com/kaitlinhuben/orion.client

Blog:

http://orionprojectcu.wordpress.com/