

# Rich Internet Applications

6. Ajax functionality



---

---

---

---

---


---

---

---

## Ajax patterns in this lecture

- This lecture refers to the following patterns:
- **Functionality and Usability Patterns**
  - *Widgets*
    - **Progress Indicator, Suggestion, Live Form**
  - *Page Architecture*
    - **Drag and Drop, Virtual Workspace**
  - *Visual Effects*
    - **One-Second Mutation**
- *Functionality*
  - **Unique URLs**



Rich Internet Applications 6. Ajax functionality #2

---

---

---

---

---


---

---

---

## Ajax patterns in this lecture

- **Programming Patterns**
  - *Browser-Server Dialogue*
    - **Periodic Refresh**



Rich Internet Applications 6. Ajax functionality #3

---

---

---

---

---

---

---

---

### Widgets



- A widget is a component of a graphical user interface with which a user interacts
- HTML widgets are very basic
- Many widgets are now available which replicate the functionality of desktop application toolkits
- Toolkits, e.g. jQuery UI, and individual widgets for specific purposes




---

---

---

---

---

---

---

---

### Progress Indicator pattern



- **Problem**
  - How can you provide feedback while waiting for server responses?
- **Forces**
  - Ajax Apps often make XMLHttpRequest Calls to query and update the server
  - To provide good feedback, you should provide an idea of which page elements are affected by any update
  - Users like to feel in control, and that control is taken away when the user is waiting for a response
- **Solution**
  - Indicate the progress of server calls. You can't always reduce delay, but you can include a Progress Indicator to ease the pain




---

---

---

---

---




---

---

---

### Progress Indicator



- For shorter delays, typical Progress Indicators include:
  - A small message like "Updating document"
  - An animated icon   
- For longer delays, the following can be used:
  - A meter showing how much progress has been made
  - An estimate of time remaining
  - A sequence of messages indicating what's happening at present




---

---

---

---

---

---

---

---

### Jakob Nielsen's guidelines



- Delay less than 0.1 second:
  - the user will feel it's instantaneous. No feedback necessary.
- Delay between 0.1 second and 1 second:
  - the user will notice it but it won't break their flow of thought. No feedback necessary
- Delay between 1 and 10 seconds:
  - the user's flow of thought is interrupted as he awaits the response. Basic feedback necessary; i.e., an indication that a delay is occurring
- Delay greater than 10 seconds:
  - the user will want to proceed to other tasks. Detailed feedback necessary



Rich Internet Applications

6. Ajax functionality #7

---

---

---

---

---

---

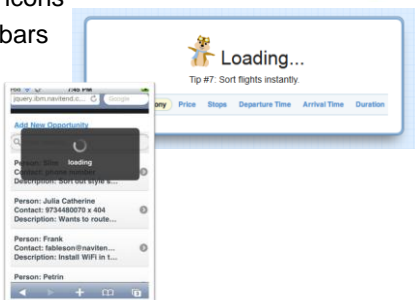
---

---

### Implementing progress indicators



- Animated icons
- Progress bars
- Pop-ups



Rich Internet Applications

6. Ajax functionality #8

---

---

---

---

---

---

---

---

### Suggestion Pattern



- **Problem**
  - How can you improve throughput?
- **Forces**
  - When presented with a free text area, people don't always know what they're meant to type in.
  - Though many users are now quick on the keyboard, there are still many users for whom the Web is a click-mostly experience.
  - Typing speed remains a bottleneck even for fast typists, most of whom think faster than they can type.
  - People make mistakes when they type.



Rich Internet Applications

6. Ajax functionality #9

---

---

---

---

---

---

---

---

## Suggestion Pattern



- **Solution**
  - Suggest words or phrases that are likely to complete what the user is typing
  - The user can then select the Suggestion and avoid typing it in full.
  - The results usually come from an XMLHttpRequest Call
  - The partial input is uploaded and the server responds with a collection of likely matches
  - Main benefit is to offer a constrained set of choices—when there are more choices than would fit in a standard option list, but still a fixed set
  - For Google Suggest, it's a list of choices that are considered probable based on search history

---

---

---

---

---

---

---

---

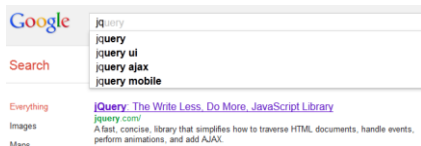
---

---

## Suggestion example



- Google Suggest used in Google Toolbar and by other sites, e.g. **YouTube**
- Further refined as Instant Search which returns full page of search results with each Ajax call




---

---

---

---

---

---

---

---

---

---

## Live Form Pattern



- **Problem**
  - How can the user submit data as quickly as possible?
- **Forces**
  - Most data submission tasks require some flexibility. A particular answer for one item may necessitate further questions.
  - Most data submission requires server-side validation.
  - Users get frustrated waiting for data to be validated and continuously refining data upon each explicit submission
- **Solution**
  - Validate and modify a form throughout the entire interaction instead of waiting for an explicit submission
  - Each significant user event results in some browser-side processing, often leading to an XMLHttpRequest Call. The form may then be modified as a result.

---

---

---

---

---

---

---

---

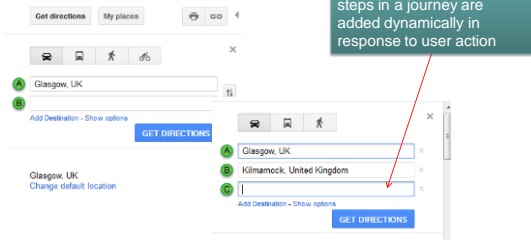
---

---

### Live Form example



#### • Google maps directions



text boxes for additional steps in a journey are added dynamically in response to user action

---

---

---

---

---

---

---

---

### Drag and Drop pattern



#### • Problem

- How can users rearrange objects on the page?

#### • Forces

- Object relationships are important and the user's task often involves changing those relationships.
- Ajax Apps often represent object relationships visually.
- Rearranging the visual structure from a separate form is messy

#### • Solution

- Provide a Drag-And-Drop mechanism to let users directly rearrange elements on the page
- Drag-And-Drop has proven itself to be a powerful control mechanism in conventional desktop applications; it is certainly achievable using standard web technologies

---

---

---

---

---

---

---

---

### Drag and Drop example



#### • GCULearn!




---

---

---

---

---

---

---

---

### Implementing drag and drop



- Supported by toolkits, e.g. jQuery UI Draggable/Droppable
- Native support in HTML 5
- Spec defines an event-based mechanism, JavaScript API, and additional markup for declaring that just about any type of element be draggable on a page




---

---

---

---

---

---

---

---

### Virtual Workspace pattern



- **Problem**
  - How can the user navigate through a large workspace?
- **Forces**
  - Ajax Apps often require data from the server.
  - On the server side, data storage is virtually unlimited
  - The application should respond to data requests as quickly as possible.
  - Bandwidth constraints mean it's not feasible for an interactive application to download the entire search result. Yet users should be free to explore the whole thing.




---

---

---

---

---

---

---

---

### Virtual Workspace pattern



- **Solution**
  - Provide a browser-side view into a server-side workspace, allowing users to navigate the entire workspace as if it were held locally
  - The illusion is that the entire workspace is already in the browser, but the reality is that the server actually provides content on demand
  - At any time, the user is looking at an "opening" or "portal" into the entire workspace




---

---

---

---

---

---

---

---

### Navigable workspaces



- Lists
- Tables
- Physical images
- Documents
- 3D Models

---

---

---

---

---

---

---

---

### Navigation



- Scrollbars
- Dragging tool
- Keyboard shortcuts
- Mouse wheel
- Direct command
- Zoom slider
- Selecting a region

---

---

---

---

---

---

---

---

### Virtual Workspace examples



Twitter website shows tweets in an endless scrolling list with **no paging** – retrieves first few results initially then loads next few with an XHR when user scrolls to bottom of list

```

Sort by key
next_page
state
0
1
retweeted
in_reply_to_status_id
[{"retweeted":false, "user":{...}, "retweet_co
in_reply_to_status_id=173868607393107970, "retweet
truncated":false, "more...": "38 more..."}]
Object { retweeted=false, user={...}, retweet_count
Object { retweeted=false, in_reply_to_status_id=17:
false
173868607393107970

```

---

---

---

---


---

---

---

---

### Virtual Workspace examples



Google Maps loads map as a set of 256x256 tiles – only loads new tiles as required when the user moves the Viewport

navigation includes dragging, controls and mouse wheel to zoom

Rich Internet Applications 6. Ajax functionality #22

---

---

---

---

---

---

---

---

### One-Second Mutation pattern

- Problem**
  - There's a lot of activity in an Ajax App; how can you help the user digest it all?
- Forces**
  - With Ajax, it's possible to update only the portion of the screen that has changed; if nothing has changed, the display remains exactly as is
  - The screen can get cluttered with a lot of information, much of which could change at any time
  - While human vision is good at spotting changes, it's easy to miss a sudden change, especially if the change is a subtle one
  - There are different types of changes, and those differences are important to users

Rich Internet Applications 6. Ajax functionality #23

---

---

---

---

---

---

---

---

### One-Second Mutation pattern

- Solution**
  - When a page element undergoes a value change or some other significant event, dynamically mutate its shape for about a second
  - In most cases, the mutation is used to reflect a permanent change. Instead of flipping from one form to another, the element gradually transitions into another form. The effect serves a few purposes:
    - The mutation and the extended transition time make it likely that the user will notice that something has happened
    - The type of mutation hints at what sort of action has taken place
    - The gradual shift in appearance feels more natural than an instant transformation

Rich Internet Applications 6. Ajax functionality #24

---

---

---

---

---

---

---

---

### Mutations



- Slide Out
- Wipe
- Reveal
- Disappear
- Slide In
- Cover Up
- Grow
- Shrink
- Metamorphise



Rich Internet Applications

6. Ajax functionality #25

---

---

---

---

---

---

---

---

---

---

### Related animation patterns



- One-Second Spotlight
  - Fade-in, fade-out, materialise
  - Can be used to draw attention to popups
- One-Second Motion
  - Move a page element from one place to another
  - Motion helps user notice the change of position



Rich Internet Applications

6. Ajax functionality #26

---

---

---

---

---

---

---

---

---

---

### One-Second Mutation/Spotlight example



- BBC live sports

Live Reporting

Reporting by Colin Miller BBC Scotland

PREVIOUS RESULTS

TODAY'S

St Mirren 1 - 1 Aberdeen 10:00

Celtic v Motherwell 10:00

Live Reporting

Reporting by Colin Miller BBC Scotland

PREVIOUS RESULTS

TODAY'S

St Mirren 1 - 1 Aberdeen 10:00

Celtic v Motherwell 10:00

Hearts v Dundee Utd 10:00

Rangers v Aberdeen 10:00

St Johnstone v Dunfermline 10:00

Upcoming Fixtures

As It Stands Table

Scores are boldened briefly when they change

Previous items slide down and new item fades in



Rich Internet Applications

6. Ajax functionality #27

---

---

---

---

---

---

---

---

---

---

### Implementing animation



- Animation is relatively simple to implement in JavaScript
- Most toolkits offer a variety of motion and animation options to make it even simpler
- jQuery has range of effects which are easy to implement

---

---

---

---

---

---

---

---

### Unique URLs pattern



- **Problem**
  - How can you assign distinct URLs to different states of your application?
- **Forces**
  - Expressive URLs are vital. The ability to link from one site to another is arguably what made the web successful
  - Browsers retain URLs in their history, and the Back Button will only work if URLs vary
  - A single address for your whole site is not enough - each distinct component of a website should have its own URL
  - In conventional websites, the URL changes when the page is refreshed - in Ajax Apps, the page never refreshes, so the browser won't change the URL

---

---

---

---

---

---

---

---

### Unique URLs pattern



- **Solution**
  - Provide Unique URLs for significant application states. Each time a significant state change occurs - e.g. the user opens a new product in an E-Commerce system - the application's URL changes accordingly
  - Since "XMLHttpRequest Call"s don't affect the page URL, the URL is by default permanently stuck on the same URL that was used to launch the Ajax App, no matter how many transfers occur
  - The only option you have to manipulate the URL is with Javascript

---

---

---

---

---

---

---

---

### Fragment identifiers



- Fragment identifier, or hash, has been used traditionally for internal links to locations with a page
- e.g. mypage.html#top, mypage.html#item1
- Browser changes URL in address bar when an internal link is clicked
- URL is included in browser history so Back button works with internal links




---

---

---

---

---

---

---

---

### Setting the fragment identifier from JavaScript



- Simply set the value of window.location.hash
- To create and recognise unique URLs you need to:
  - Set window.location.hash to a meaningful value when user changes state of a page
  - Read the window.location.hash value on page load and set state accordingly




---

---

---

---

---

---

---

---

### The Back button



- Unique URLs are not enough
- If you change the state of a page several times, the URL will now change each time
- Clicking the Back button will access the previous URL
- But, the page doesn't reload, so the code you use to change the state to reflect the URL will not run




---

---

---

---

---

---

---

---

## Workaround – polling the hash

- If you set the function which updates state depending on the URL (or on the hash) to run repeatedly, it can update the state when the URL changes
- Polling the hash
- Polling interval of about a second makes the Back button appear to behave in the traditional way
- A rather awkward solution, but it does work...



Rich Internet Applications

6. Ajax functionality #34

---

---

---

---

---

---

---

---

## Browser differences

- ...in Firefox, but not in IE
- IE's history behaves differently
- IE needs a different solution involving a hidden iFrame...
- ...which doesn't work in Firefox
- Need to write code to check browser and behave accordingly



Rich Internet Applications

6. Ajax functionality #35

---

---

---

---

---

---

---

---

## Unique URLs example

- This page loads a list of package names into a select
- Shows details of the selected package when the select value changes

when selected package changes, get data and set hash to change URL

```
var url = "packageinfo_json.aspx?name=" + escape(packageSelect.value);
window.location.hash = escape(packageSelect.value);
```

when page loads, initialise state and repeat initialisation of state repeatedly

```
initializeStateFromURL( );
setInterval(initializeStateFromURL, 1000);
```



Rich Internet Applications

6. Ajax functionality #36

---

---

---

---

---

---

---

---



## HTML 5 History API



- Provides native support for history management
- **pushState** method
- **replaceState** method
- **window.state** property
- **popstate** event
- Not currently supported by all browsers and some incompatibilities between browsers which do



Rich Internet Applications

6. Ajax functionality #40

---

---

---

---

---

---

---

---

## Cross-browser history



- Various libraries abstract browser differences in browser support for history management, for example:
- jquery.history.js
- History.js
  - Follows native HTML 5 API where possible
  - provides a cross-compatible experience for all HTML5 Browsers
  - provides a backwards-compatible experience for all HTML4 Browsers using a hash-fallback



Rich Internet Applications

6. Ajax functionality #41

---

---

---

---

---

---

---

---

## History and unique URLs in the wild



- Not all RIA sites handle unique URLs and the back button
- Google Maps does not change the URL when you change the viewport
- Twitter does not change the URL when you scroll down the search results
- In these cases it is probably obvious to the user how to undo an action, by moving back the map or scrolling up the list



Rich Internet Applications

6. Ajax functionality #42

---

---

---

---

---

---

---

---

### Real time Ajax applications



- In most Ajax apps information should be returned from the server as quickly as possible once it is requested
- However, some applications require information to be displayed on a page whenever it becomes available
- For example, chat applications, live news feeds
- These are known as **push** applications




---

---

---

---

---

---

---

---

### Comet



- Ajax applications – client requests data
- Comet applications – server pushes data to the client whenever new data is available
- Name coined by Alex Russell (Dojo):
  - <http://alex.dojotoolkit.org/2006/03/comet-low-latency-data-for-the-browser/>
- Not a specific technology
- Term describes “event-driven, server-push data streaming” programming style




---

---

---

---

---

---

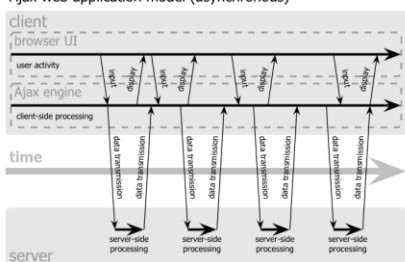
---

---

### Ajax web application model



Ajax web application model (asynchronous)




---

---

---

---

---

---

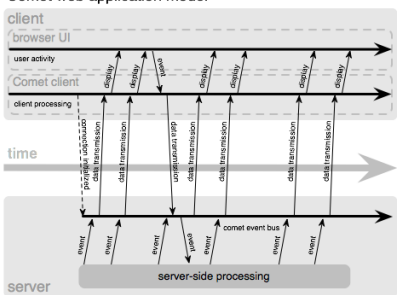
---

---

### Comet web application model



Comet web application model




---

---

---

---

---

---

---

---

---

---

---

---

### What does Comet stand for?



- It doesn't!

if you have



then why not




---

---

---

---

---

---

---

---

---

---

---

---

### Polling (or Periodic Refresh Pattern)



- Browser can poll the server for updates every few seconds
- Use **setInterval** in JavaScript
- If data changes infrequently many requests will be redundant
- Increasing polling interval increases latency
- Need to find compromise for polling interval
- Scaling to large number of clients can cause excessive network traffic




---

---

---

---

---

---

---

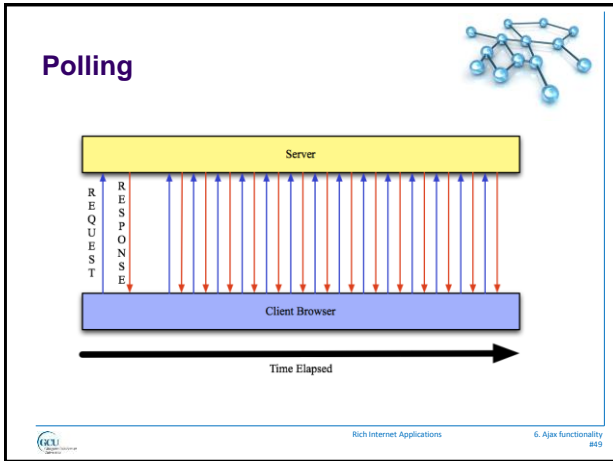
---

---

---

---

---



---

---

---

---

---

---

---

---

- ### Long Polling
- Emulates server push
  - The browser makes an async request
  - Server may wait for data before responding.
  - After processing of the response, the browser creates and sends another request
  - Browser always keeps a request outstanding with the server
  - Needs server-side support for the technique, e.g CometD
- Rich Internet Applications 6. Ajax functionality #50

---

---

---

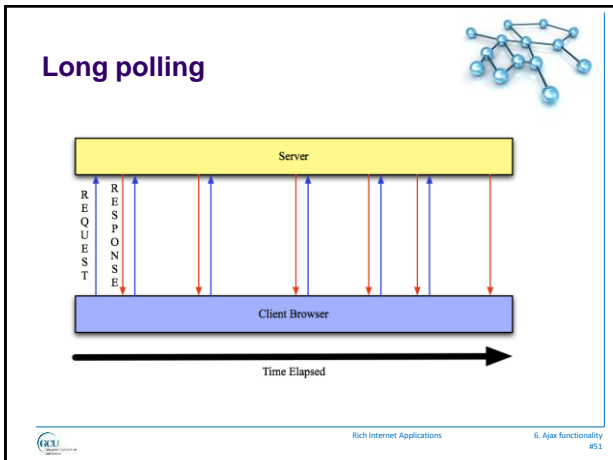
---

---

---

---

---



---

---

---

---

---

---

---

---

### Forever frame



- The forever-frame technique uses HTTP 1.1 chunked encoding to establish a single, long-lived HTTP connection in a hidden iframe
- Data is pushed incrementally from the server to the client over this connection, and rendered incrementally by web browser
- A hidden iframe element is opened in the browser after page load, establishing a long-lived connection inside the hidden iframe

---

---

---

---

---

---

---

---

### Server Sent Events



- Technology for providing push notifications in the form of DOM events
- The Server-Sent Events EventSource API is now being standardized as part of HTML5
- Specify URL of service when creating Eventsource object in JavaScript
- Register event listener functions
- Server must respond with Content-Type: text/event-stream

---

---

---

---

---

---

---

---

### WebSockets



- The WebSocket specification defines a full-duplex single socket connection over which messages can be sent between client and server
- Uses HTTP request/response as handshake to establish connection, then uses WebSocket protocol for bidirectional communication
- Requires browser and server support

---

---

---

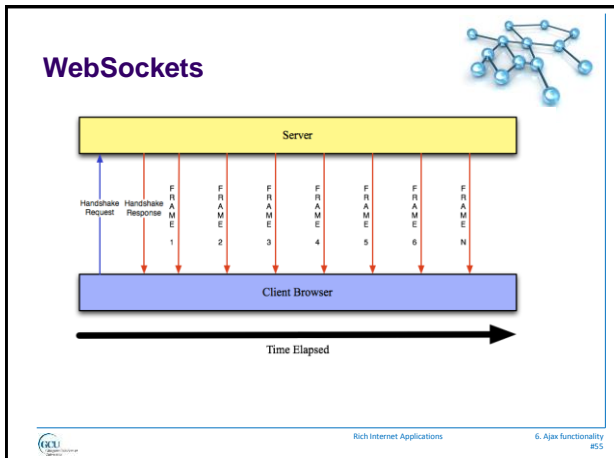
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

### SignalR

- Async signaling library for .NET to help build real-time, multi-user interactive web applications
- Client and server side components
- Publish and subscribe architecture, through creating a server side **hub**
- Clients connect to hub, and hub can publish to all clients

Rich Internet Applications 6. Ajax functionality #56

---

---

---

---

---

---

---

---

---

---

### SignalR transports

- Can use:
  - WebSockets
  - Server sent events
  - Long polling
  - Forever frame
- Depends what is supported by client and server
- Negotiated when connection is requested

Rich Internet Applications 6. Ajax functionality #57

---

---

---

---

---

---

---

---

---

---

### SignalR example - client



```

$(function () {
  var chat = $.connection.chat; connect to hub named "chat"

  chat.addMessage = function (message) {
    $('#messages').append('<li>' + message + '</li>');
  }; JavaScript method which processes data

  $("#broadcast").click(function () {
    chat.send($('#msg').val()); calls "send" method of hub (C# method)
  });

  // Start the connection
  $.connection.hub.start();
});

```

---

---

---

---

---

---

---

---

### SignalR example - server



```

public class Chat : Hub class derived from Hub class
{
  public void Send(string message)
  {
    Clients.AddMessage(message); call JavaScript method on all connected clients(!)
  }
}

```

---

---

---

---

---

---

---

---

### What's next?



- Patterns and libraries for single-page applications

---

---

---

---

---

---

---

---