

Data Models

Contract Objects

contract_id:	12321412,	IntegerProperty
challenge_name:	808 Cali Bros Reddit,	StringProperty
short_name:	808CaliBrosReddit,	StringProperty
objective_type:	geolocation,	StringProperty with 'choices' option
time_period:	1,	IntegerProperty
time_unit:	weeks,	StringProperty with 'choices' option
start_date:	04/15/13,	DateProperty
stakes_id:	5415645	IntegerProperty with 'repeat' option

Geolocation Objective Objects

geo_objective_id:	58493821,	IntegerProperty
contract_id:	12321412,	IntegerProperty
checkin_loc:	4°30'2", 32°2'1",	GeoPtProperty
checkin_radius:	200, (meters)	IntegerProperty
loc_name:	"Cravings"	StringProperty

General Objective Objects

gen_objective_id:	2141	IntegerProperty
contract_id:	12321412	IntegerProperty
objective_name:	"Ran a mile"	StringProperty

Stakes Objects

stakes_id:	5415645,	IntegerProperty
position:	first,	StringProperty with choices
stakes_desc:	"doesn't have to clean",	StringProperty

General Progress Objects

objective_id:	58493821,	IntegerProperty
combatant_id:	44314,	IntegerProperty
checkin_count:	3,	IntegerProperty
last_checkin:	04/15/13	DateTimeProperty

Combatant Objects

combatant_id:	44314,	IntegerProperty
name:	"Team B"	StringProperty
challenges_won:	3	IntegerProperty

User Objects

user_id:	441564315,	IntegerProperty
google_username:	liamhession	StringProperty
phone_number:	479-231-2312	StringProperty

challenges_won: 4, IntegerProperty

Combatant-User Objects

user_id: 441564315, IntegerProperty

combatant_id: 44314, IntegerProperty

Contract-Combatant Objects

contract_id: 12321412, IntegerProperty

combatant_id: 44314, IntegerProperty

position: 2, IntegerProperty (place in the challenge)

Desired Users Objects

contract_id: 12321412, IntegerProperty

users_to_join: [dude@gmail.com, bro@gmail.com], StringProperty with repeat option

Client Storage

After a user logs in, localStorage will store all their contract info and associated challenges. This way, if they have no connection they can still go to any of their active challenge pages and log a check-in. This check-in info will be uploaded to the servers after they regain access to the internet.

Protocols/APIs

/

GET -> (no params)

template context: [exampleText, topChallenge1, topChallenge2, topChallenge3]

Template page: "frontpage.html"

Creates a home page that has a new challenge field with exampleText in it, then a list of the current most popular challenge types.

/new

POST -> {description = "Who can go to Siebel Center the most in a week"}

Sets up a new item, gets its shorthand name, redirects to a partially-filled details page.

/[challenge_name]/details

GET -> (no params)

template context: [description, objective, length, time_units, start_date = {month, day, year},

stakes = {place1: stakes, place2: stakes...}, checkin_action]

Template page : "details.html"

Lists the details of this challenge. Has edit mode, which is entered by default on pageload if any field is empty. Saves activate AJAX calls to "[challenge_name]/edit". Certain edits would require authentication by others.

/[challenge_name]/edit

POST -> {objective = "location", unit = "weeks", length = "2", year = "2013", month = "6", day = "10", stakes1 = "Free lunch", stakes2 = "", stakes3 = "", stakes4 = ""}

plus: {checkin-loc = "23.44, 13.03", radius = "300", checkin-loc-name = "Cravings"}

or: {objective_name = "Won a game"}

Multiple options for what extra data is being posted in order to be added to the database. Updates stored data asynchronously, no page rendered, nothing returned.

/[challenge_name]/invite

GET -> (no params)

template context: [joined = [{combatant: "combatant1", users: ["username1", "username2"]}, {combatant: "combatant2", users: ["username3"]}]]

Template page : "invite.html"

This page has an input field for email addresses of people you want to add to the challenge.

Should warn them that only gmails associated with google accounts will work at this time. It also shows a list of the combatants with their associated usernames/emails.

/[challenge_name]/send-invite

POST-> {email="liamhession@gmail.com"}

This handler doesn't render a page, it just sends out the email asking that the person at "email" join the challenge. That email contains a link to the join page

/[challenge_name]/join

POST -> relevant_logins = reddit_user

Template page: "gen-join.html", "reddit-join.html", "geo-join.html"

gen- context: [challenge-name: "Who can win the most games of beer pong in a week", inviter: "Liam". prizes=[prize1,prize2,punishment1,punishment2]

This page reloads when they press the "Join" button, adding them to the challenge. If the objective necessitates a login that is passed in here

/[challenge_name]/checkin

GET -> (no params)

page context and template that it's rendered in depends on the challenge objective:

geo - context: [id, location, radius, placename, last_checkin]

rendered on "geo-checkin.html"

general - context: [id, objective, last_checkin]

rendered on "gen-checkin.html"

To be used primarily on the mobile site, this is the page for marking that you did whatever the challenge is about, or checking in by geolocation. The user's location is compared to the destination in this case, and they can check in if in the radius.

/[challenge_name]/do-checkin

POST -> objective_id = 214

No page rendered here, it just completes the checkin action. Also then updates the positions of all involved in challenge in case they changed. Should be called asynchronously.

/[challenge_name]/status

GET -> (no params)

template context: [name, start, spans_complete, combatants=[{name, count, com_id, position}]]

rendered on: "details.html"

Shows the current status of every combatant in the challenge, ordered by number of checkins.

Also displays bar of how much time has elapsed in the challenge.

/user

GET -> (no params)

template context: [challenges = {challenge_name, link, com_name, position}, username]

List of all the challenges that user is involved with, as well as their combatant name and position in that challenge