CSE 154: Web Programming   Spring 2017

Homework 5 API Documentation

Overview

We have provided two web services for you two use on Homework 5: a Pokedex API and a Game Management API. The Pokedex API provides data about each of the 151 Pokemon, including moves, type, and weakness. Each type of query produces output in plain text or JSON format (*You can test queries by typing in their URL in your browser's address bar and seeing the result*). If you submit an invalid query, such as one missing a necessary parameter, the request will return an HTTP error code of 400 (Invalid request) rather than the default 200.

The rest of this document provides the necessary information about the endpoints and query types for the requests you will make for HW 5.

Pokedex API

Endpoint: https://webster.cs.washington.edu/pokedex/pokedex.php

The first web service, `pokedex.php`, provides data about each of the 151 Pokemon and accepts two different types of "GET" queries, specified using a query string with a parameter.

Query 1: Get Pokemon Names

Request Format: pokedex.php?pokedex=all
Request Type: GET
Returned Data Format: plain text
Description: This first request takes the parameter `all` and returns a plain text response with all 151 Pokemon names and sprite image names, each on its own line. The Pokemon name is followed by its sprite image name separated by a single ":". These sprite image names correspond to the Pokemon's sprite image in the https://webster.cs.washington.edu/pokedex/sprites/ folder.
Request: https://webster.cs.washington.edu/pokedex/pokedex.php?pokedex=all
Output: (abbreviated)

```
1  Abra:abra.png
2  Aerodactyl:aerodactyl.png
3  Alakazam:alakazam.png
4  ...
5  Zubat:zubat.png
```

Query 2: Get Pokemon Data

Request Format: pokedex.php?pokemon={name}

Request Type: GET

Returned Data Format: JSON

Description: The second request type takes as a parameter any Pokemon name and returns a detailed JSON object containing data about this Pokemon. The returned data will be used to populate a card for that Pokemon.

Example Request: https://webster.cs.washington.edu/pokedex/pokedex.php?pokemon=pikachu

Example Output:

```
1  {
2    "name": "Pikachu",
3    "hp": 160,
4    "info": {
5      "id": "25",
6      "type": "electric",
7      "weakness": "ground",
8      "description": "Melissa's favorite Pokemon! When several Pikachu gather, their electricity could
          build and cause lightning storms."
9    },
10   "images": {
11     "photo": "images/pikachu.jpg",
12     "typeIcon": "icons/electric.jpg",
13     "weaknessIcon": "icons/ground.jpg"
14   },
15   "moves": [
16     {
17       "name": "Growl",
18       "type": "normal"
19     },
20     {
21       "name": "Quick Attack",
22       "dp": 40,
23       "type": "normal"
24     },
25     {
26       "name": "Thunderbolt",
27       "dp": 90,
28       "type": "electric"
29     }
30   ]
31 }
```

The values of the returned JSON object include the name of the Pokemon (e.g., Pikachu), the type of the Pokmeon (e.g., "electric"), its weakness type (e.g., "ground"), the health points, or hp (e.g., 80), the set of images (photo for the main Pokemon image and typeIcon and weaknessIcon for the type and weakness icon image paths, respectively), and the set of moves (each Pokemon has between 1 and 4 moves; Pikachu has 4). Each move has a type, and moves that do damage have a "dp", or damage point attribute. Moves that do not have a "dp" attribute (e.g., Growl) affect stats of the player or opponent's Pokemon, which is handled elsewhere in the program (during the game mode).

Game Management API

Endpoint: https://webster.cs.washington.edu/pokedex/game.php
The second web service, game.php, accepts two POST query types to initiate and update the state of a card game.

Query 3: Start Game

Request Format: game.php endpoint with POST parameters of startgame (set to true) and mypokemon
Request Type: POST
Returned Data Format: JSON
Description: The third request you will use initiates a game and passes two parameters, startgame and mypokemon to game.php. In contrast to the first two "GET" requests, this request is a "POST" request. Upon success, the request returns a JSON response of the initial game state (with information for both players' Pokemon) and unique game id (guid) and player id (pid) for the player to use to access and update the current game state.
Example Request:
POST parameters of startgame=true and mypokemon=pikachu
Example Output:

```
1  {
2    "guid" : "game_12345abc",
3    "pid" : "player_cfe67890",
4    "p1" : {
5      "name" : "Pikachu",
6      "hp" : 160,
7      "current-hp" : 160,
8      "images" : {
9        "photo" : "images/pikachu.jpg",
10       "typeIcon" : "icons/electric.jpg",
11       "weaknessIcon" : "icons/ground.jpg"
12     },
13     "info": {
14       "id": "25",
15       "type": "electric",
16       "weakness": "ground",
17       "description": "Melissa's favorite Pokemon! When several Pikachu gather, their electricity could
                build and cause lightning storms."
18     },
19     "moves": [
20       {
21         "name": "Growl",
22         "type": "normal"
23       },
24       {
25         "name": "Quick Attack",
26         "dp": 40,
27         "type": "normal"
28       },
29       {
30         "name": "Thunderbolt",
31         "dp": 90,
32         "type": "electric"
33       }
34     ],
35     "buffs": [],
36     "debuffs": []
37   },
38   "p2" : {
39     "name" : "Ditto",
40     "hp" : 206,
```

```
41      "current-hp" : 206,
42      "images": {
43       "photo": "images/ditto.jpg",
44       "typeIcon": "icons/normal.jpg",
45       "weaknessIcon": "icons/fighting.jpg"
46      },
47      "info": {
48       "id": "132",
49       "type": "normal",
50       "weakness": "fighting",
51       "description": "Duncan's favorite Pokemon (he has an awesome painting of Ditto on his wall). It can
                transform into anything. When it sleeps, it changes into a stone to avoid being attacked."
52      },
53      "moves": [
54       {
55        "name": "Transform",
56        "dp": 40,
57        "type": "normal"
58       }
59      ],
60      "buffs": [],
61      "debuffs": []
62    }
63 }
```

You may assume that the `guid` and `pid` attributes returned are unique to the started game.

Query 4: Play a Move

Request Format: game.php endpoint with POST parameters of `guid`, `pid`, and `movename`
Request Type: POST
Returned Data Format: JSON
Description: This query submits a move played by your Pokemon on the current turn and requires three parameters: `move` as your Pokemon's move name, `guid` as your unique game ID, and `pid` as your unique player id. The move name should be passed as an all-lowercase string, and if there are any spaces in the move name (e.g., "Quick Attack"), they should be removed when passed as a parameter (e.g., "Quick Attack" would be passed as "quickattack"). The game state is updated by applying that move's effects to either player (depending on the specific effects of the move). The request will also call the opponent's move, which may update the health or buffs of your Pokemon. Upon success, the request returns the current game state, including each player's current Pokemon status and the results of the two moves (yours and the opponents), as a JSON object.

An example return is given on the following page, where the `guid` provided is fictitious and you will need to provide the one retrieved from the previous `startgame` request:

Example Request:
POST parameters of `guid=game_12345abc`, `pid=player_cfe67890`, and `movename=quickattack`
Example Output:

```
1  {
2    "guid" : "game_12345abc",
3    "results" : {
4      "p1-move" : "Thunderbolt",
5      "p2-move" : "Transform",
6      "p1-result" : "hit",
7      "p2-result" : "miss"
8    },
9    "p1" : {
10     "name" : "Pikachu",
11     "type" : "electric",
12     "weakness" : "ground",
13     "hp" : 80,
14     "current-hp" : 60,
15     "moves" : [
16       {
17         "name": "Brick Break",
18         "dp" : 75,
19         "type" : "fighting"
20       },
21       {
22         "name": "Growl",
23         "type" : "normal"
24       },
25       {
26         "name": "Quick Attack",
27         "dp" : 40,
28         "type" : "normal"
29       },
30       {
31         "name": "Thunder",
32         "dp" : 80,
33         "type" : "electric"
34       }
35     ],
36     "buffs" : [],
37         "debuffs" : []
38   },
39   "p2" : {
40     "name" : "Ditto",
41     "type" : "normal",
42     "weakness" : "fighting",
43     "hp" : 60,
44     "current-hp" : 20,
45     "moves" : [
46       {
47         "name": "Transform",
48         "dp" : 40,
49         "type" : "normal"
50       }
51     ],
52     "buffs" : [],
53         "debuffs" : ["attack", "attack"]
54   }
55 }
```