Coding Challenge API Reference Guide

1. Application Logon - public String[] login(String user, String pass)

/\*\*

\* login - first method called by user, if the username and password combination

\* is correct, returns a token, security level, status string. If false, returns

\* null, null, and a failure message. Stores token and date stamp in order invalid token

\* after certain interval.

\*

\* @param user

\* @param pass

\* @return

\*/

2. List Incidents by ZIP - public ArrayList <Incident> geolocSearch(String token, int zip) throws TokenInvalidException

/\*\*

\* geolocSearch - method call used to return incidents based on passed in ZIP code

\*

\* @param token

\* @param zip

\* @return

\* @throws TokenInvalidException

\*/

3. Retrieve Incident by Record Number - public Incident incidByRecNum(String token, int recNum) throws TokenInvalidException

/\*\*

\* incidByRecNum - method to return an incident based on the Record Number passed in

\*

\* @param token

\* @param recNum

\* @return

\* @throws TokenInvalidException

\*/

4. Update Incident by Record Number - public String updateIncidentbyRecnum(String token, Incident newIncid) throws TokenInvalidException

/\*\*

\* updateIncidentbyRecnum - method that updates an incident based on its record number.

\* essentially overwrites previous incident

\*

\* @param token

\* @param newIncid

\* @return

\* @throws TokenInvalidException

\*/

5. Add Incident - public String addIncident(String token, Incident newIncid) throws TokenInvalidException

/\*\*

\* addIncident - method that adds an incident to the records. Note: this method assigns a Record Number to the new incident (one greater than the highest current incident number in the records.)

\*

\* @param token

\* @param newIncid

\* @return

\* @throws TokenInvalidException

\*/

6. Get Admin Condition Report Data - public ArrayList<Incident> getAdminConditionCannedReport(String token, String eventName, int eventYear) throws TokenInvalidException

/\*\*

\* getAdminConditionCannedReport - a wrapper that returns report data, from a BE perspective

\*

\* @param token

\* @param eventName

\* @param eventYear

\* @return

\* @throws TokenInvalidException

\*/

7. Get CRE Building Closure/Delayed Opening Data - public ArrayList<Incident> getCREReport(String token, String eventName, int eventYear) throws TokenInvalidException

/\*\*

\* getCREReport - a wrapper that returns report data, from a BE perspective

\*

\* @param token

\* @param eventName

\* @param eventYear

\* @return

\* @throws TokenInvalidException

\*/

8. Reset data - public void resetData();

/\*\*

\* resetData - method the contestants can call to reset the data on their app to the default \* values in case issues arise. should be solely used for dev and test, should not be used by the teams in their final code.

                \*

                \*/

9. Get list of GEOLOCS – public Hashtable<Integer, String> getGLCInfo();

/\*\*

\* getGLCInfo - Get the list of existing GEOLOCs.

\*

\* @return Hashtable with each entry containing an Integer for the GEOLOC and a String for the description

\*/