JavaSpecs

A Web-Based Platform for Java Specifications

Team May1639
Arik Coats | Evan Dye | Robert Kloster

Specifications

What are specifications?

- Formal contract between the software (provider) and the user (client)
- Defines code behavior: preconditions, postconditions, invariants, etc.
- Similar to formatted pseudo code documentation

Benefits

- Avoid more errors while coding
- Faster coding and debugging.
- Reliable, testable code.

```
/*@
        requires 0 <= amount;
          ensures \result == balance
          && balance == \old(balance - amount) ;
@*/
public int withdraw_amount(int amount)
{
        ...
}
```

Problem Statement

Despite their usefulness, there is a severe lack of formal specifications for common software libraries. There are several reasons for this:

- In order to specify your software, all of the code the software relies on must be specified as well.
- Formal specifications can be difficult to write and may require uncommon knowledge.
- Writing formal specifications can be time consuming, so many companies do not consider them worthwhile or cost effective.

Purpose of project

The implementation and development of a Q&A web forum dedicated to the creation, discussion, and refinement of formal behavioral specifications for commonly used Java libraries.

Solution Plan for CprE/SE 492

Construct a Q&A forum for software developers to view, discuss, create, and approve specifications, and meets the following requirements:

1. The forum will contain the source code for classes and methods of commonly used Java libraries for easy reference.

2. The forum will offer users meaningful, related posts from Stack Overflow.

3. The forum will provide a RESTful API for easy access of data by 3rd party systems.

We Used:

Question2Answer (Q2A):	Online question-and-answer web forum structure (Version 2).
MyBB:	Online web forum structure (Version 1).
phpMyAdmin:	Database access and maintenance
MySQL:	Database language
Primary Languages:	HTML, PHP, MySQL, JavaScript, Java

Major Challenges

Transitioning from MyBB to Question2Answer

- MyBB: too restrictive, not suited to our purposes
- Question2Answer: lightweight, easy to design for and has a better format
- Transitioned existing work and developed new features within a week to justify the transition to our advisor/client

Acquiring new web server virtual machine (VM)

- Project file space lacked command line access, could not implement several major features
- Developed and tested features locally
- Gathered requirements for a web server and submitted them in a request to the Electronics and Technology Group
- Migrated project to new web server and integrated locally developed features into the website

Source Code Extraction

Researched many possible solutions to extract source code information.

Unused solution: GrepCode API

- Limited API can not extract sufficient data.
- Cannot be used in an automated system.

Unused solution: Java Reflection API

- Limited available data for extraction.
- Tedious manual parsing.
- Difficult to implement for multiple java files.

Chosen solution: use third party parser

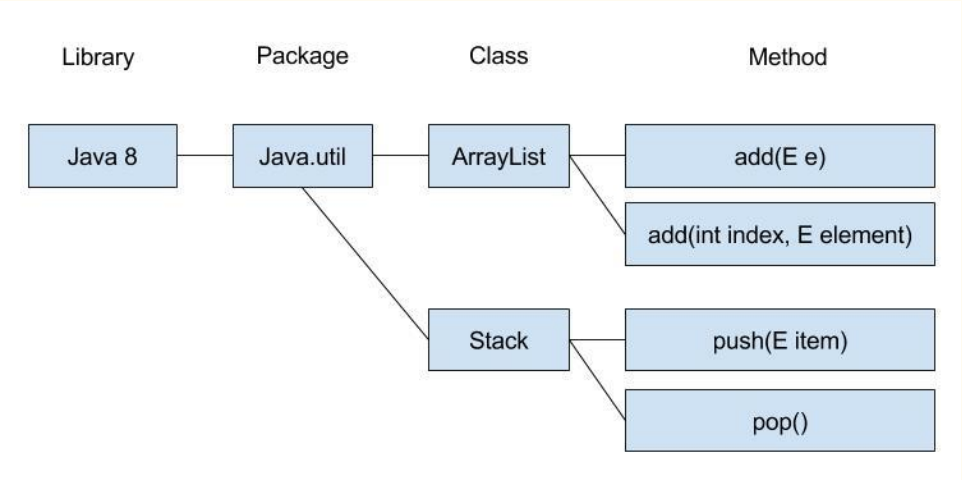
- Eclipse JDT, Abstract Syntax Tree
- JDBC, MySQL

Parser

- Searches directory for Java files
- Parse source code into component parts using the Eclipse JDT library
- Insert parsed source code data into forum database using JDBC

Navigation

- Navigate library structure for supported libraries
- View source code
- Navigation bar maintains a path with clickable links similar to how a file explorer maintains a path to the current directory.



Navigation

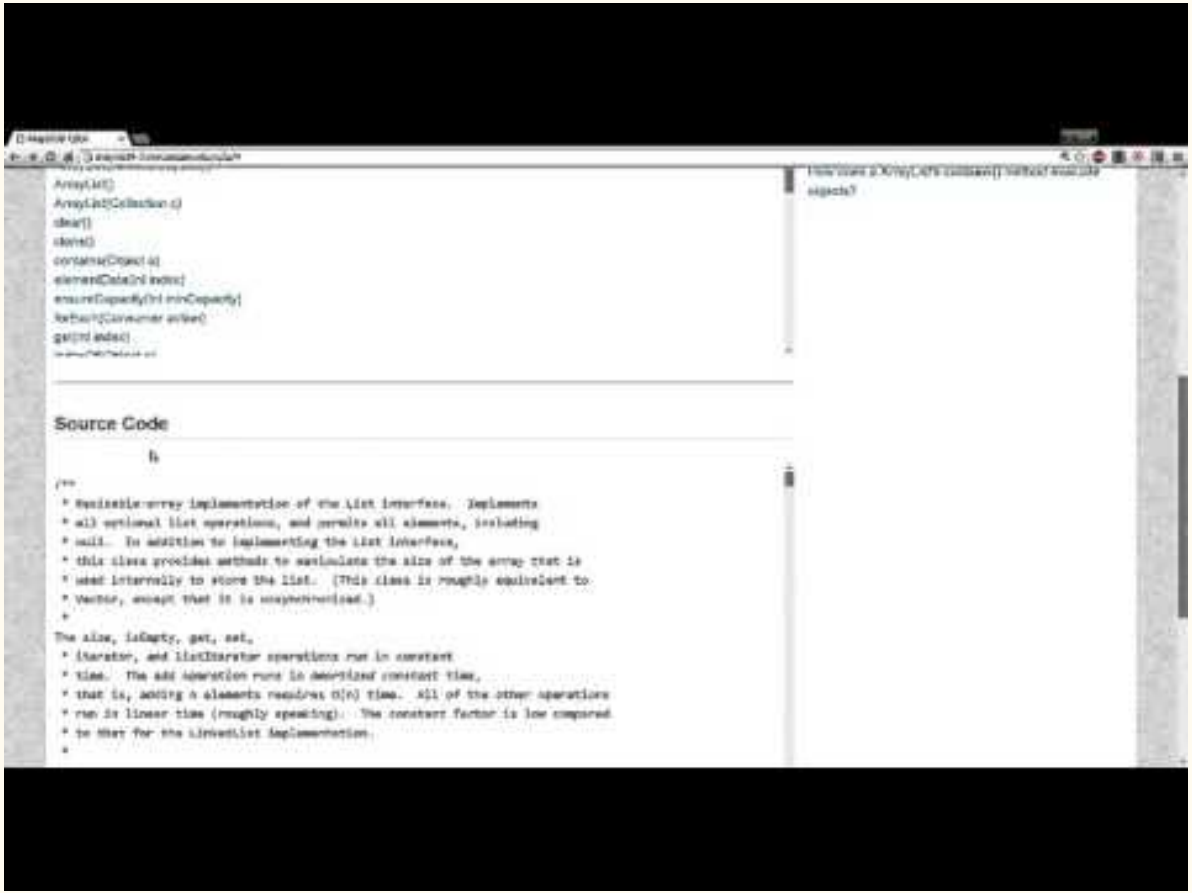
Challenge: MyBB

- No feasible way to implement navigation on MyBB
- Attempted directly inserting forums into database, nearly broke website.

Solution: Question2Answer (Q2A)

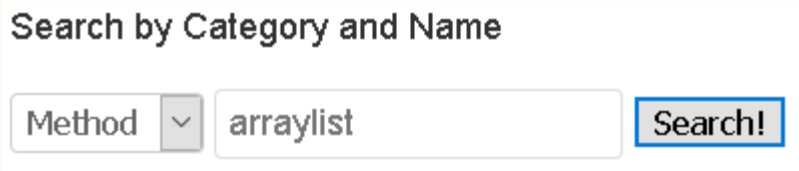
- Lightweight, non restrictive platform similar to StackExchange
- Given a week to transition website to Q2A and prove that it is feasible.
- Set up Q2A and developed navigation system using AJAX and PHP.

Navigation Demo

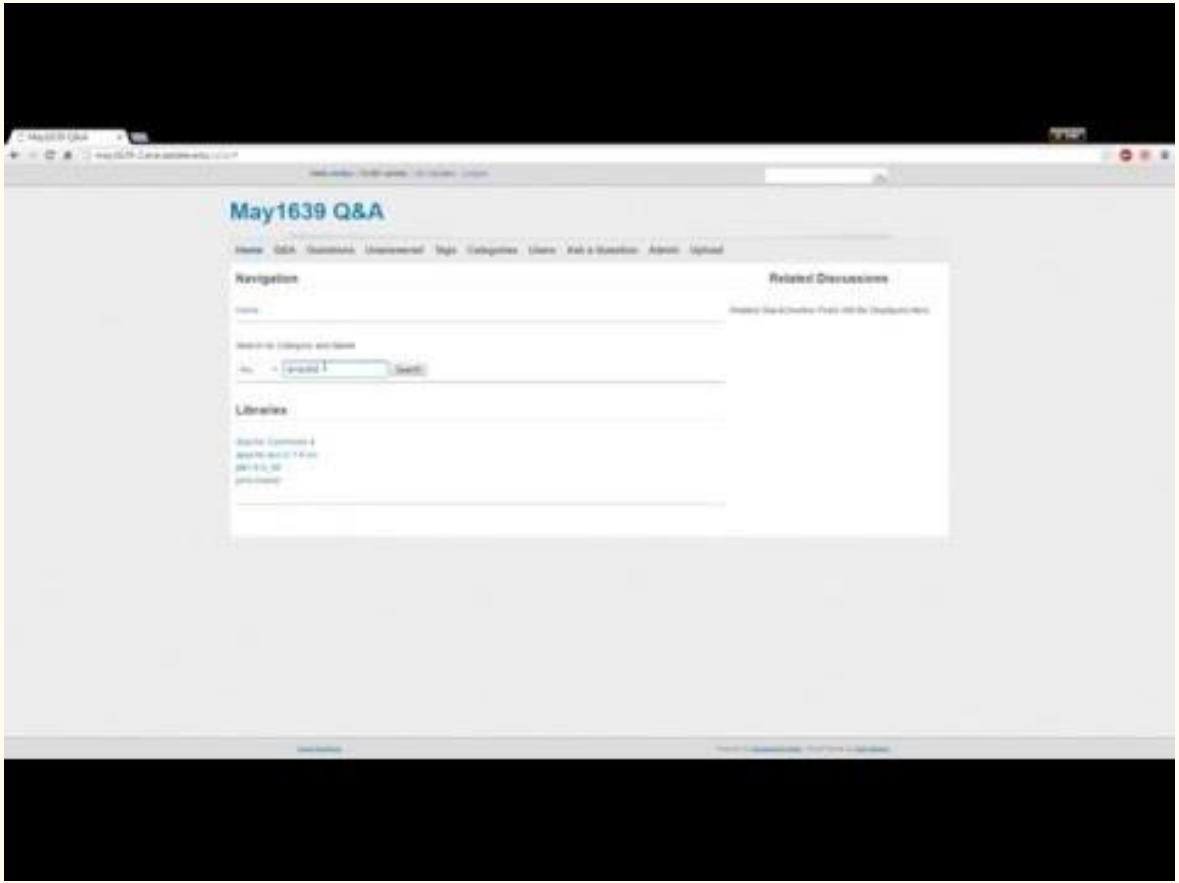


Search

- Search for name of library, package, class or method to quickly find the information you want.
- Returns list of paths to matches with working links for each item.
- On selection, navigates to location and updates main navigation bar



Search Demo



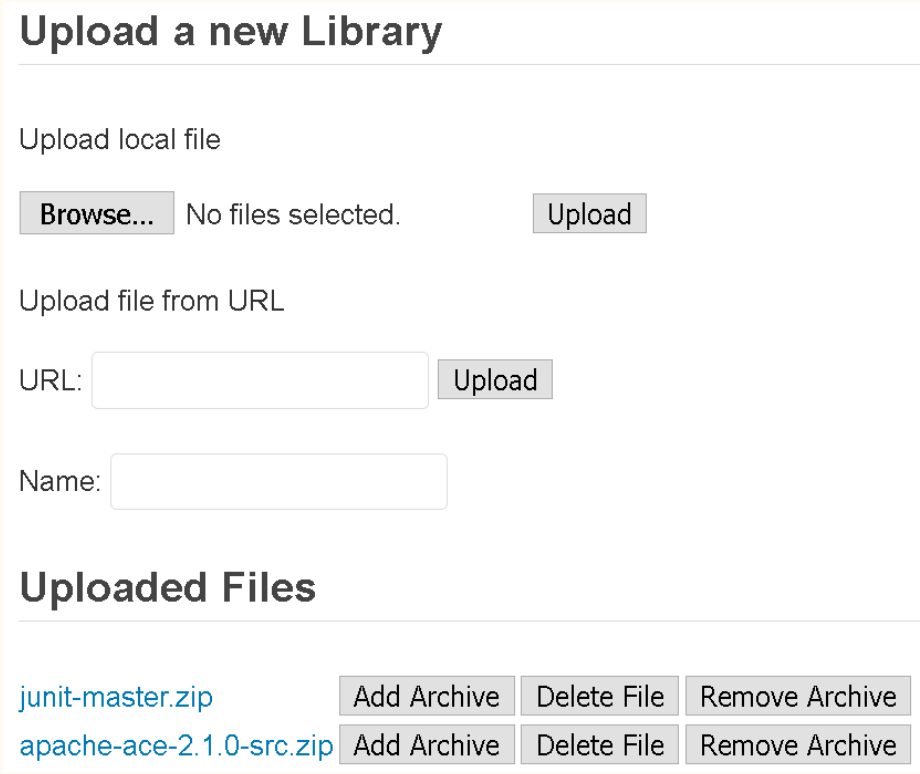
Upload

1. Allow admins to upload new libraries from:

    - A file on the user's machine
    - A url to a downloadable archive

2. Display all uploaded files and provide buttons to:

    - Download the file
    - Extract and parse source code into database
    - Remove library from database
    - Delete file



Upload

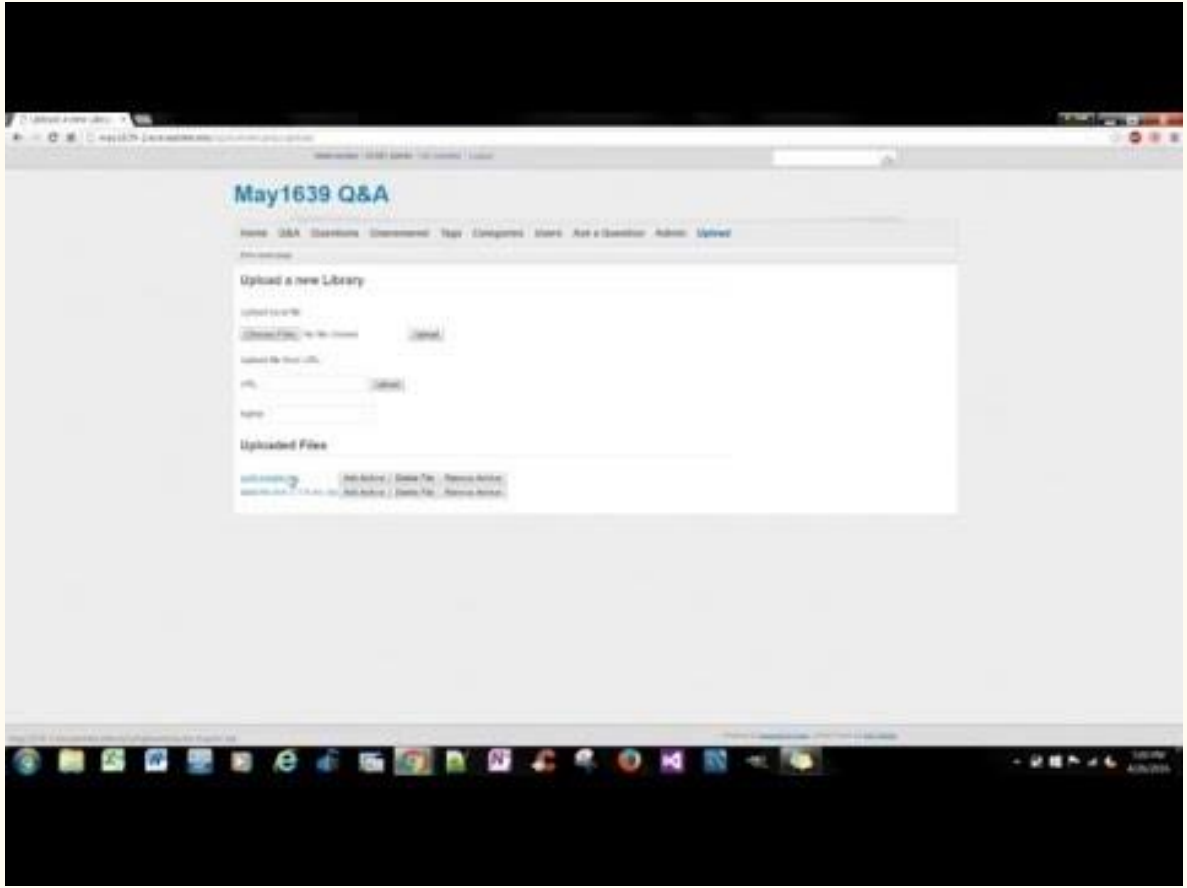
Challenges: Project File Space

- No Command line access: Java
- PHP file upload restrictions

Solutions

- XAMPP: Developed upload system locally
- Web Server: Integrated upload system into website on our new project web server

Upload Demo



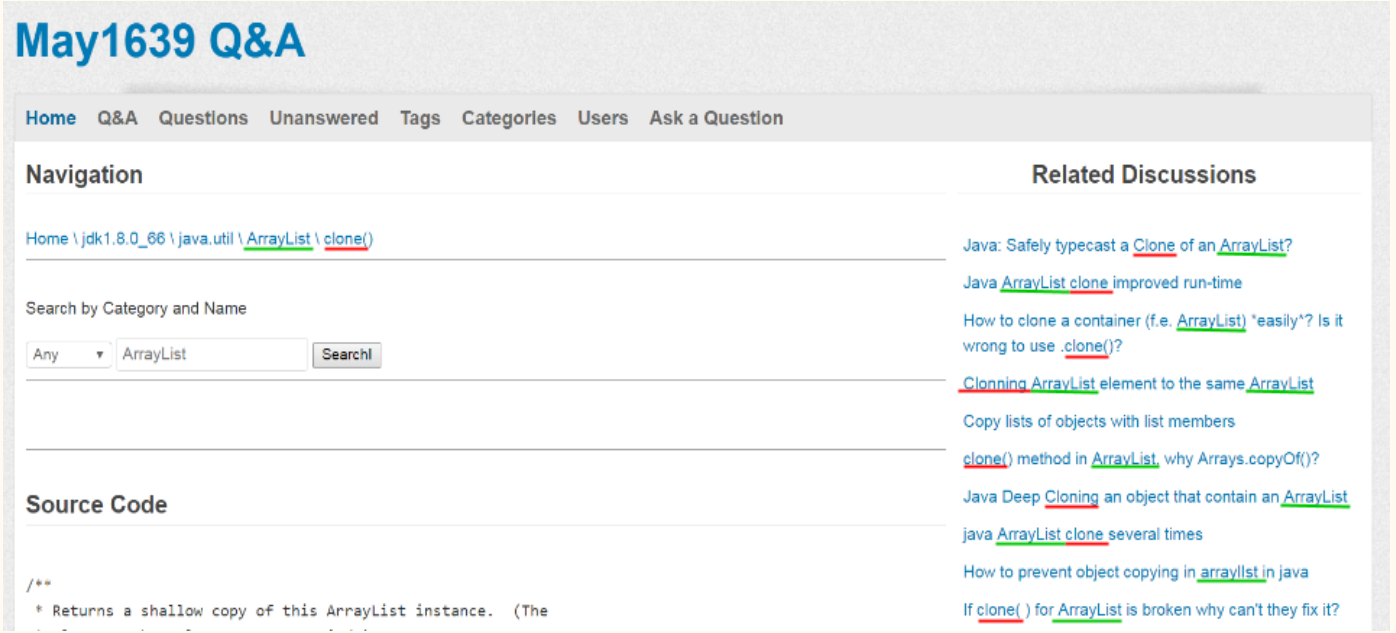
Related Posts: Goal

For each class/method:

- Generate a list of links to related Stack Overflow posts.

Purpose:

- Provide resources to research the class/method under discussion.



Related Posts: Post Information

Stack Exchange Data Dumps provide all necessary information.

- Stack Overflow Posts XML Data (~36 GB)

Parsing the Information

- Used PHP XMLReader for sequential parsing.
- Parsed necessary post information into a database, including:
    - Title
    - Tags
    - Score
    - Accepted Answers
- Created a dictionary of all title words and tags.

Related Posts: Ranking

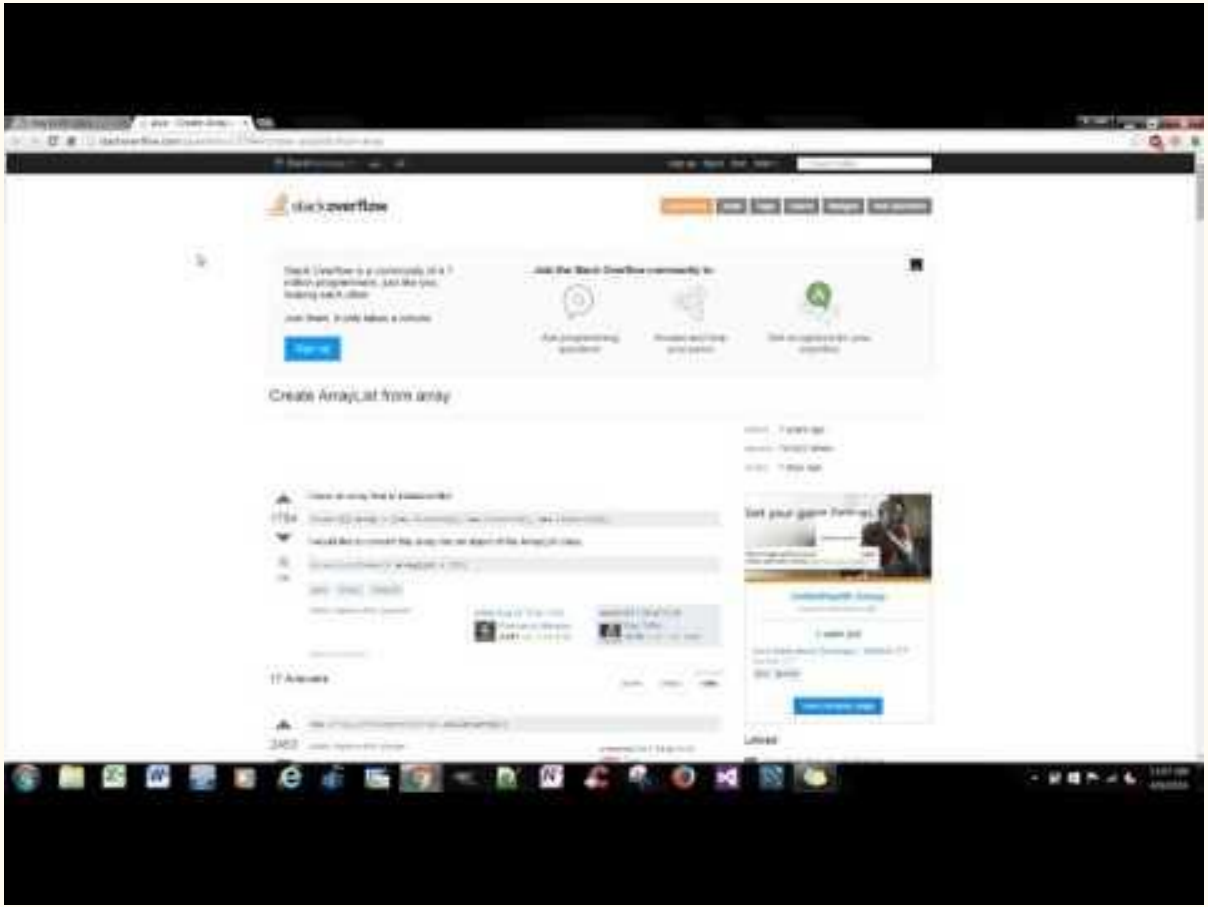
1. Consider relevant post information for a given class or method:
   - Tags - the class and/or method name is a tag for a question.
   - Title - the class and/or method name is in the title of a question.
   - View Count - indicates popular question, likely with reliable answers.
   - Score - indicates quality question or answer.

2. Rank potentially relevant posts.
   - $\text{Rank} = C_1 * \text{Normalize}(R) + C_2 * \text{Normalize}(T) + C_3 * \text{Normalize}(V) + C_4 * \text{Normalize}(S)$
     - R = Number of relevant tags (i.e. class, method) for a question.
     - T = Number of relevant words (i.e. class, method) in a post question.
     - V = View count of a question.
     - S = Score of a question.
     - Where C_i is a weighted constant.

Related Posts: Challenges

1. Stack Exchange API Method
   - Problem
     - 300 queries per day limit per IP (10,000 with Key).
     - 30 requests per second limit per IP.
     - Expensive to parse all interesting content in real time.
   - Solution - Stack Exchange Data Dump
     - Store information in a local database.
     - Provides easy access to any required information.

2. Stack Exchange Data Dump Method
   - Problem
     - Expensive to search through millions of posts.
   - Proposed Solution - Pre-Rank Posts
     - For each class/method, determine related posts in advance.
     - Store ranks in database.

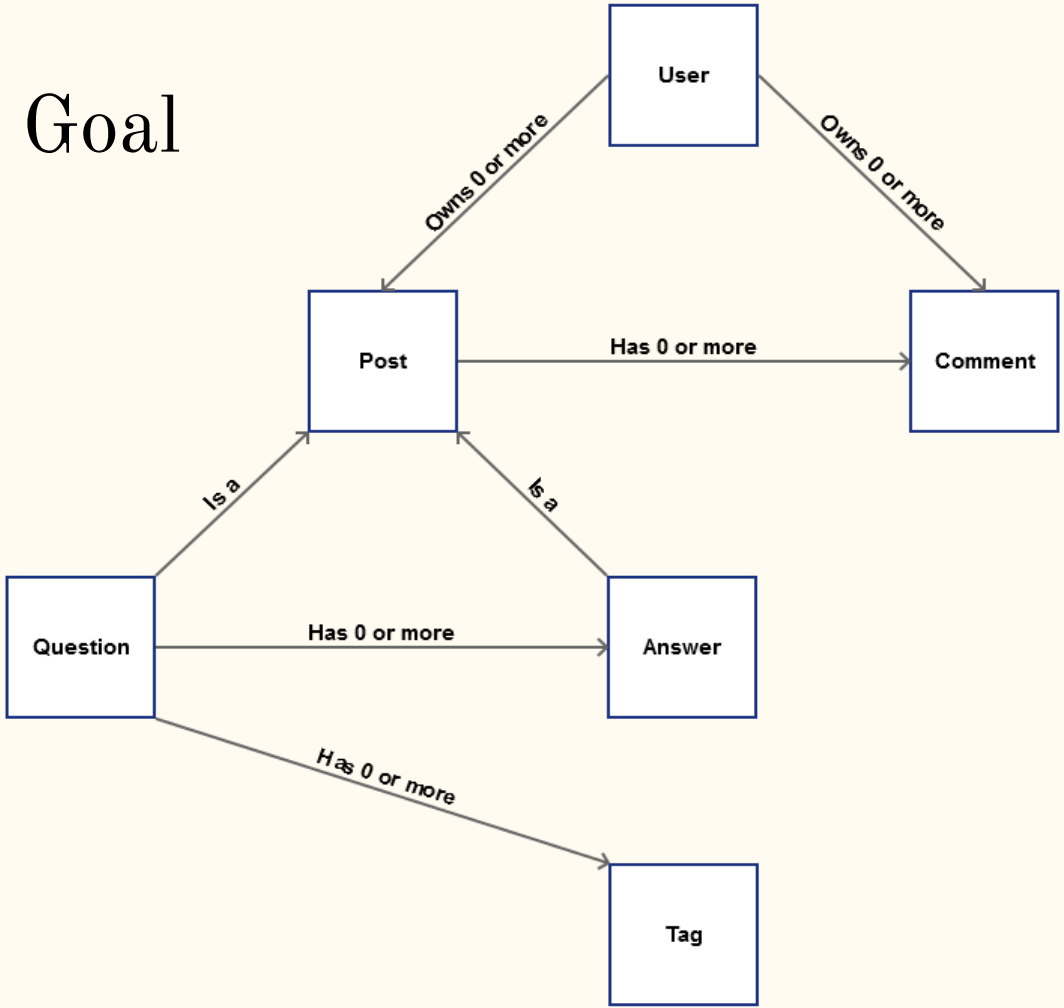

stackoverflow
Create ArrayList from array

API Implementation: Goal

Provide 3rd-party software with information from the database about:

Questions	Answers
Posts	Comments
Users	Tags

through a RESTful API.



API Implementation: Challenges and Solutions

Challenge: Designed after StackExchange

- Majority of StackExchange API calls didn't apply for the project's intended use
- Limited/inflexible searching and sorting abilities

Solutions

- Combined and condensed API calls by returned type of data
  - Allows for easy integration for new API calls in the future
- Designed input structure after GitHub API repository search call
  - Significantly increased flexibility of searching and sorting

API Implementation: Search Functionality

Integer and Datetime Variables

- Comparative conditions: =, !=, <, >, <=, >=
- Range conditions: low_value .. high_value

Boolean Variables

- Comparative conditions: true, false

String Variables

- Contains conditions: IN, NOT IN
  - Performed across all string variables by default
- Can specify which string variables are included in the search

API Implementation: Additional Functionality

Sort

- Can sort by any number of given variables
- Can select either ascending or descending for each sort variable

Pagination

- Can set pagesize from 1 - 100 (default is 30)
- Can select specific page (default is page 1)

Question Variables

Integer	**String**	**Datetime**	**Boolean**
question_id	title	creation_date	is_answered
owner	body	last_activity_date	
view_count	tags	last_edit_date	
score			
up_vote_count			
down_vote_count			
answer_count			
accepted_answer_id			

http://may1639-2.ece.iastate.edu/q2a/api.php/**questions**


```
{
  "Items":[
    {
      "question_id":1,
      "title":"Is this forum better?",
      ...
    },
    {
      "question_id":2,
      "title":"Test Question",
      ...
    },
    ...
  ]
}
```

http://may1639-2.ece.iastate.edu/q2a/api.php/questions?**conditions=question_id:>2**


```
"Items":[
    {
      "question_id":3,
      "title":"Testing Tags",
      ...
    },
    {
      "question_id":4,
      "title":"How to make Widgets?",
      ...
    },
    ...
  ]
}
```

http://may1639-2.ece.iastate.edu/q2a/api.php/questions?conditions=**Widget** question_id:>2


```
{
  "Items":[
    {
      "question_id":4,
      "title":"How to make Widgets?",
      ...
    }
  ]
}
```

Summary

Specifications: beneficial for testing and quality assurance

Problems: time and knowledge constraints, specification dependencies

JavaSpecs: Q&A web platform for specifications of common Java libraries

- Source Code on server
- Can upload new libraries
- Display helpful posts from StackOverflow
- API offers connection between JavaSpecs and data-mining softwares.

Questions?

Resources

1. Rajan, Hridesh, Tien N. Nguyen, Gary T. Leavens, and Robert Dyer. "Inferring Behavioral Specifications from Largescale Repositories by Leveraging Collective Intelligence." “37th International Conference on Software Engineering: NIER Track” May 2015, ICSE’15, Florence, Italy.

Misc.

How To Use API?

Manual included in ‘Final Report’ on Project Website Documents page

http://may1639.github.io/documents.html