



# **Rich Internet Applications:** Better Practices for Financial Services

Stephen Turbek, Avenue A | Razorfish

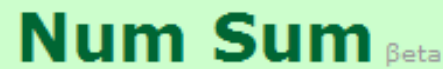


# New technologies, new opportunities

**Rich Internet Applications (RIA) are web applications that transfer some of the processing to the client end.**

- RIA's typically:
  - run in a web browser, or do not require software installation.
  - Can be built to run without a HTTP connection (but this is rare)
- Macromedia Flex
  - Better, more industry standard, & faster Flash
- AJAX Technologies
  - Asynchronous JavaScript+CSS+DOM+xmlHttpRequest

# Lots of new ideas from Web 2.0



Backpack™



Basecamp™





# Lots of new ideas



## But: Financial Services have real problems

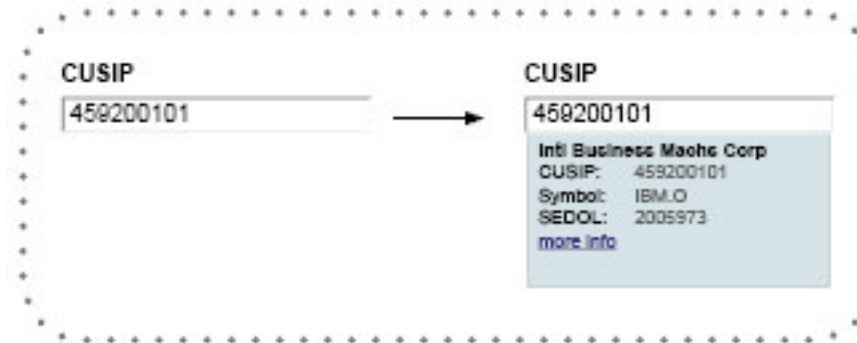
- Financial data is ridiculously complex
- FS applications are by nature highly transactional
- Time is money and training time is too much money
- If an application is not intuitive, users will not adopt it
- FS applications have people *whose job it is to use it*



# **Some Better Practices**

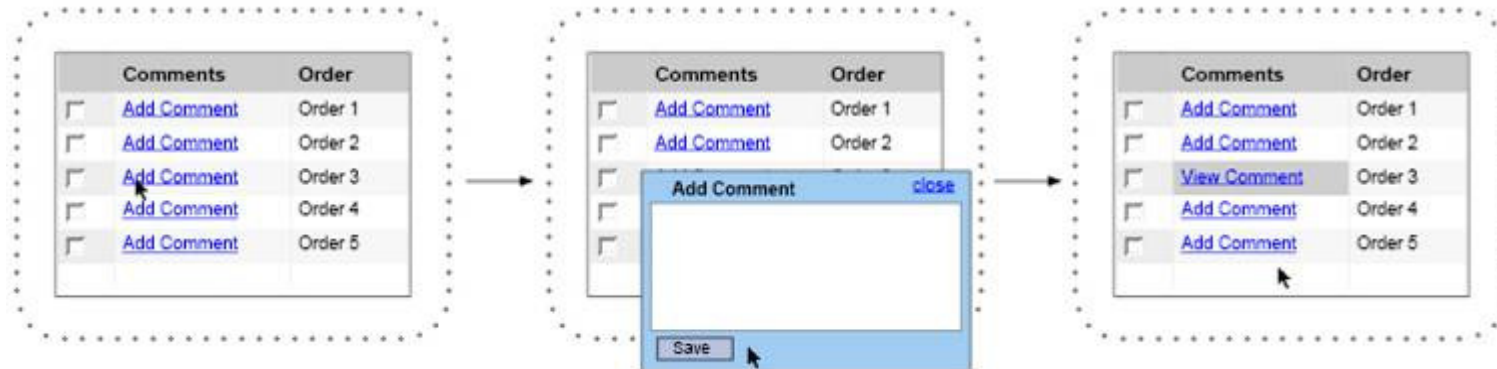


# Confirm data entry with info layers



Reduce errors by “unpacking” numeric codes

# Display features as needed



Reduce page complexity with layers



# Highlight changes

Trade Currency  
US Dollar

Security	Ticker	Buy / Sell	Price
<input type="checkbox"/> ABC corp	ABC.n	Buy	123.45
<input type="checkbox"/> BCD corp	BCD.n	Sell	123.45
<input type="checkbox"/> CDE corp	CDE.n	Buy	123.45
<input type="checkbox"/> DEF corp	DEF.n	Buy	123.45

Trade Currency  
Thai Baht

Security	Ticker	Buy / Sell	Price
<input type="checkbox"/> ABC corp	ABC.n	Buy	4,805.291
<input type="checkbox"/> BCD corp	BCD.n	Sell	4,805.291
<input type="checkbox"/> CDE corp	CDE.n	Buy	4,805.291
<input type="checkbox"/> DEF corp	DEF.n	Buy	4,805.291

On complex pages, highlight the effects of user actions

# Enable users to control *their* data

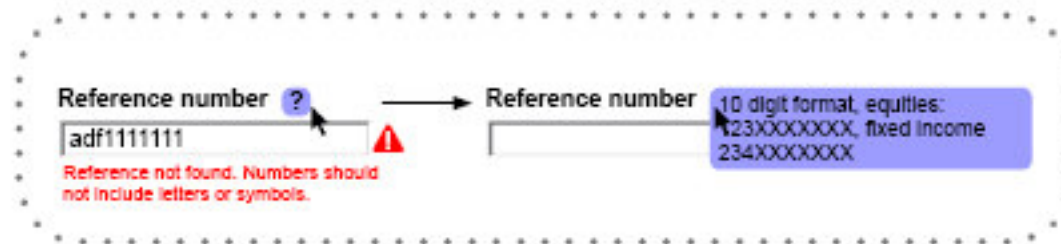


The screenshot shows a data table with a header row and four data rows. Above the table are several control options: 'Select All None', 'Print selected', 'Export selected as EXCEL', and 'Delete selected'. The table has columns for 'Security', 'Ticker', 'Buy / Sell', 'Price', and 'Date Time'. Each row in the table has a checkbox in the first column. The table is enclosed in a dashed border.

	Security ▼	Ticker	Buy / Sell	Price	Date Time
<input type="checkbox"/>	ABC corp	ABC.n	Buy	123.45	01 JAN 2006 09:09:09 AM EST
<input type="checkbox"/>	BCD corp	BCD.n	Sell	123.45	01 JAN 2006 09:09:09 AM EST
<input type="checkbox"/>	CDE corp	CDE.n	Buy	123.45	01 JAN 2006 09:09:09 AM EST
<input type="checkbox"/>	DEF corp	DEF.n	Buy	123.45	01 JAN 2006 09:09:09 AM EST

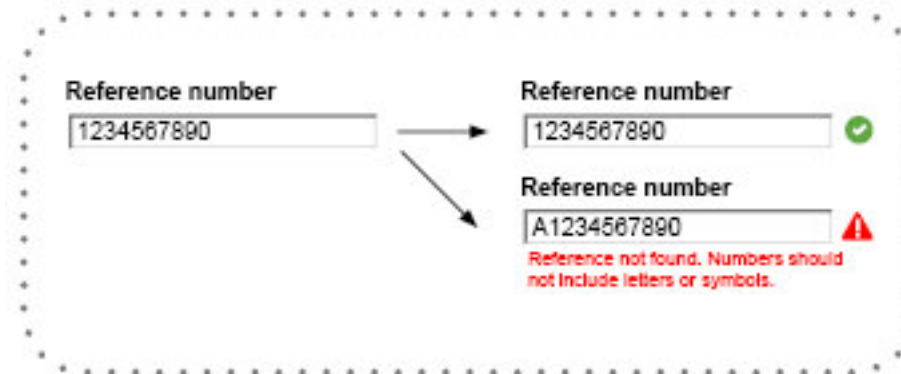
Let clients Search, Filter, Sort, and Export, or build a simple API and let them do it

# Help users use help



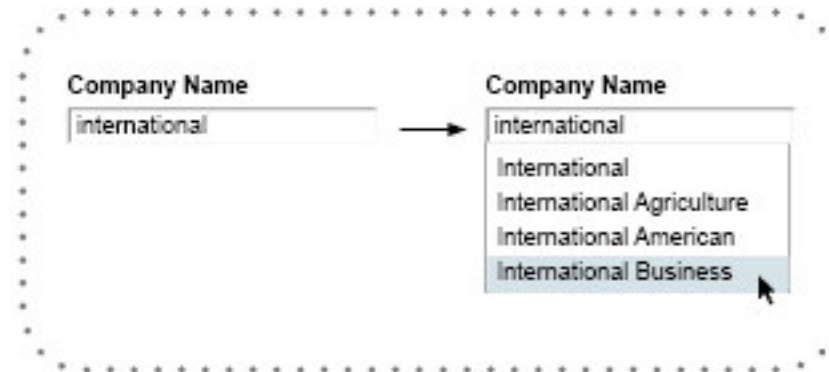
Put help where it can be found, when it is needed

# Validate data during data entry



Finding errors **AFTER** submitting is unpleasant and inefficient: help the user avoid them

# Suggest values during data entry



“Find as you type” need not be limited to search, but can speed up data entry tasks, such as company name

# Design for the 80 AND the 20

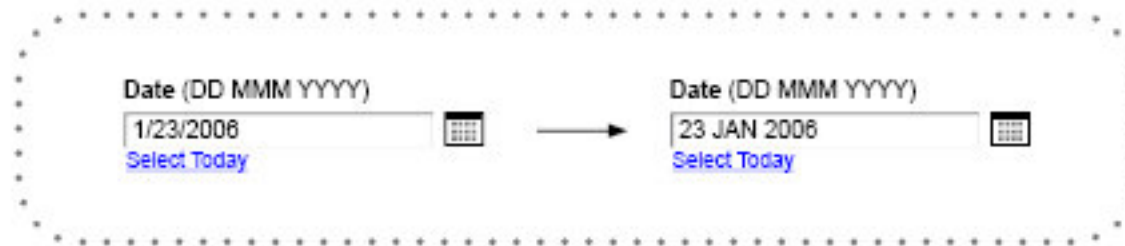
The diagram illustrates a design for an expandable form section. It shows two states of a form element:

- Collapsed State:** A header bar labeled "Optional Fields" with a "Show" link and a downward-pointing arrow icon on the right.
- Expanded State:** A larger container labeled "Optional Fields" with a "Hide" link and an upward-pointing arrow icon on the right. Inside this container, there are four input fields arranged in two rows:
  - Row 1: "Reference number" (input: 1234567890) and "CATID" (input: 1234567890)
  - Row 2: "Catalog number" (input: 1234567890) and "SEC ID" (input: 1234567890)

An arrow points from the collapsed state to the expanded state, indicating the transition.

Contain optional and rarely used form fields in an expandable section

# Correct data format errors automatically



Inconsistent formats are a common cause of errors.  
Fix them during the entry process and train the user

# New challenges from new solutions

- No code, no problems
- Progress: a browser crash can now lose lots of data.
- Web based applications need to be tested like desktop applications: Regression testing can be difficult
- Breaking the browser: Reload and Back
- Customer / User Acceptance Testing processes need to be updated





**More is possible = More will be expected**



# Thanks!

[Stephen.Turbek@avenuea-razorfish.com](mailto:Stephen.Turbek@avenuea-razorfish.com)

Presentation available at: [stephenturbek.com](http://stephenturbek.com)