

SingPath API Specification

This is an evolving specification for the SingPath API. This specification will focus on the parts of the design that are currently being worked on and expand to cover all features.

Vision and Direction:

We would like university students to be able to ramp up on this API quickly and begin to enhance the codebase by adding new features, adding test coverage, and removing duplicate code. To support this, we would like the specification to improve with this target in mind and for the code to be as easy as possible for students to dive in to a particular area and start adding value and making good pull requests. Similarly, we would like to be able to reach out to professional developers on oDesk to assist students with issues that may be more challenging for them and to ensure that the code quality of the codebase is steadily improving and adopting best practices. And finally, we would like to eventually open source this code base so that a larger group of students and professionals might be able to contribute to it. To accomplish this, we would like to ensure that appropriate security has been placed on the API so that it will be more difficult for mischievous students to identify ways to negatively impact system performance and the results for themselves and their peers. Additional API protections and limits on the creation of new objects will need to be put in place to limit the damage that automated bots designed by students might be able to cause.

Test Environment: Nitrous.io Python Box

In order to help students and oDesk developers ramp up on the SingPath and Google App Engine environments quickly, a step-by-step setup guide for enabling a Nitrous.io python box will be maintained in the readme file. This online IDE will not only provide a faster way to edit and test code, but will also be the common environment to run all tests prior to accepting Pull Requests. The goal is for this project to eventually be built and tested by Travis.io after it is a public project, but in the interim, the project will be built and tested in a Travis.io box based on the steps outlined in the readme file.

Setup in Readme

1. Sign-up to Nitrous.io
2. Create a new Python box
3. git clone <https://github.com/SMU-SIS/Singpath>
4. cd Singpath
5. source nitrous/bin/activate
6. nosetests -w unittests

Or to check for test coverage you can run `--with-coverage` (427 seconds to test with coverage)

7. nosetests --with-coverage --cover-package=singpath,livejsonapi -w unittests

Coverage was 43% for these two packages on January 20th, 2014. We would like to see

coverage move to be greater than 50% as unneeded files are removed and additional tests are written.

In order to pull in the remote GUI projects you will need to init the submodules and then pull the latest code.

8. `git submodule update --init`

Updating Submodules: Update remote GUI project by running

9. `git submodule foreach git pull origin master`

[SingPath GUI Specification](#) - The genshyft Angularjs GUI that uses this API.

Github: <https://github.com/ChrisBoesch/genshyft>

Development Server: This is where passing Pull Requests are deployed to.

<http://development.singpath.appspot.com>

Running local server:

`dev_appserver.py --host=0.0.0.0 --high_replication .`

Accessing `/bootstrap` will load the local database with some default data for testing.

Deploying to development

`appcfg.py --version=development update .`

/bootstrap - Local testing setup for integration testing

There is a `/bootstrap` url included to set the system up with some test data to do local, manual integration testing, and debugging of issues. This `/bootstrap` could be improved to better support the creation and interactive debugging of new features. There is a possibility that the Nitrous.io box will run out of memory. We've seen this happen from time to time on 384KB memory boxes, but things usually work without any issues.

References:

Virtualenv tutorial: <http://iamzed.com/2009/05/07/a-primer-on-virtualenv/>

Todos:

1. After launching the local `dev_appserver` and accessing `/bootstrap`, the user should be able to browse through a GUI and:
 - a. play a practice game
 - b. play a quest

- c. Edit a tournament
 - d. Start a tournament
 - e. Play in the tournament that they started
2. Add API test support for all of the API's used in the game-app-test.js file of the genshyft GUI project.
3. Increase coverage to 50% by adding unit tests for uncovered code.
4. Find and remove large blocks of duplicate code using a tool such as clonedigger

Specification

SingPath is designed primarily to support software education in blended learning environments that have a classroom session and the ability for students to practice prior to coming to class.

Features and API's defined in urls.py

- [Practice](#)
- Quests
- [Tournaments](#)
- Events
- Mastery-based Coaching
- [Purpose-driven Development](#)
- Challenges
- Rankings
- School Registration
- Drag-n-Drop

Practice

Here is what you will see after you bootstrap, have logged in, and are playing a practice game. All the bootstrap problems require you to make $b=2$.



Sandra

Log Out

SingPath

Quests

Practice

Challenges

Rankings

Feedback

More... ▾

Profile

Level Progress: 5/10



Game progress 3 of 5

Current Practice Question

Name: test_problem_9

Description: Make b==2.

Path Name: test_path0

Level: 1

Difficulty: Easy

Skip

Solution

Run

```
1 b=1
2
```

Sample Answer

Outcome

Called	Expected	Received	Correct
b	2	1	false

Tournaments

Tournament: operations for tournament

GET	/list_tournamentQns/{tourQns}	Retrieve tournament qns	
Implementation Notes Retrieves tournament questions			
Response Class list_tournamentQns { language (String) pathLevel(String), questionSet (array[String]) } questionSet{ questionID (integer), question (String), questionDescription (String), questionExamples (String), skeleton (String) }			
Parameters			
Parameter	Description	Parameter type	Data Type
language	Coding language	Query	String
pathLevel	Coding language topic	Query	String
questionSet	List of questions with respect to the language	Query	String

GET	<code>/get_heat_ranking</code>	Retrieve heat ranking																
<p>Implementation Notes Retrieves the heat ranking</p> <p>Response Class ranking {</p> <pre> status (String), mentee (String), playerid (integer), Unique identifier of the current player's heat ranking solved_problems (integer), flagUrl (String), finished (String) questionSet (array[String]), gravatar (String), mentor (String), mentorID (integer), unique identifier of the player who is a mentor professional (boolean) = ['true' or 'false'] : whether player is professional, total_problems (integer) nickname (String) menteeID (integer), unique identifier of the player who is a mentee </pre> <p>}</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> <th>Parameter type</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>menteeID</td> <td>Unique id of the player who is being mentored</td> <td>Query</td> <td>Integer</td> </tr> <tr> <td>mentorID</td> <td>Unique identifier of the player who is a mentor</td> <td>Query</td> <td>Integer</td> </tr> <tr> <td>playerID</td> <td>Unique identifier of the current player's heat ranking</td> <td>Query</td> <td>Integer</td> </tr> </tbody> </table>			Parameter	Description	Parameter type	Data Type	menteeID	Unique id of the player who is being mentored	Query	Integer	mentorID	Unique identifier of the player who is a mentor	Query	Integer	playerID	Unique identifier of the current player's heat ranking	Query	Integer
Parameter	Description	Parameter type	Data Type															
menteeID	Unique id of the player who is being mentored	Query	Integer															
mentorID	Unique identifier of the player who is a mentor	Query	Integer															
playerID	Unique identifier of the current player's heat ranking	Query	Integer															

GET	/added_tournaments	Player creates and adds a tournament								
<p>Implementation Notes A group tournament is added after the data parsed</p> <p>Response Class added_grpTournaments {</p> <pre> shortTitle (String), description (String), password (String), roundCount (integer) rounds (String) utcOffset (String) tournamentID (integer) passwordConfrim (String), addDetails (String), status (String) isGroup (boolean) mentorAssignment (String) numberOfGrp (integer) numPlayerPerGrp (integer) </pre> <p>}</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> <th>Parameter type</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>tournamentID</td> <td>Unique identifier for the tournament</td> <td>Path</td> <td>Integer</td> </tr> </tbody> </table>			Parameter	Description	Parameter type	Data Type	tournamentID	Unique identifier for the tournament	Path	Integer
Parameter	Description	Parameter type	Data Type							
tournamentID	Unique identifier for the tournament	Path	Integer							

POST	/add_grptournament	Update tournaments that player has added								
<p>Parameters</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> <th>Parameter type</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>grpTournament</td> <td>Parses the data</td> <td>Path</td> <td>-</td> </tr> </tbody> </table>			Parameter	Description	Parameter type	Data Type	grpTournament	Parses the data	Path	-
Parameter	Description	Parameter type	Data Type							
grpTournament	Parses the data	Path	-							

POST	/updateTournament	Any change in details made to create tournament is added	
Parameters			
Parameter	Description	Parameter type	Data Type
shortTitle	Title of tournament	Form	String
description	Description of tournament	Form	String
Password	Password for the tournament	Form	String
tournamentID	Unique id for the tournament	Query	Integer
addDetails	Details about the tournament	Form	String
status	Whether tournament is open or not	Form	boolean

Purpose-driven Development

purposeDrivenVideos: Operations about purpose driven videos

GET	/purposevideos/{Videos}	Find Video by ID								
<p>Implementation Notes Returns a video based on ID</p> <p>Response Class purposeDrivenVideos {</p> <p>id (integer) : unique identifier for the purpose driven video, no (integer), title (String), img (String), thumbnail (String), vlink (String), description (String), question (String, <i>optional</i>), feedback(integer, <i>optional</i>), unlocked (boolean) = ['true' or 'false']: whether user can view that video</p> <p>}</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> <th>Parameter type</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>ID of video that needs to be fetched</td> <td>Path</td> <td>integer</td> </tr> </tbody> </table>			Parameter	Description	Parameter type	Data Type	id	ID of video that needs to be fetched	Path	integer
Parameter	Description	Parameter type	Data Type							
id	ID of video that needs to be fetched	Path	integer							

GET	/purposevideos/admin/{Videos}	Find Video by ID								
<p>Implementation Notes Returns a video based on ID</p> <p>Response Class purposeDrivenVideosADMIN {</p> <p>no (integer), unique identifier for the purpose videos on Admin page title (String), img (String), thumbnail (String), vlink (String), description (String),</p> <p>}</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> <th>Parameter type</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>no</td> <td>ID of video that needs to be fetched</td> <td>Path</td> <td>integer</td> </tr> </tbody> </table>			Parameter	Description	Parameter type	Data Type	no	ID of video that needs to be fetched	Path	integer
Parameter	Description	Parameter type	Data Type							
no	ID of video that needs to be fetched	Path	integer							

POST	/record_purpose_video_unlock	Update user's score on feedback form and watched videos	
Parameters			
Parameter	Description	Parameter type	Data Type
Id	ID of video that needs to be updated that it is unlocked	Path	integer
feedback	Record feedback score of watched video	Form	integer
result	Update videos that are unlocked and saves new feedback scores	Form	String

POST	/admin_update	Admin update purpose driven videos	
Parameters			
Parameter	Description	Parameter type	Data Type
Id	ID of video that needs to be updated that it is unlocked	Path	integer
result	Update videos that are unlocked and saves new feedback scores	Form	String