



# Java Plays – Mobile Starter Kit - Starter (Extend)

**Presented by:**

IBM

IBM **Cloud**

# Mobile Starter Kit

## Starter

- Basic Mobile Application (Android)
- Basic Bluemix Java Application – REST Services

Mobile Application (Android)



Bluemix Java Application



REST

USE CASE – A Cognitive health care solution that provides information about a disease or a symptom

## Extend Bluemix Java App

- Extend the Basic RESTful Service to perform real time operations

Extend Bluemix Java Application

Publish REST Service (Liberty & Java)



Bluemix Services

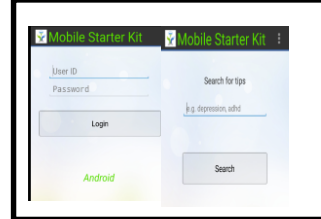


## Extend Mobile Application

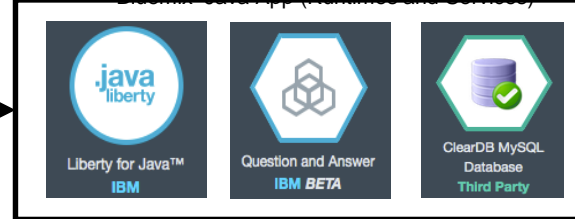
- Extend the Mobile App to Integrate with the REST Services (Login, Searchtips)

Extend Mobile Application (Android)

Mobile Application



Bluemix Java App (Runtimes and Services)



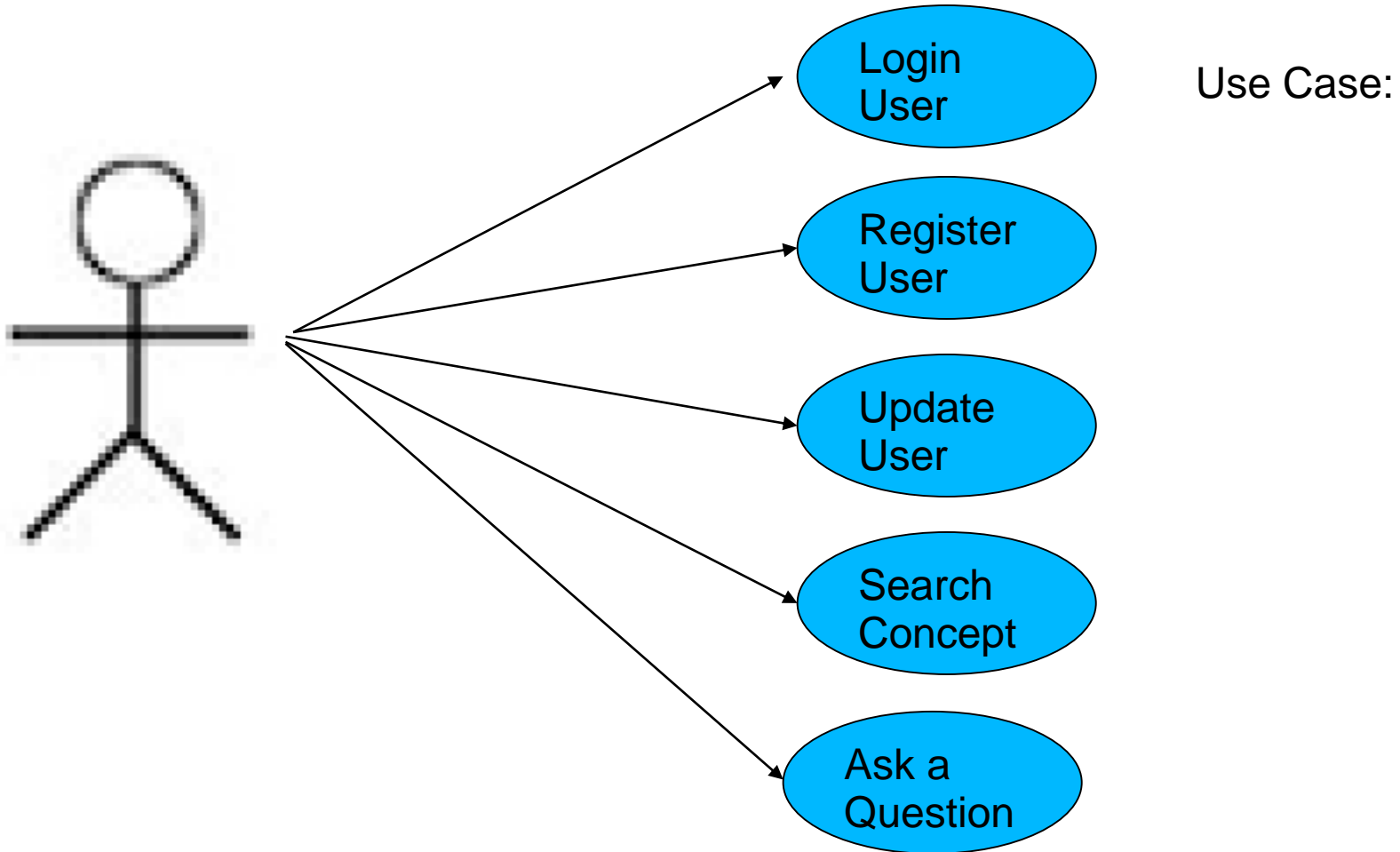
REST

## Agenda

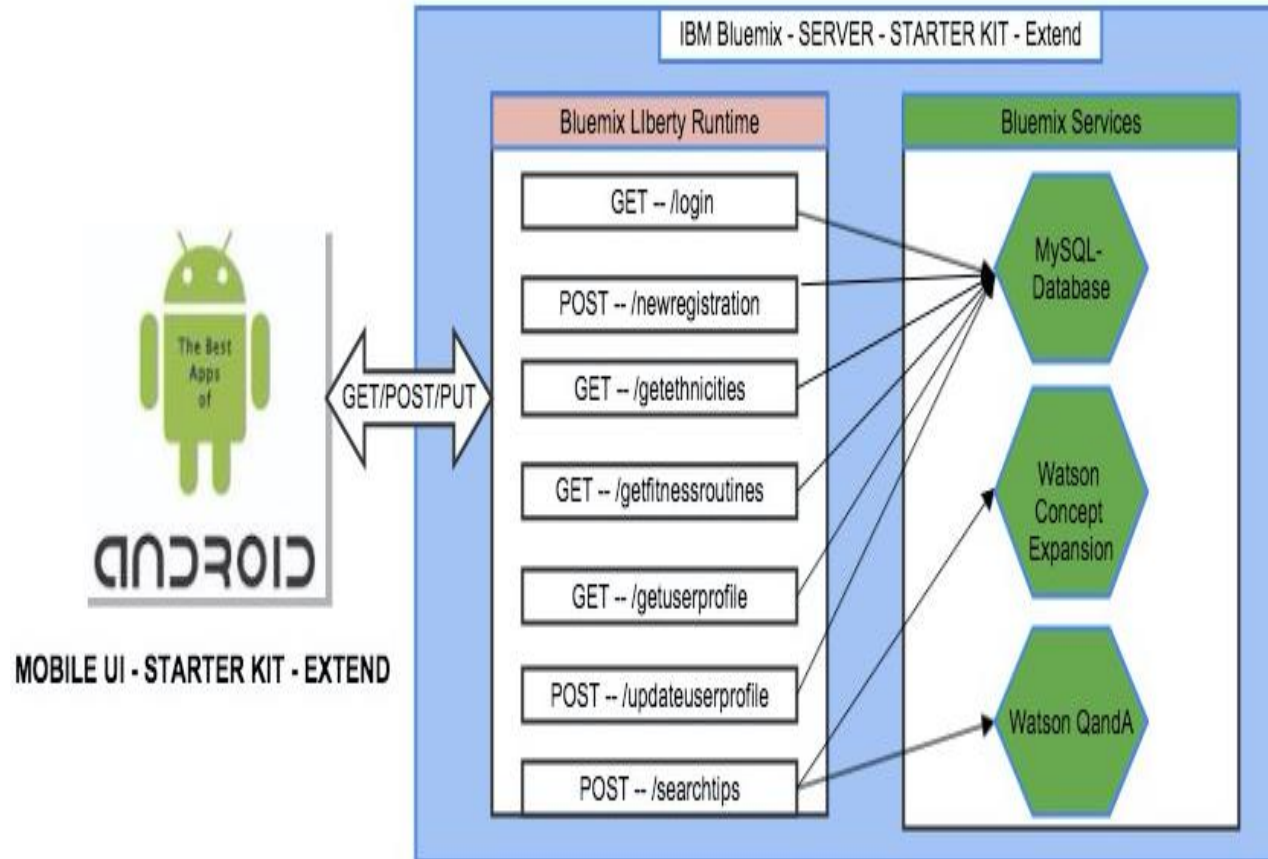
- **Extend Bluemix Java Application**
  - **Use Case and Application Architecture**
  - **Create Bluemix Services**
  - **Connect to Bluemix Services**
    - **Read From VCAP**
    - **Connect to Services**
  - **Create REST Service Calls for Use Case**
    - **login**
    - **register**
    - **searchtips**

- **Extend Mobile UI (Application)**
  - **Mobile Application Design**
  - **Mobile Application UI Flow**
  - **Task Execution Protocol**
  - **Integrating Components**

## Extend Bluemix Java Application – Use Case



## Extend Bluemix Java Application – Application Architecture



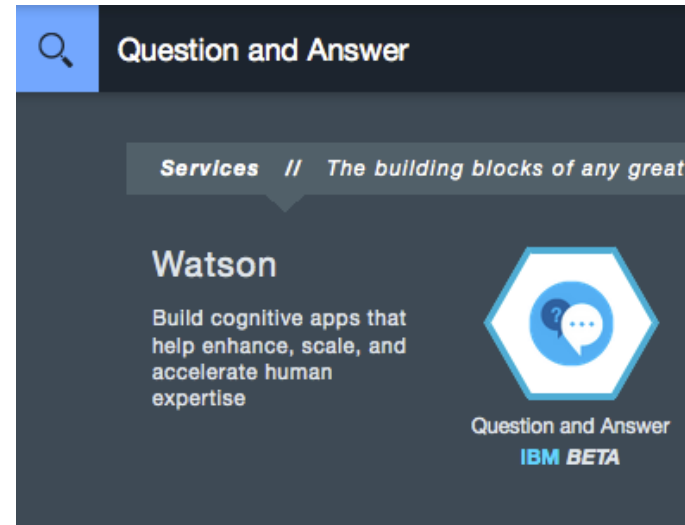
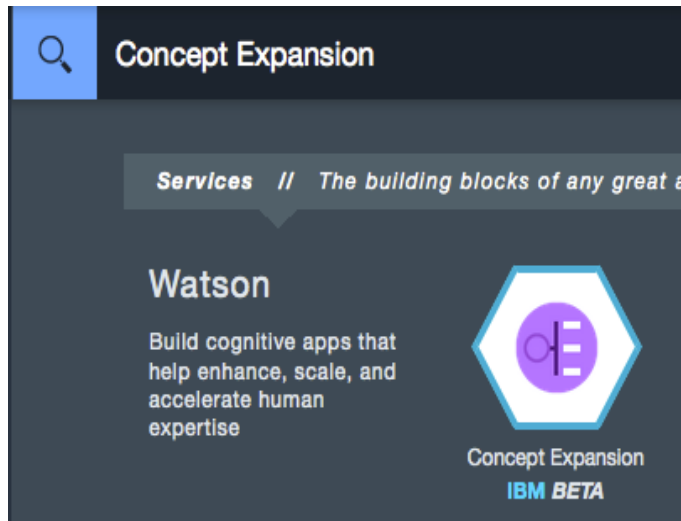
- There are two parts for this application
- Client Side is a Android Mobile Application that is used to retrieve and display information of the disease or symptom searched by a user.
- Server Side is a Bluemix Java application that publishes required information via RESTful API's invoking backend Bluemix Cognitive Services. These RESTful API calls provide Android Mobile Application with the information about the disease or symptom requested by the user.

## Extend Bluemix Java Application – Pull the code

Pull the code rest services api java code from the jazzhub git repository.

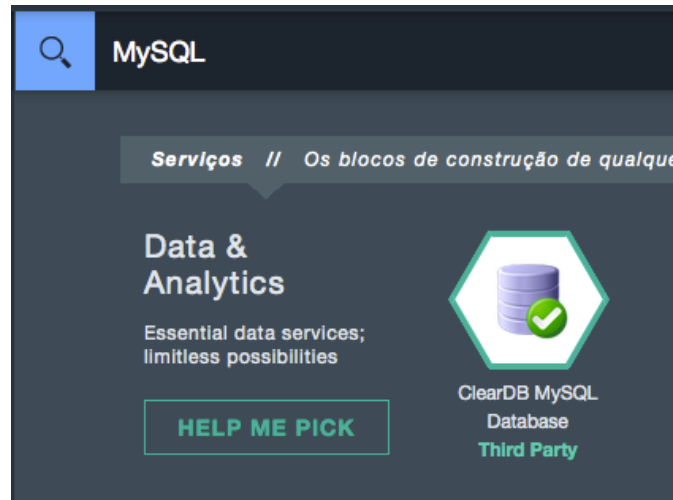
- <https://hub.jazz.net/git/ecosysdevcnc/javaplays-mobilestarterkit/>
  - javaplays-restjavafull-mobilestarterkit

## Extend Bluemix Java Application – Create Bluemix Services



### Create 3 Bluemix Services

- Login to Bluemix → Catalog → Search for Service → Create
  - Watson Concept Expansion Service
  - Watson Q and A Service
  - Clear MySQL DB Service



# Bluemix Java Application – Read VCAP Environment Variables

## ▪ ServiceDiscovery.java – Read VCAP environment variables

```
public void processVCAP() {  
  
    String MySQL_Service_Name = "cleardb";  
    String WatsonConceptExpansion_Service_Name = "concept_expansion";  
    String WatsonQandA_Service_Name = "question_and_answer";  
  
    // VCAP_SERVICES is a system environment variable  
    // Parse it to obtain the for DB2 connection info  
    String VCAP_SERVICES = System.getenv("VCAP_SERVICES");  
    if (VCAP_SERVICES != null) {  
        // parse the VCAP JSON structure  
        BasicDBObject obj = (BasicDBObject) JSON.parse(VCAP_SERVICES);  
        String thekey = null;  
        Set<String> keys = obj.keySet();  
        // Look for the VCAP key that holds the SQLDB information  
        for (String eachkey : keys) {  
            // Just in case the service name gets changed to lower case in  
            // the future, use toUpperCase  
            if (eachkey.contains(MySQL_Service_Name)) {  
                thekey = eachkey;  
                getDBConnectionParams(thekey, obj);  
            } else if (eachkey  
                .contains(WatsonConceptExpansion_Service_Name)) {  
                thekey = eachkey;  
                getWatsonConceptExpansionParams(thekey, obj);  
            } else if (eachkey.contains(WatsonQandA_Service_Name)) {  
                thekey = eachkey;  
                getWatsonQandAParams(thekey, obj);  
            }  
        }  
    }  
}
```

Service Discovery class → `com.ibm.bluemix.startkit.services`

- The process VCAP method in the service class reads all the VCAP environment various defined for that application.
- There are three services configured in Bluemix for this application
  - ClearDB MySQL
  - Concept Expansion
  - Question and Answer



# Bluemix Java Application – Connect to Services

- Service Discovery.java – Read Bluemix Services environment variables

```
private void getWatsonQandAParams(String qakey, BasicDBObject vcapobj) {  
  
    BasicDBList qadblist = (BasicDBList) vcapobj.get(qakey);  
    vcapobj = (BasicDBObject) qadblist.get("0");  
    // parse all the credentials from the vcap env variable  
    vcapobj = (BasicDBObject) vcapobj.get("credentials");  
  
    setQAURL((String) vcapobj.get("url"));  
    setQAUserName((String) vcapobj.get("username"));  
    setQAPassword((String) vcapobj.get("password"));  
  
}  
  
private void getWatsonConceptExpansionParams(String cekey,  
    BasicDBObject vcapobj) {  
  
    BasicDBList cedblist = (BasicDBList) vcapobj.get(cekey);  
    vcapobj = (BasicDBObject) cedblist.get("0");  
    // parse all the credentials from the vcap env variable  
    vcapobj = (BasicDBObject) vcapobj.get("credentials");  
    setCEURL((String) vcapobj.get("url"));  
    setCEUserName((String) vcapobj.get("username"));  
    setCEPassword((String) vcapobj.get("password"));  
  
}  
  
@SuppressWarnings("unchecked")  
public void getDBConnectionParams(String mysqlbkey, BasicDBObject vcapobj) {  
  
    BasicDBList mysqlblist = (BasicDBList) vcapobj.get(mysqlbkey);  
    vcapobj = (BasicDBObject) mysqlblist.get("0");  
    // parse all the credentials from the vcap env variable  
    vcapobj = (BasicDBObject) vcapobj.get("credentials");  
    setDBHostName((String) vcapobj.get("hostname"));  
    setDBName((String) vcapobj.get("name"));  
    setDBPort((String) vcapobj.get("port"));  
    setDBUser((String) vcapobj.get("username"));  
    setDBPassword((String) vcapobj.get("password"));  
  
}
```

- getWatsonQandAParams → This method retrieves all the VCAP environment variables available for Watson Q and A Service in Bluemix
- getWatsonConceptExpansionParams → This method retrieves all the VCAP environment variables available for Watson Concept Expansion Service in Bluemix
- getDBConnectionParams → This method retrieves all the VCAP environment variables available for Clear MySQL DB Service in Bluemix

## Bluemix Java Application – Login REST Service

### ▪ login REST Service

```
@Path("/login")
@POST
public String checkLogin(String creds) {
    DBHandler dbService = null;
    UserProfileTableHandler userTableHandler = null;
    try {

        dbService = new DBHandler(_dbHost, _dbPort, _dbUser, _dbPassword,
                                   _dbName);
        userTableHandler = new UserProfileTableHandler();
        JSONObject credentials = new JSONObject(creds);
        String userID = credentials.getString("user_id");
        String password = credentials.getString("password");
        boolean checkResult = userTableHandler.isRegistered(userID, password, dbService);
        dbService.closeConnection();
        if (checkResult) {
            return "Successful";
        } else {
            return "Failed";
        }
    }
}
```

- ServiceAPI.java → This class defines all the REST Services for this application
- /login REST Call →
  - Application receives the credentials from the Mobile UI in a JSON format.
  - Parses the credentials
  - Connects to the Clear SQL MyDB Service
  - Perform a select query on the login table to validate the credentials
  - Returns a success or failure based on the response

## Bluemix Java Application – Login REST Service (Cont'd)

- login REST Service

```
public boolean isRegistered( String userID, String password, DBHandler dbService ){  
  
    String queryString = "SELECT users.name FROM users WHERE (";  
    queryString = queryString + "user_id='" + userID + "'";  
    queryString = queryString + " AND " ;  
    queryString = queryString + "password='" + password + "'";  
    queryString = queryString + ")";  
  
    try {  
        ResultSet userList = dbService.runSelectQuery( queryString );  
  
        if( userList.next() ){  
            return true;  
        }  
        else{  
            return false;  
        }  
    }  
}
```

- /login REST Call →
  - Application receives the credentials from the Mobile UI in a JSON format.
  - Parses the credentials
  - Connects to the Clear SQL MyDB Service
  - Perform a select query on the login table to validate the credentials
  - Returns a success or failure based on the response

# Bluemix Java Application – Register New User REST Service

## ▪ Register new user REST Service

```
@Path("/registernew")
@POST
public String registernew(String registrationDetail) {

    DBHandler dbService = null;
    UserProfileTableHandler userTableHandler = null;
    try {
        dbService = new DBHandler(_dbHost, _dbPort, _dbUser, _dbPassword, _dbName);
        userTableHandler = new UserProfileTableHandler();
        JSONObject registration = new JSONObject(registrationDetail);
        String userID = registration.getString("user_id");
        String password = registration.getString("password");
        String name = registration.getString("name");
        int age = registration.getInt("age");
        String zip = registration.getString("zip");
        String ethnicity = registration.getString("ethnicity");
        String fitness = registration.getString("fitness");
        String profession = registration.getString("profession");
        String interests = registration.getString("interests");

        boolean checkResult = userTableHandler.isRegistered(userID, dbService);
        if (checkResult) {
            if (dbService != null) {
                try {
                    dbService.closeConnection();
                } catch (SQLException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }
            }
        }
    }
}
```

## • /registernew REST Call →

- Application receives the user information from the mobile UI in a JSON format.
- Performs an insert user in to user table in the database
- Returns a success or failure based on database insert response

# Bluemix Java Application – Register New User REST Service

## ▪ Register new user REST Service

```
public boolean insertUser( UserProfile newUser, DBHandler dbService ){

    boolean isAlreadyRegistered = isRegistered( newUser.getID(), dbService );
    if( isAlreadyRegistered ){
        return true;
    }

    int ethnicityID;
    int fitnessID;
    try {
        ethnicityID = EthnicityTableHandler.getEthnicityID( newUser.getEthnicity() , dbService );
        fitnessID = FitnessTableHandler.getFitnessID ( newUser.getFitnessRoutine(), dbService )

        if( ethnicityID == -1 || fitnessID == -1 ){
            return false;
        }

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();

        return false;
    }

    String queryString = "INSERT INTO users (user_id, password, name, age, zip_code, fk_ethnicity, fk";
    queryString = queryString + "'" + newUser.getID() + "'" + ",";
    queryString = queryString + "'" + newUser.getPassword() + "'" + ",";
    queryString = queryString + "'" + newUser.getName() + "'" + ",";
```

- /registernew REST Call
  - Application receives the user information from the Mobile UI in a JSON format.
  - Performs a insert user in to user table in the database
  - Returns a success or failure based on database insert response

# Bluemix Java Application – Register New User REST Service

## ▪ searchtips REST Service

```
@Path("/searchtips")
@POST
public String searchTips( String params ){

    org.json.JSONObject paramObj = null;

    SearchTipsTask csd = new SearchTipsTask( );

    csd.createCEService( _ceBaseURL, _ceUsername, _cePassword );
    csd.createQAService( _qaBaseURL, _qaUsername, _qaPassword );

    try{

        paramObj = new JSONObject( params );

        String keyword      = paramObj.getString( "keyword" );
        String searchResult = csd.runSteps( keyword );

        //System.out.println( searchResult );
        return searchResult;

    }
```

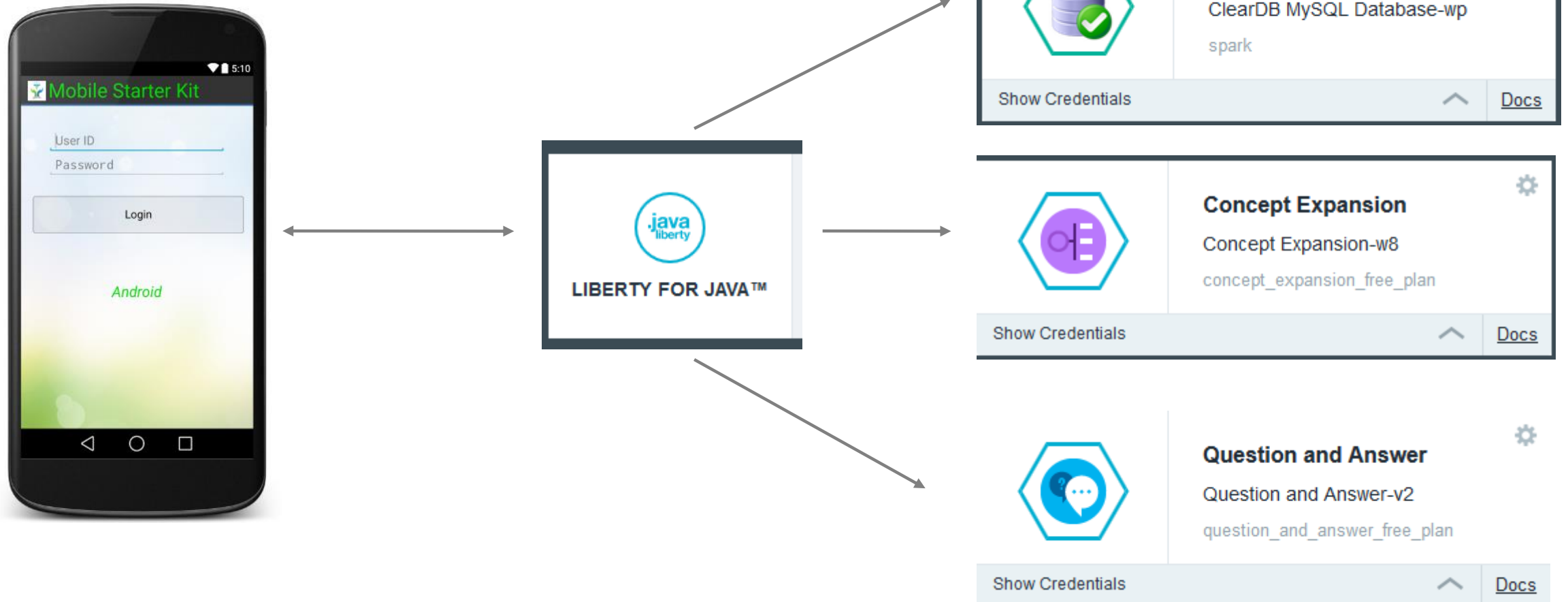
## • /searchtips REST Call →

- This rest call communicates with Watson Concept Expansion and Watson Q and A service for the input passed in to the mobile application search screen
- This call parses and converts the output from the watson services to JSON format
- The JSON format output will be passed as an output to the /searchtips rest call.
- The results are displayed on the Mobile UI Search screen.



# **Extend Mobile Application – Using Full Power of Bluemix**

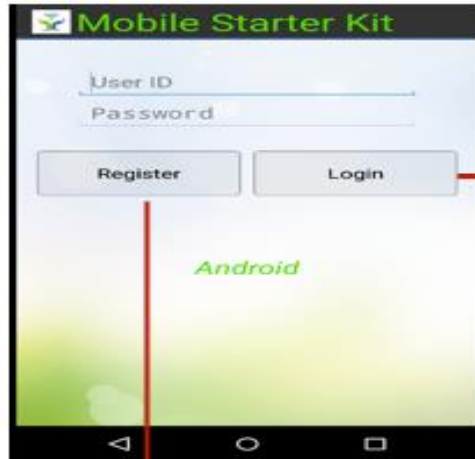
## Light UI – Heavy Server side





# Mobile UI Flow

HOME SCREEN



Mobile Starter Kit

User ID  
Password

Register Login

Android

SEARCH DISEASE OR SYMPTOM SCREEN



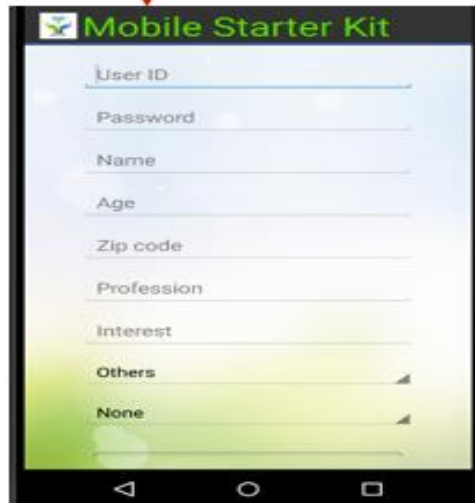
Mobile Starter Kit

Search for tips

depression

Search

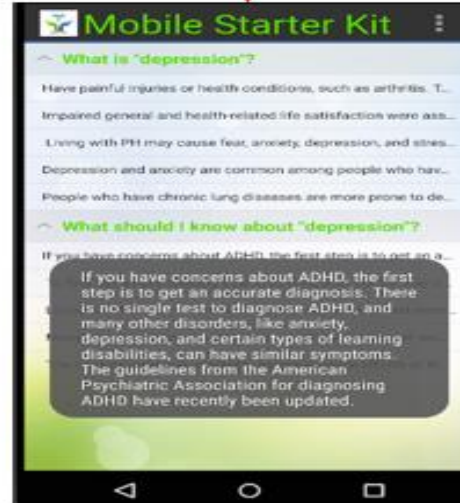
REGISTER SCREEN



Mobile Starter Kit

User ID  
Password  
Name  
Age  
Zip code  
Profession  
Interest  
Others  
None

SEARCH RESULT SCREEN



Mobile Starter Kit

What is "depression"?

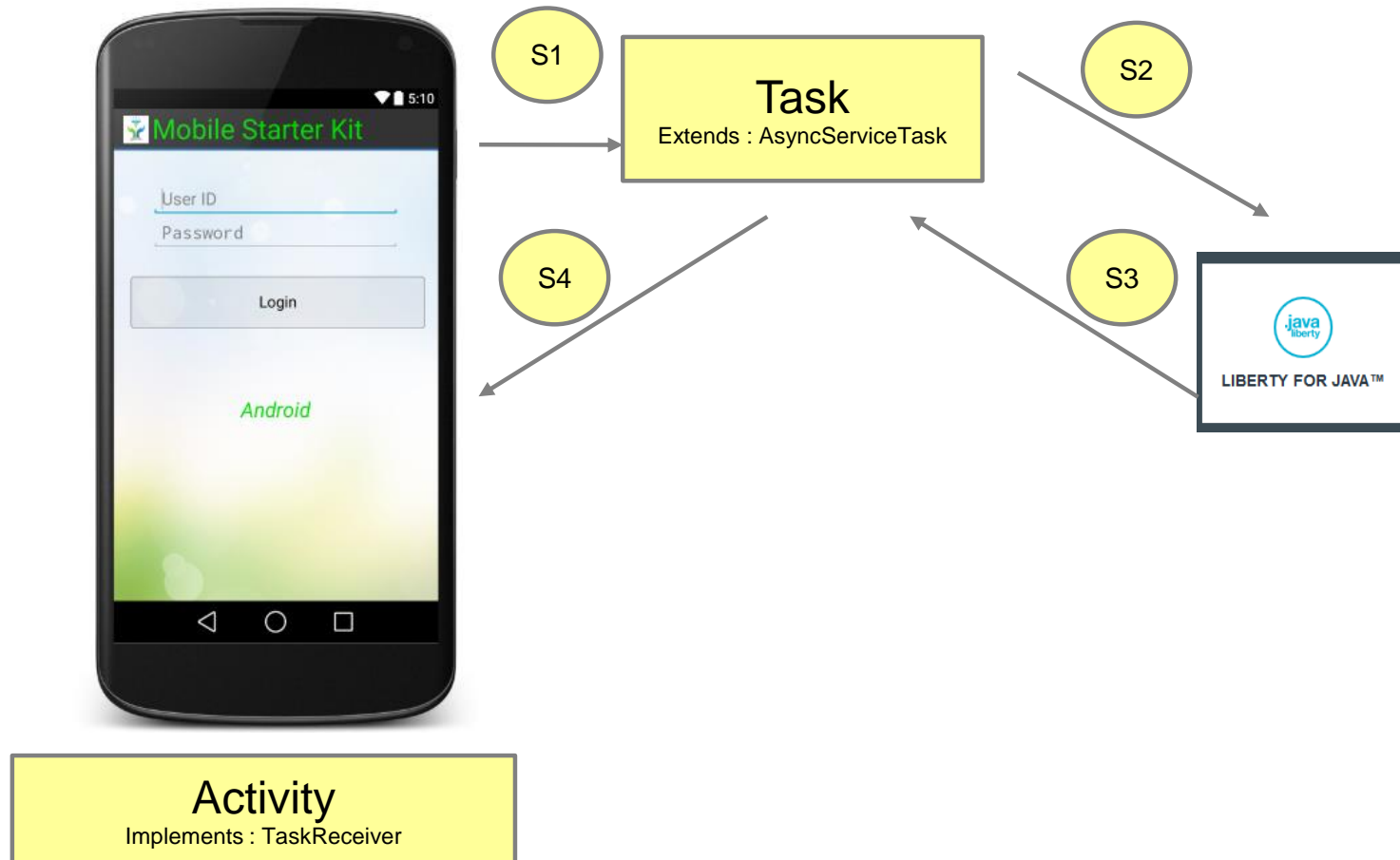
Have painful injuries or health conditions, such as arthritis. T...  
Impaired general and health-related life satisfaction were ass...  
Living with PH may cause fear, anxiety, depression, and stres...  
Depression and anxiety are common among people who hav...  
People who have chronic lung diseases are more prone to de...

What should I know about "depression"?

If you have concerns about ADHD, the first step is to get an a...

If you have concerns about ADHD, the first step is to get an a...  
There is no single test to diagnose ADHD, and many other disorders, like anxiety, depression, and certain types of learning disabilities, can have similar symptoms. The guidelines from the American Psychiatric Association for diagnosing ADHD have recently been updated.

## Task Execution Protocol – Step by step



- S1 :
  - Activity creates the task and assigns an ID to it
  - Activity registers itself as a listener to the postExecution event of the task
  - Activity starts the task in a separate thread
- S2 :
  - Task executes a POST method and waits for the result
- S3 :
  - The response is received by the task thread
  - Task raises a postExecute() event and passes the response to any listener
- S4 :
  - Activity responds to the event and process the result

# Task Execution Protocol – Step by step

```
//SearchUI.java - if the user clicks the button
{
    ...
    SearchTipsTask stt = new SearchTipsTask( ... );
    stt.setReceiver(this);
    stt.execute(_seeds);
    ...
}
```

S1

```
//AsyncServiceTask.java
@Override
protected String doInBackground( String... params )
{
    //performTask is implemented by the
    //SearchTipsTask class
    return performTask( params );
}
```

S2



S3

```
@Override
public void receiveResult( String response,
                          String source ) {
    {
        //make sure the task ID is the search task
        if( source.equals( correct_id ) ){
            //perhaps do something with the response
        }
    }
}
```

S4

```
//AsyncServiceTask.java
@Override
protected void onPostExecute( String result ){
    _response = result;
    if( _tr != null ){
        //calls the receiver's receiveResult function
        //adds the serviceID here
        //in this case the receiver is SearchUI.java
        _tr.receiveResult( _response, _serviceID);
    }
}
```

## How are all these stitched together? - without headache

- com.ibm.android.asyncTask.TaskReceiver - Interface
  - Every activity that required to use a REST API implements this class
  - This class enforces the activities to declare a receiveResult function

```
public interface TaskReceiver{  
    public void receiveResult( String response, String source );  
}
```

String response

- The response received from REST call

String Source

- The ID of the asynchronous task generating the response

```
//Example : SearchUI.java  
public void receiveResult( String response, String source ) {  
  
    //make sure the task ID is the search task  
    if( source.equals( SearchTipsTask._SEARCH_DOC_ID ) ){  
  
        //do something with the response  
    }  
}
```

## How are all these stitched together? - without headache

- com.ibm.android.asyncntask.AsyncServiceTask - Abstract
  - Every task that calls REST API extends this class
  - This class enforces the tasks to define a REST API call

```
public abstract String performTask( String... params)  
    throws IOException, JSONException, URISyntaxException;
```

String... params

- Input parameter needed for the task

```
//Example : SearchTipsTask  
@Override  
protected String performTask(String... params) throws IOException,  
    JSONException, URISyntaxException {  
  
    //retrieve the parameter  
    String keyword = params[0];  
  
    //POST and return the response  
    return performPost( ... );  
}
```

# Questions?