

The Java Network API and Parsing HTML

`java.net.*`

`javax.swing.text.html.*`

Administrivia

- MondoHashMap due today
 - Late time counting now...
- Milestone 2 due Feb 9 (next Wed), start of class
 - 8 days...
- Office hours on Wed truncated
 - 9:00-10:15 AM
 - Or send mail for another time

Design exercise

- **Given:** A `LinkedList` class, with methods `size()`, `prepend()`, `append()`, `first()`, `last()`, `iterator()`, `removeFirst()`, and `removeLast()`
- **Design:** A sub-class, `StackGenList`, that supports a method `getStackView()` which returns a `Stack` object that, in turn, supports `push()`, `pop()`, `topElement()`, and `depth()`
- **Constraints:**
 - You *do not* have access to the internals of the `LinkedList` (it has no protected members)
 - You *may not* copy any of the `LinkedList` data
 - The `Stack` view and the `LinkedList` must always remain synchronized

M2 timeline

- Today: understand `java.net.URL`,
`javax.swing.html.HTMLEditorKit`, and
`javax.swing.html.HTMLEditorKit.ParserCallback`
- Tomorrow: write your own web page fetch and
parse demo program(s); start web spider
- Wed-Fri: extend spider, work on coverage; cycle
detection; filtering
- By Fri: be able to spider CS web site; build and save
REVERSE INDEX
- Weekend: testing; debugging; refinement
- Early next week: performance testing and
documentation

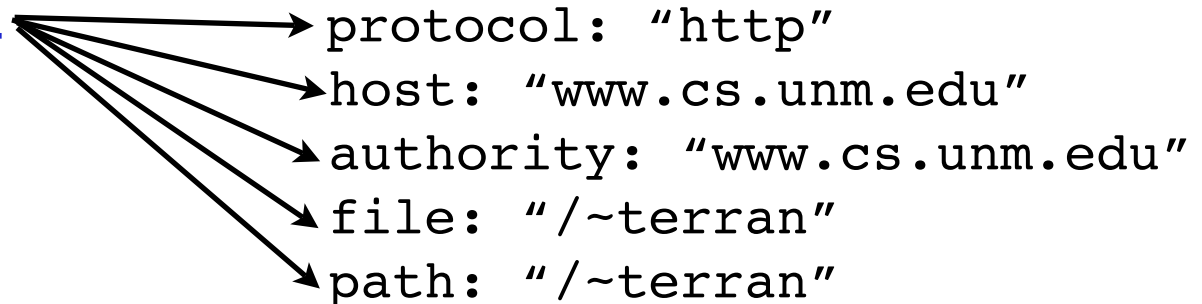
Fetching a web page

- Core utility: `java.net.URL`
 - Abstract representation of resource name and location
 - Parses and stores parts of URL
 - Brokers different protocol types
- Accessing content: `java.net.URLConnection`
 - Opens, tracks, closes network connections
 - Provides content data
 - Provides meta-data

The web fetch cycle

```
URL u=new URL("http://www.cs.unm.edu/~terran/");
```

Breaks out parts of URL



Sets protocol handler

handler: `HttpHandler`

```
URLConnection con=u.openConnection()
```

Generates protocol-specific connection handler

```
return new handler.openConnection(this)
```

```
return new HttpURLConnection(this)
```

```
con.connect()
```

Opens socket connection to host

```
con.getHeaderFields()
```

Requests metadata

```
con.getInputStream()
```

Requests resource content; returns stream view of data