Automating and Customizing the Web

Michael Bolin
blog.bolinfest.com

http://bit.ly/5XNo5O

Outline

- Background
- Chickenfoot API with Examples
- Putting it all together
- Exercises

What would we like to be able to do with Web Applications?

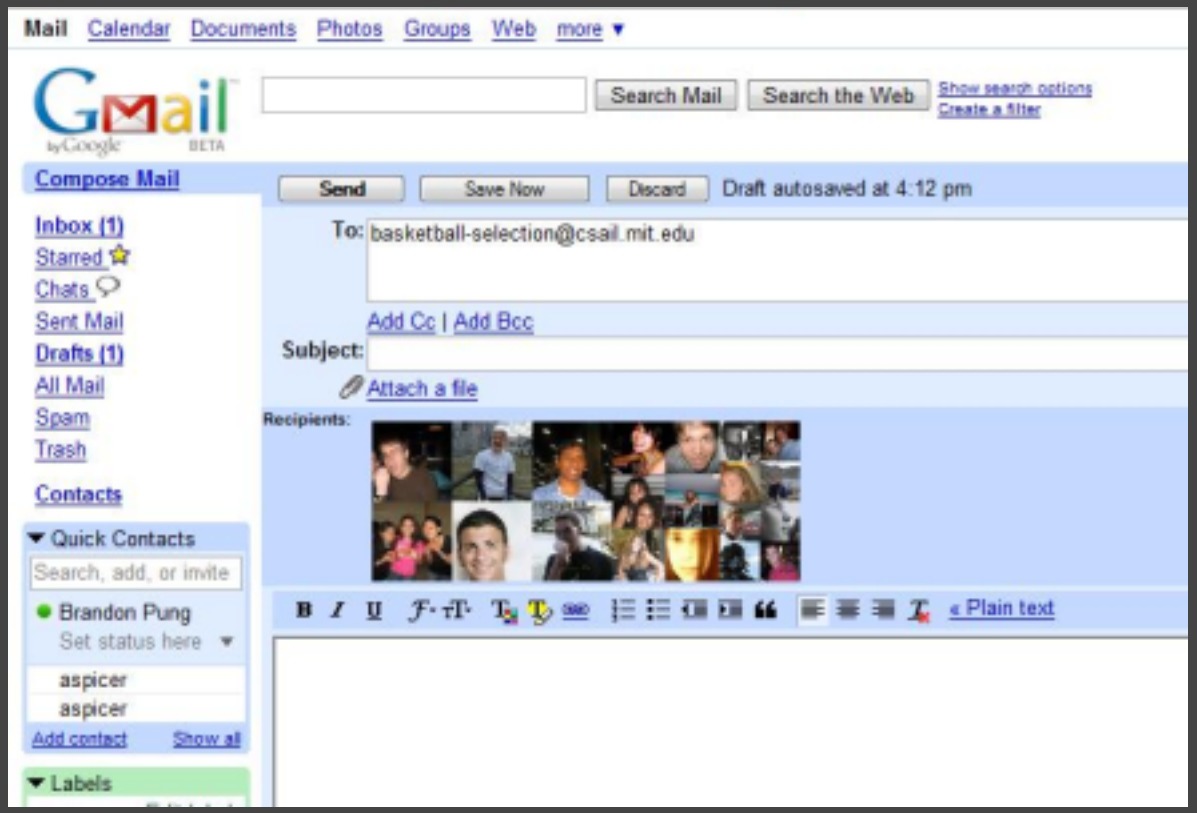
- Automate repetitive operations
  - Example: Fill in defaults on a web form
  - Example: Scrape data from gasbuddy.com to monitor local gas price trends
- Transform appearance of a web page
  - Example: Change color schemes for better contrast
  - Example: Remove unnecessary or distracting features
- Integrate multiple web sites (i.e., mashups)
  - Example: housingmaps.com
  - Example: calmap.bolinfest.com

Design Principles

- The goal of Chickenfoot is to enable end-user programming on the Web
- Users may need to know HTML and CSS, but they should not have to be familiar with the HTML and CSS of the page
- Users should be able to share scripts with each other:
  - Scripts are JavaScript files, so they are easy to send and edit
  - Chickenfoot web site has a wiki for scripts
  - Chickenfoot script can be packaged as a Firefox extension
  - IBM has a similar internal system called Koala that combines these features

Kangaroo

http://groups.csail.mit.edu/uid/kangaroo/



Install Chickenfoot

Install Chickenfoot from the Chickenfoot web site:

http://groups.csail.mit.edu/uid/chickenfoot/install.html

Let's take a tour of the Chickenfoot UI:

- Script editor
- Output pane
- Patterns pane
- Triggers pane

Chickenfoot API

- Page navigation
- Input automation
- Pattern matching
- Page modificatoin
- Widgets and input handling

Comprehensive list on Chickenfoot web site:
http://groups.csail.mit.edu/uid/chickenfoot/api.html

Page Navigation

go(url)

```
go("google.com"); // goes to Google
go("http://mail.google.com"); // goes to Gmail
```

Note that the Chickenfoot script suspends until the page has finished loading

openTab(url)

```
openTab("google.com"); // opens in a new tab
```

Returns a "window" object that can be used with the with keyword in JavaScript:

```
with (openTab('google.com')) {
  click('news link');
}
```

Input automation

enter([textbox,] value)

```
enter("my query"); // only one textbox
enter("username", "bolinfest");
```

click(button_or_link)

```
click("first search"); // button or link
click("search button"); // button only
```

check(checkbox); uncheck(checkbox)

```
check("stay signed in");
```

pick([listbox,] choice)

```
pick("language", "English");
```

Pattern Matching

find(pattern)

```
var table = find("first table");
for each (var row = table.find('row')) {
  output(row.find('2nd cell').text);
}
```

find() returns an object of type "Match" which has the following API:

- find(): does a find within the context of the match
- text: text content in the match
- html: HTML content in the match
- element: HTML Element of the match
- next: next Match in the sequence

Page modification

insert(position, chunk)

```
insert(after("about link"), "Hi!");
```

remove(pattern)

```
remove("feeling lucky button");
```

replace(pattern, chunk)

```
replace("first image",
"<img src='http://bolinfest.com/ninja.png'>");
```

This is an example where the end-user should know HTML, but does not need to know the HTML of the page.

Widgets and input handling

Button(label, action); Link(label, action)

```
insert(after("feeling lucky button"),
  new Button("please click me",
  function () { alert("clicked!"); }));
```

onClick(pattern, action)

```
onClick("first image",
  function () { alert("clicked!"); });
```

Note that both Button and Link qualify as "chunks," which means they can be used with insert() and replace().

Putting it all together

Demo: Icon Search

Complete Icon Search Script

```
function iconSearch() {
  var query = find('textbox').element.value;
  click('images');
  click('advanced');
  enter('all of the words', query);
  pick('icon');
  pick('black and white');
  click('google search');
}

insert(after('feeling lucky'),
new Button('Icon Search', iconSearch));
```

Resources

Chickenfoot web site:
http://groups.csail.mit.edu/uid/chickenfoot/

"Get Started With Chickenfoot" on webmonkey.com:
http://www.webmonkey.com/tutorial/Get_Started_With_Chickenfoot

Questions?

Exercises

- Script that goes to orbitz.com and looks up a one-way ticket from JFK to SFO on 11/23/09 for 2 adults
- Add a link that appears on google.com to kick off the previous script
- prompt() the user about the number of travellers instead of hard-coding it to 2
- Instead of showing a list of files when visiting http://bolinfest.com/targetalert-images/ show a list of images
- Implement TargetAlert
- When on a page on Amazon.com that shows a book, add a link to the NYU library
  - Add a button or link that will take you to the web page for the book in the NYU library

Solutions

Don't cheat!

orbitz.com

```
go('orbitz.com');
pick('one-way');
enter('from', 'JFK');
enter('to city', 'SFO')
enter('leave', '11/23/09');
pick('adult', '2');
click('find flights');
```

orbitz.com from google.com

```
function orbitz() {
  go('orbitz.com');
  pick('one-way');
  enter('from', 'JFK');
  enter('to city', 'SFO')
  enter('leave', '11/23/09');
  pick('adult', '2');
  click('find flights');
}

insert(after('advanced search'),
new Link('Check flight on orbitz', orbitz));
```

orbitz.com with prompt

```
go('orbitz.com');
pick('one-way');
enter('from', 'JFK');
enter('to city', 'SFO')
enter('leave', '11/23/09');
var num = prompt('How many travellers?', '2');
pick('adult', num);
click('find flights');
```

List of images

```
for each (var match in find('link')) {
  var href = match.element.href;
  replace(match, '<img src="' + href + '>');
}
```