IBM VAADIN CHALLENGE

LAB → VAADIN – BLUEMIX – WATSON SAMPLE APP

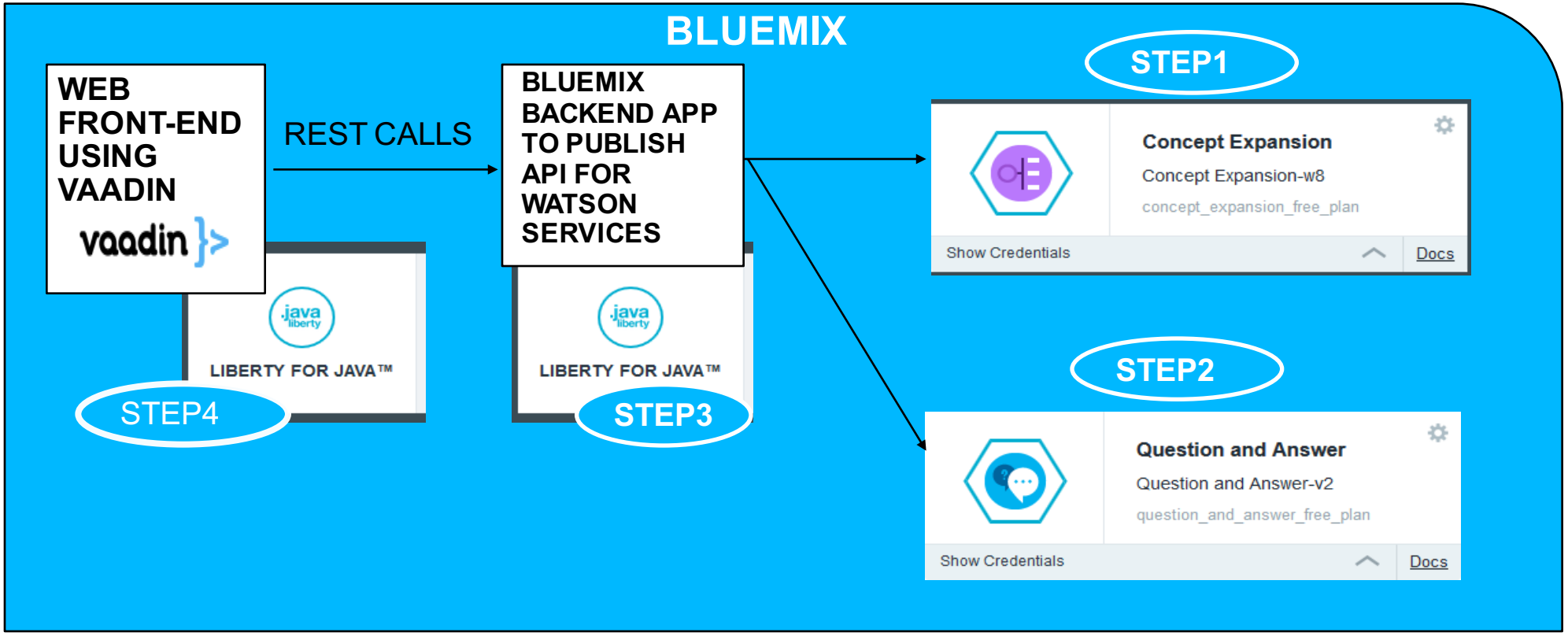
Presented by:

IBM EcoD

IBM Cloud

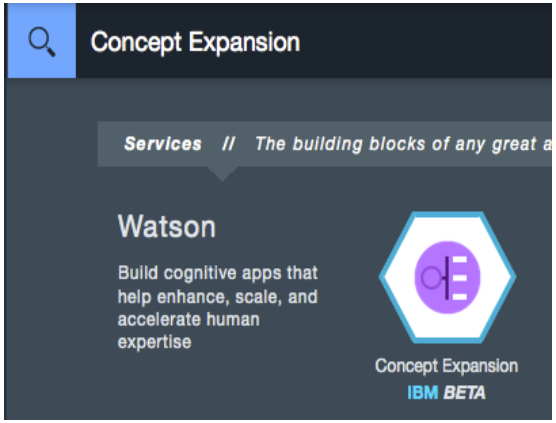
VAADIN – BLUEMIX – WATSON HEALTHCARE SAMPLE APP

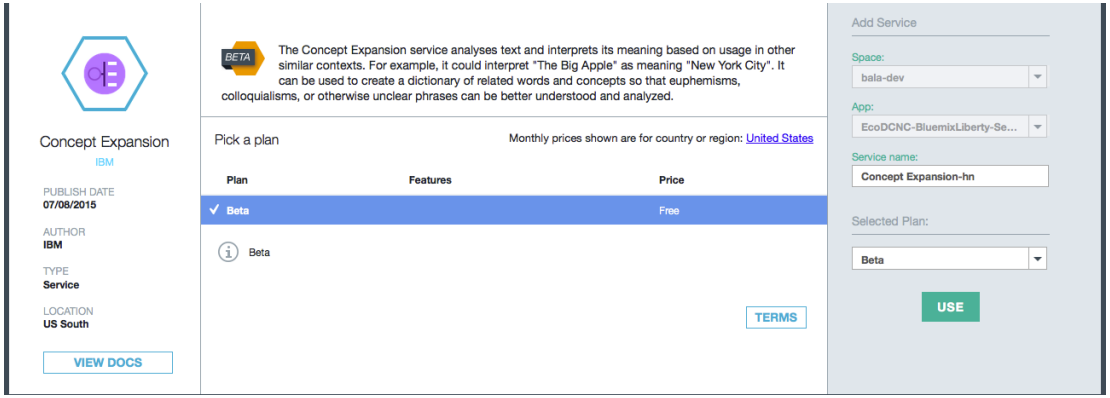
VAADIN – BLUEMIX – WATSON SAMPLE APP → LOGICAL FLOW → STEPS FOR THE LAB



STEP 1 – Create Watson Concept Expansion Service in Bluemix

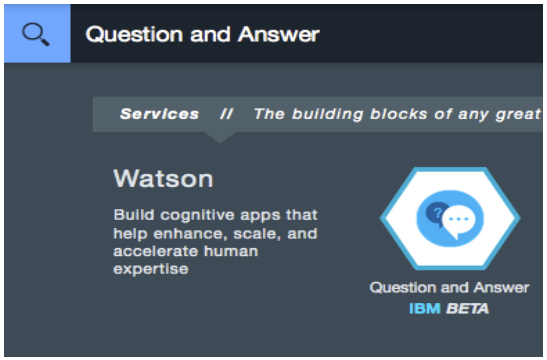
- Login to Bluemix → Select Catalog → Search Concept Expansion→ Click Create

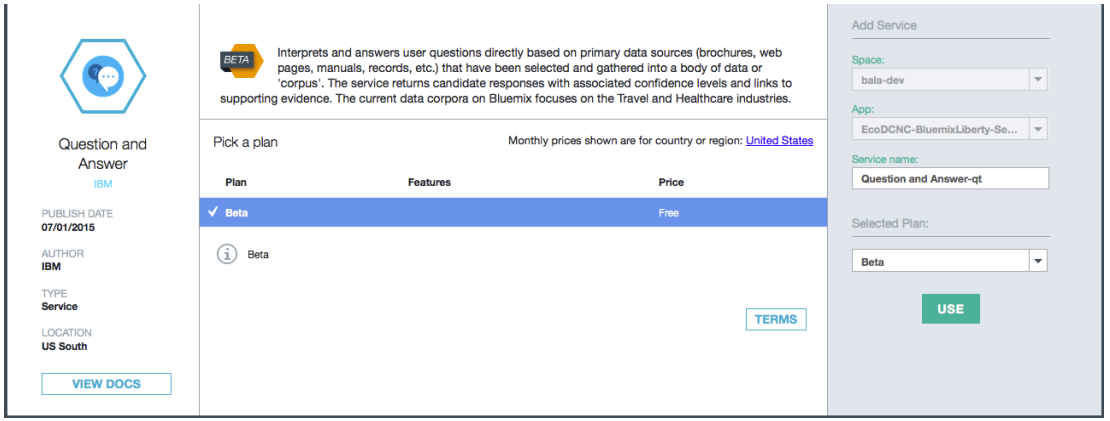




STEP 2 – Create Watson Question and Answer Service in Bluemix

- Login to Bluemix → Select Catalog → Search Watson Q and A→ Click Create



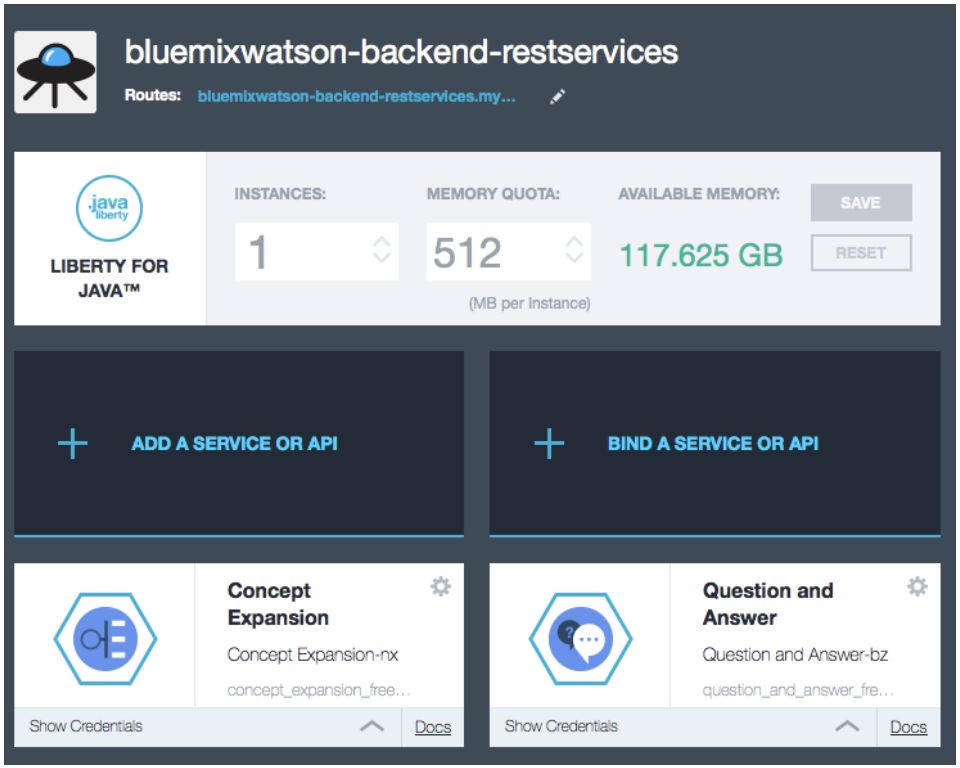


STEP 3 – CREATE AND DEPLOY BLUEMIX RESTFUL SERVICES WATSON APP via COMMAND LINE

- Download the war file from
  - https://github.com/ibmecod/javaplays-rest-watsonservices/blob/master/jprestjavafullmsk.war
- Login to Bluemix via cmdline
  - cf login <endpoint> (Example:https://api.ng.bluemix.net)
  - Select the space in Bluemix
- Push the application to Bluemix
  - cf push <appname> -p jprestjavafullmsk.war (appname → can be anyname you want to name your application. Example : bluemixwatson-backend-restservices)

STEP 3 – DEPLOY BLUEMIX RESTFUL SERVICES WATSON APP via COMMAND LINE – Cont'd

Open Bluemix Admin Console → Click on the application Deployed →Bind the Services configured in Step1 and 2 → For this application



STEP 3 – DEPLOY BLUEMIX RESTFUL SERVICES WATSON APP via COMMAND LINE – Cont'd

- Get the name of the route from Bluemix after the application is started → Ready TO GO !!!
- Note down the URL for use in Step 4
  - Sample Application URL for Backend - http://bluemixwatson-backend-restservices.mybluemix.net/
- Review, Change, Deploying the code → Follow the steps in Appendix A and B sections below

STEP 4 – CREATE AND PUSH VAADIN USER INTERFACE APPLICATION TO BLUEMIX

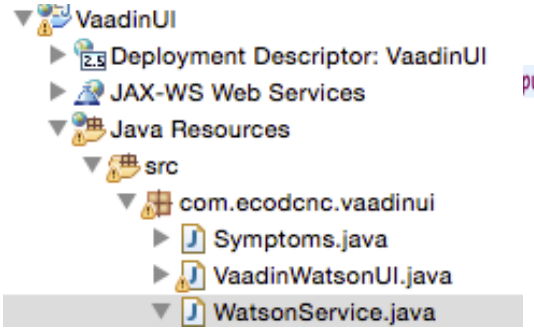
- Download the war file from
  - https://github.com/ibmecod/vaadin/blob/master/vaadin-bluemixwatson.war
- Login to Bluemix via cmdline
  - cf login <endpoint>, Select the space in Bluemix
- Push the application to Bluemix
  - cf push <appname> -p vaadin-bluemixwatson.war
- Get the name of the route from Bluemix after the application is started → Ready TO GO !!!

http://<appname>.mybluemix.net

Sample Application URL - http://vaadin-watson-healthcare.mybluemix.net/

STEP 4 (Not Mandatory) – Review, Change and Deploy the Code

- Import the code from Github
  - https://github.com/ibmecod/vaadin.git → vaadinUI-bluemixwatson-app
- Import the project in to Eclipse à Open WatsonServices.java → Update the URL with BackEnd REST Services URL mentioned in Step 3



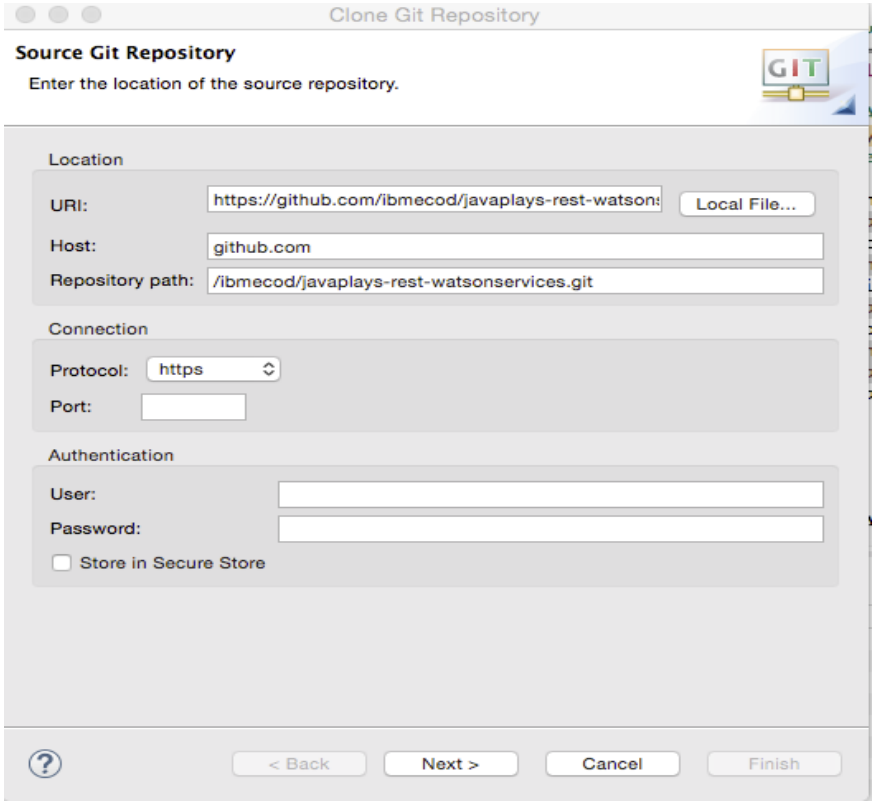

```
public class WatsonService {

    private static final String targetURL = "http://bluemixwatson-backend-restservices.mybluemix.net/api/service/searchtips/";
    String response;

    public String getResponseFromBluemixWatson(String forminput) {
            try {
```

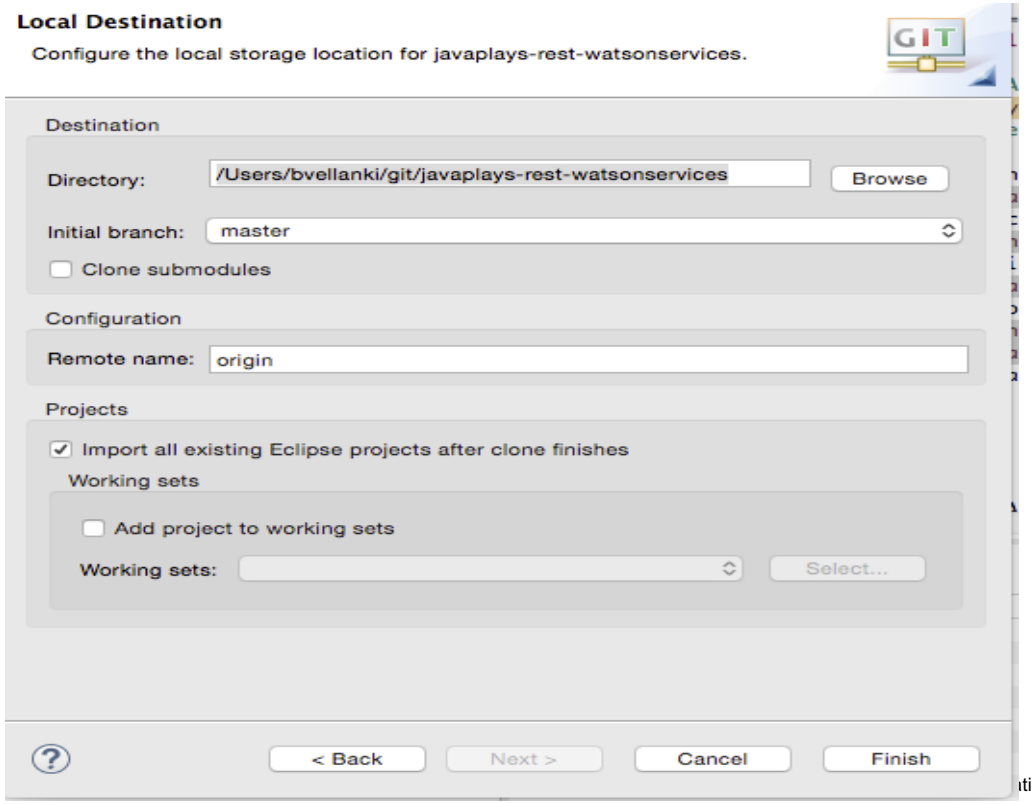
- Code Walkthrough → Vaadin UI → Appendix C

APPENDIX A – BLUEMIX BACKEND APP TO PUBLISH RESTFUL API FOR WATSON – PULL THE CODE TO ECLIPSE

APPENDIX A - BLUEMIX BACKEND APP TO PUBLISH RESTFUL API FOR WATSON

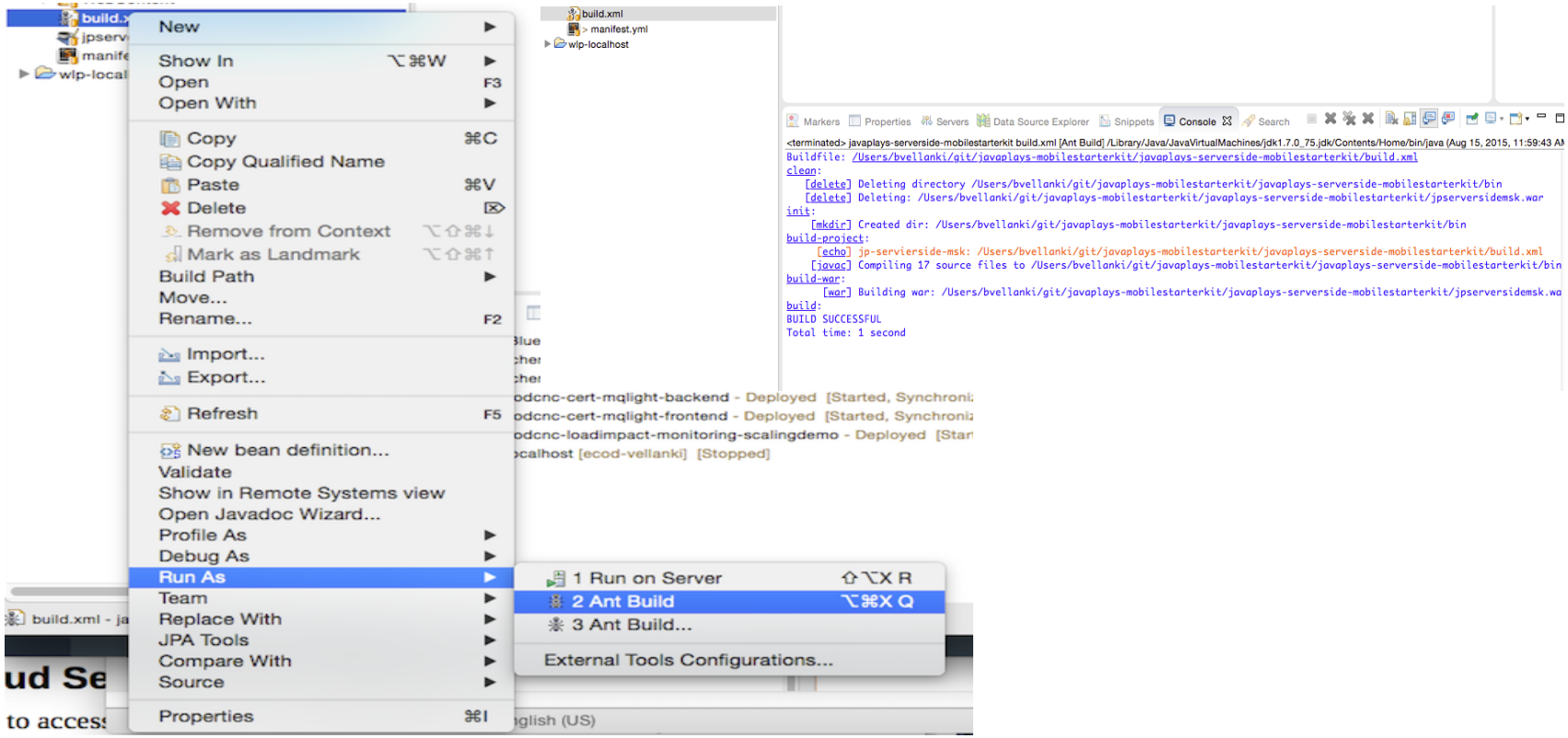
- **Import the Code from Git in to your eclipse workspace for development**
  - https://github.com/ibmecod/javaplays-rest-watsonservices.git





APPENDIX A - BLUEMIX BACKEND APP TO PUBLISH RESTFUL API FOR WATSON

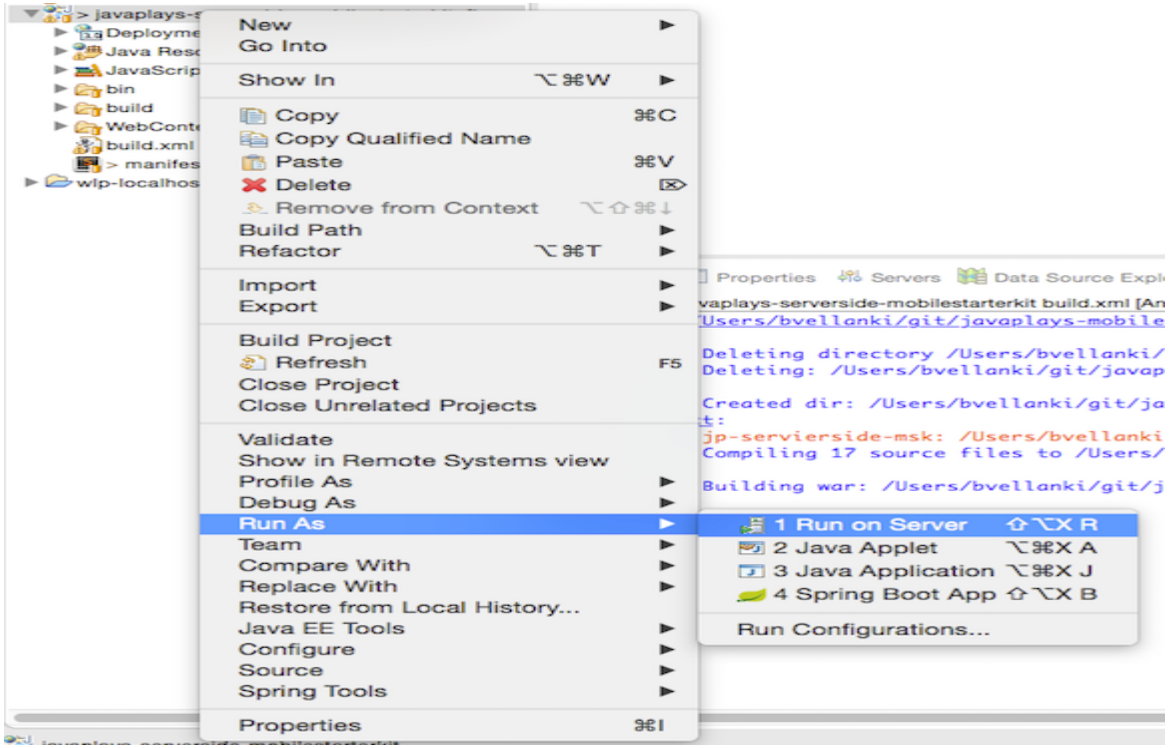
- Build.xml →Run the Ant Build



APPENDIX A - BLUEMIX BACKEND APP TO PUBLISH RESTFUL API FOR WATSON

STEP 1:

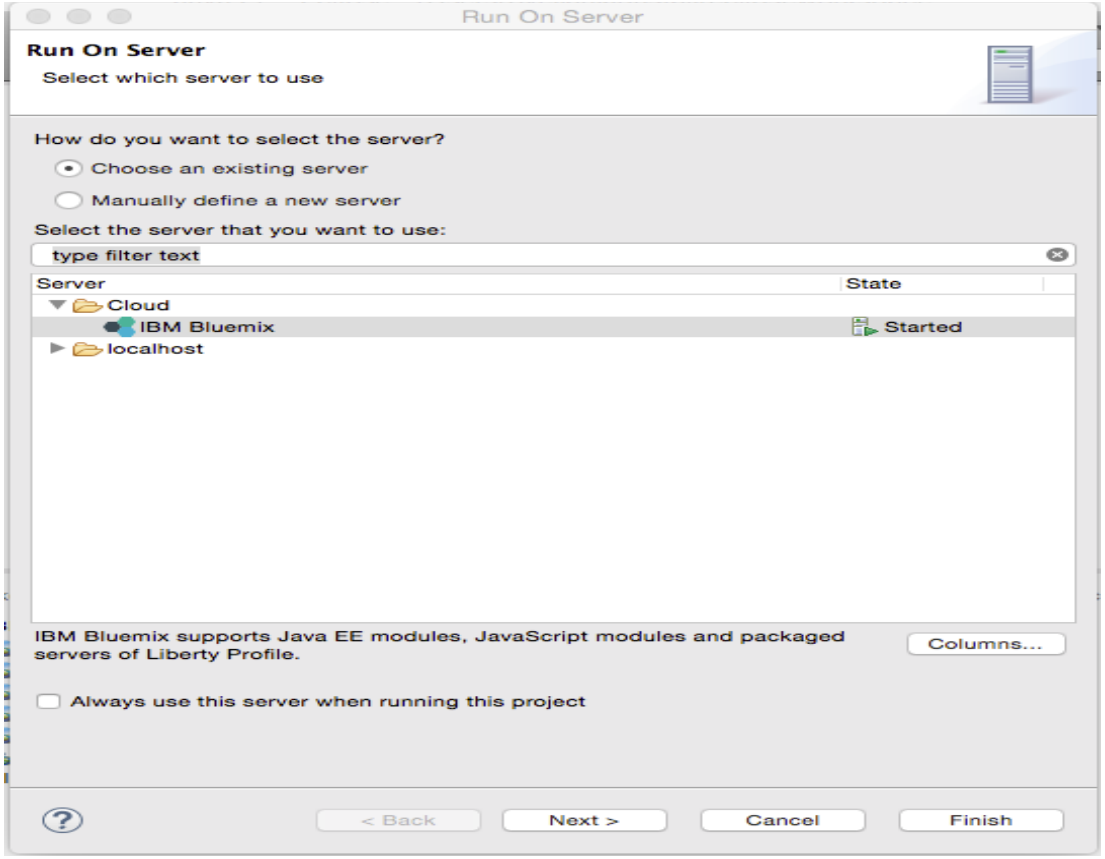
Deploy the application to Bluemix from Eclipse



APPENDIX A - BLUEMIX BACKEND APP TO PUBLISH RESTFUL API FOR WATSON

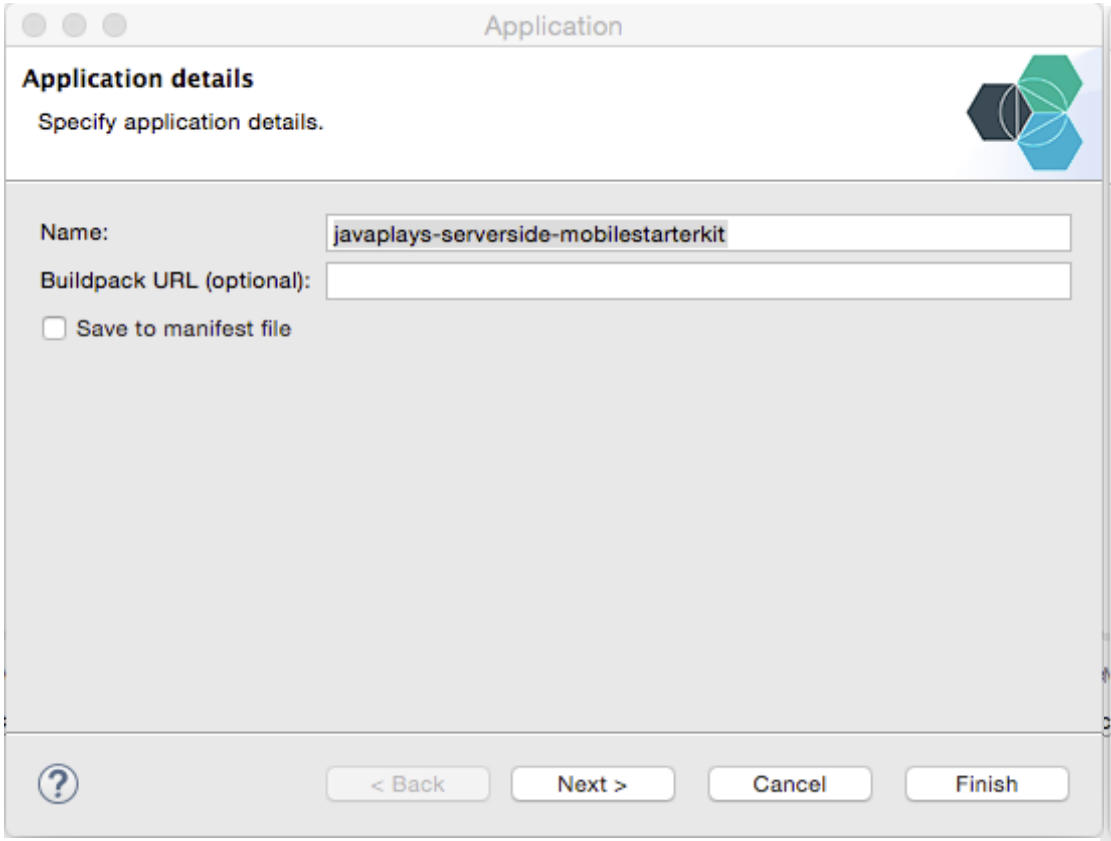
STEP 2:

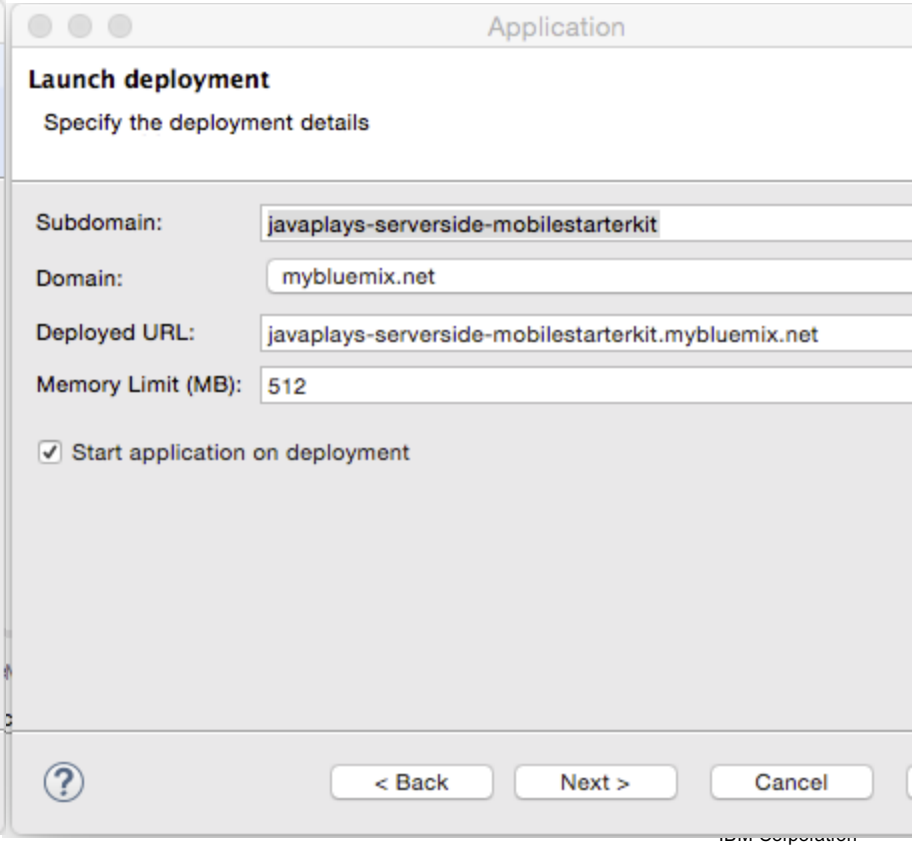
Deploy the application to Bluemix from Eclipse



APPENDIX A - BLUEMIX BACKEND APP TO PUBLISH RESTFUL API FOR WATSON

STEP 3: Deploy the application to Bluemix from Eclipse – Step3

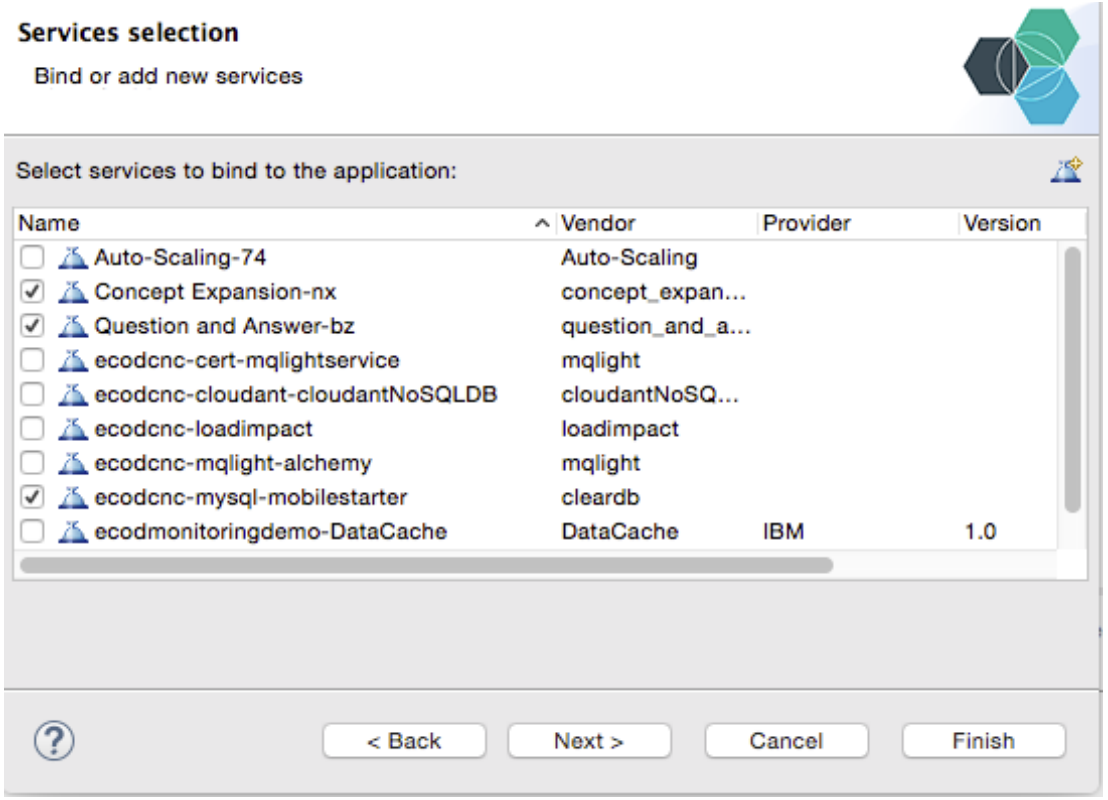




APPENDIX A - BLUEMIX BACKEND APP TO PUBLISH RESTFUL API FOR WATSON

STEP 4:

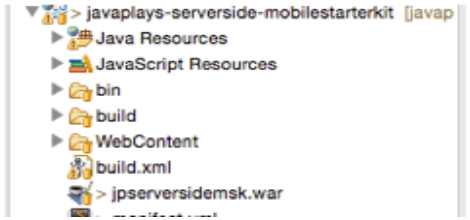
Deploy the application to Bluemix from Eclipse – Configure Services for the application → Select Services shown



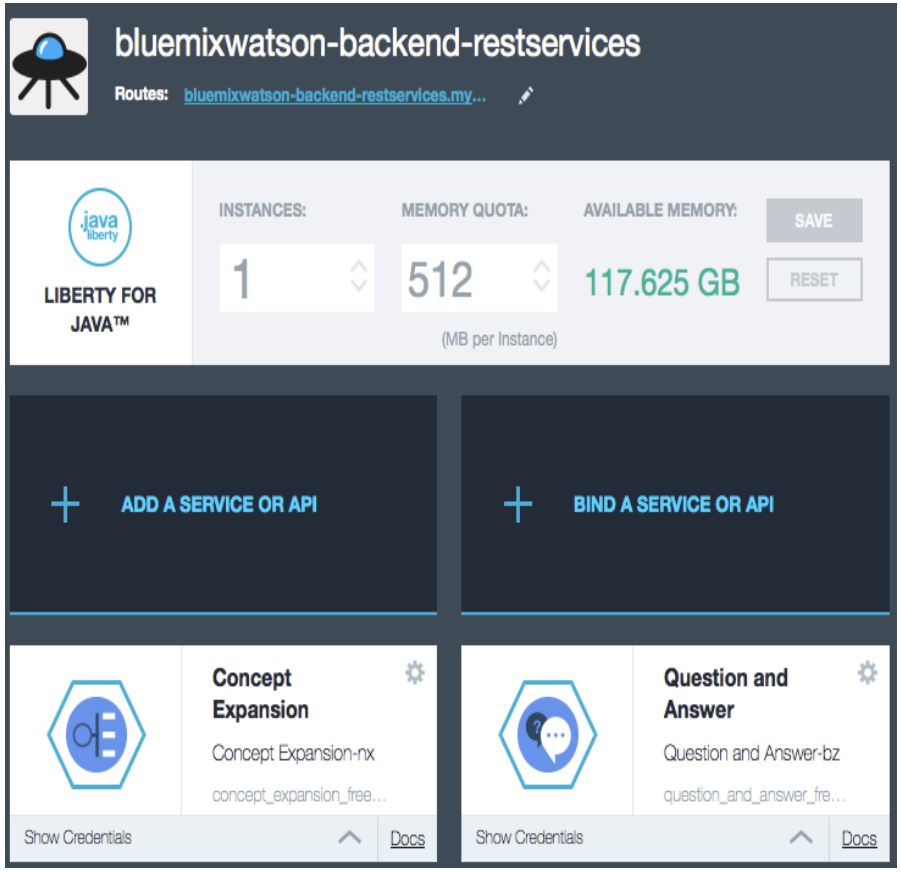
STEP 4 – APPENDIX A - BLUEMIX BACKEND APP TO PUBLISH RESTFUL API FOR WATSON

STEP 4:

Deploy the application to



STEP 4 – BLUEMIX BACKEND APP TO PUBLISH RESTFUL API FOR WATSON



APPENDIX B – CODE WALKTHROUGH - BLUEMIX BACKEND APP TO PUBLISH RESTFUL API FOR WATSON

Bluemix Watson Backend Application – How to 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
  - Concept Expansion
  - Question and Answer

Bluemix Watson Backend 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 mysqldbkey, BasicDBObject vcapobj) {

    BasicDBList mysqldblist = (BasicDBList) vcapobj.get(mysqldbkey);
    vcapobj = (BasicDBObject) mysqldblist.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 Watson Backend 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 passed as an output to the /searchtips rest call.
  - The results are displayed on the Vaadin UI Screen

APPENDIX C – CODE WALKTHROUGH – VAADIN UI – FRONT END APPLICATION

VAADIN UI – Build UI

- VaadinWatsonUI Code

```
34 import com.vaadin.ui.VerticalLayout;
35 import com.vaadin.ui.themes.ValoTheme;
36
37
38 @PreserveOnRefresh
39 @Title("VaadinUI")
40 @Theme("vaadinui")
41
42 public class VaadinWatsonUI extends UI {
43
44     TextField postQuestionText = new TextField();
45     Button postQ = new Button("Ask Watson about a Health Symptom?");
46     String basepath = VaadinService.getCurrent().getBaseDirectory().getAbsolutePath();
47     // Show the image in the application
48     FileResource resource = new FileResource(new File(basepath + "/WEB-INF/images/iconNoCommunityPhoto155.png"));
49     Image image = new Image(null, resource);
50     WatsonService ws = new WatsonService();
51     WatsonResponse wresp = new WatsonResponse();
52     VerticalLayout vlayout = null;
53     Table table1;
54     Table table2;
55
56
57     @Override
58     protected void init(VaadinRequest request) {
59         FormLayout formLayout = new FormLayout();
60         postQ.setStyleName(ValoTheme.BUTTON_PRIMARY);
61         postQ.setClickShortcut(ShortcutAction.KeyCode.ENTER);
62         HorizontalLayout hlayout = new HorizontalLayout(postQ);
63         HorizontalLayout head = new HorizontalLayout();
64         head.addComponent(new Label("WELCOME TO IBM VAADIN CHALLENGE"));
65         head.setWidth("500px");
66         HorizontalLayout head1 = new HorizontalLayout();
67         head1.addComponent(new Label("VAADIN BLUEMIX WATSON HEALTHCARE SERVICES SAMPLE PPLLICATION"));
68         head1.setWidth("700px");
69         hlayout.setSpacing(true);
70         postQuestionText.setImmediate(true);
71         postQuestionText.setInputPrompt("Example: Depression");
72         postQuestionText.setWidth("500px");
73         formLayout.addComponents(image, head, head1, postQuestionText, hlayout);
74         setVisible(true);
75
76         postQ.addClickListener(new Button.ClickListener() {
```

VaadinWatsonUI class → com.ecodcnc.vaadinui.VaadinWatsonUI

This class builds the Vaadin UserInterface uisng different layouts

VAADIN UI – Talk to Backend Services and Parse

- WatsonService Code →

```
public class WatsonService {

    private static final String targetURL = "http://bluemixwatson-backend-restservices.mybluemix.net/api/service/searcht
    String response;

    public String getResponseFromBluemixWatson(String forminput) {
            try {

            URL restServiceURL = new URL(targetURL);

            HttpURLConnection conn = (HttpURLConnection) restServiceURL.openConnection();
            conn.setDoOutput(true);
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Accept", "application/json");

            String input = "{\"keyword\"" + ":"  + forminput + "}";

            OutputStream os = conn.getOutputStream();
            os.write(input.getBytes());
            os.flush();
            String output;
            if (conn.getResponseCode() != 200) {
                throw new RuntimeException("HTTP GET Request Failed with Error code :" + conn.getResponseCode());
            }

            BufferedReader br = new BufferedReader(new InputStreamReader((conn.getInputStream())));

            System.out.println("Output from Server .... \n");
            while ((output = br.readLine()) != null) {
                setResponse(output.toString());
                System.out.println(output);
            }

            conn.disconnect();

        } catch (MalformedURLException e) {
```

VaadinWatsonUI class → com.ecodcnc.WatsonService

This class make http rest call to Integrate VaadinWatsonUI with the Backend services

VAADIN – BLUEMIX – WATSON HEALTHCARE SAMPLE APP

Questions?