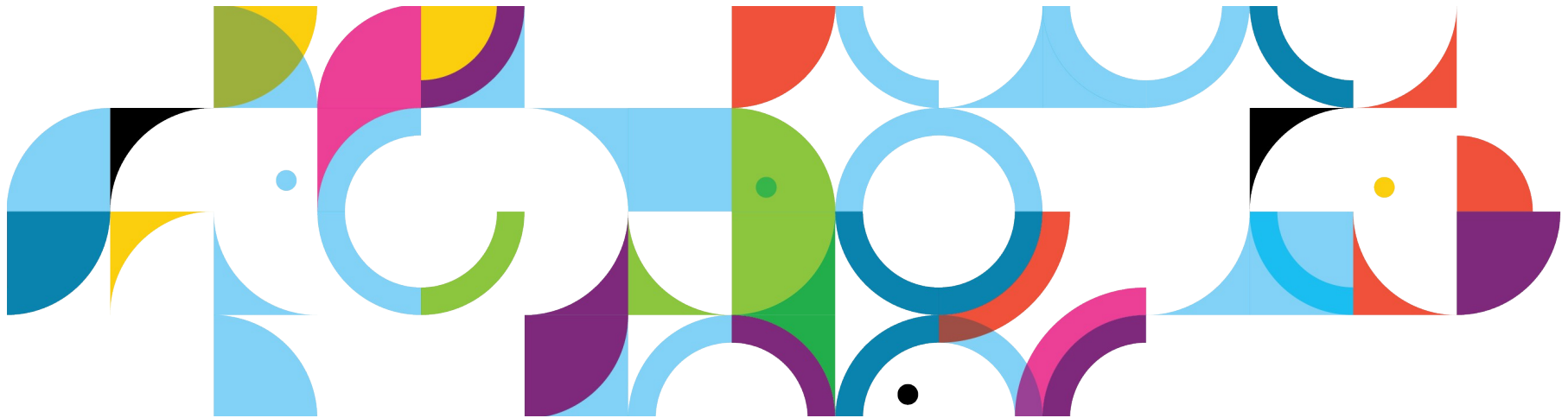


Social Business Toolkit SDK

IBM Social Business



Please note:

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Overview

- **Overview - What is the Social Business Toolkit?**
 - Composed by a set of RESTful Java and Javascript APIs and services
 - Allows easy integration with IBM Social Products and other services
 - APIs based on web patterns including JSON, ATOM, REST
 - Open standards ready
 - Hybrid deployment (on-premises and cloud)

From developers...

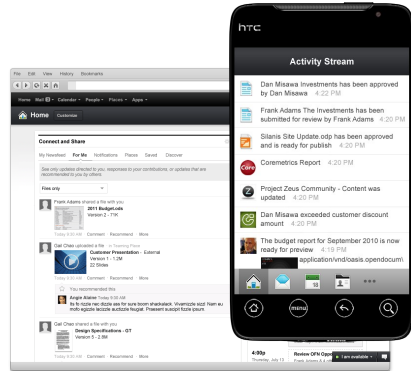


...to developers

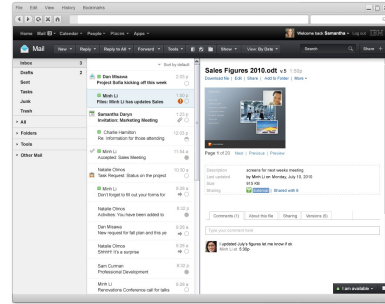
Open Standards



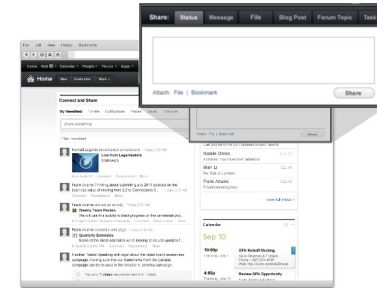
Unified user experiences



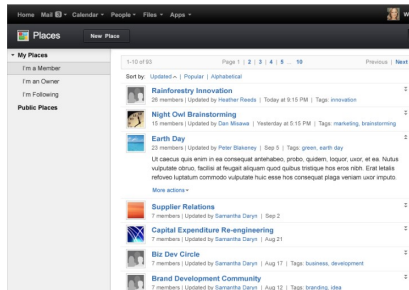
Activity Streams



Embedded Experiences



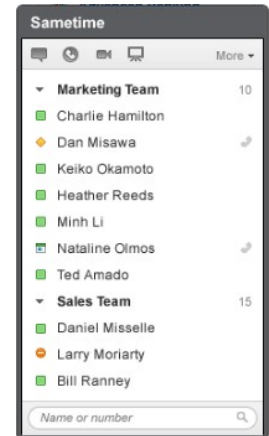
Share Box



- Collaboration**
- email
- Communities
- Activities
- File sharing
- Wikis, forums

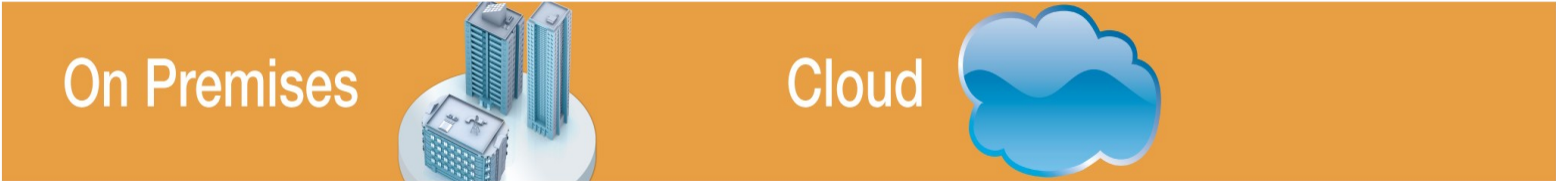
Unified user experiences and Consistent extensibility

independent of product, delivery model (cloud, on-premises), and client type (desktop client, browser, mobile)



Unified Communication

Hybrid Deployment



IBM Social Business Toolkit SDK - Now!

Pre-configured Tomcat

OpenSocial Support

Code Snippets

Security



Sample Applications



Server Assets
J2EE/Domino



Documentation



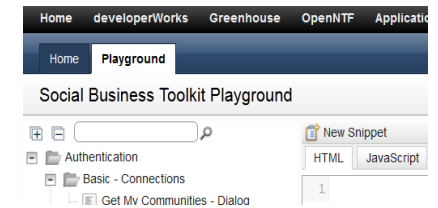
Language Bindings



JS
dojo



Developer Playground



Benefits

Instead of dealing with this:

```
<?xml version="1.0" encoding="UTF-8"?>
<atom:entry xmlns:app="http://www.w3.org/2007/app"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:cmis="http://docs.oasis-open.org/ns/cmismessaging/200908/"
  xmlns:cmism="http://docs.oasis-open.org/ns/cmismessaging/200908/"
  xmlns:cmisra="http://docs.oasis-open.org/ns/cmismessaging/200908/"
  xmlns:lcmis="http://www.ibm.com/xmlns/prod/sn/cmism" xmlns:snx="http://www.ibm.com/xmlns/prod/sn">
  <atom:author>
    <atom:name>Shankar Vakil</atom:name>
    <atom:email>shankar.vakil@notes.llrenovations.com</atom:email>
    <lcmis:email>shankar.vakil@notes.llrenovations.com</lcmis:email>
    <snx:userid>20527378</snx:userid>
    <snx:userState>active</snx:userState>
    <lcmis:displayName>Shankar Vakil</lcmis:displayName>
    <lcmis:principalId>20527378</lcmis:principalId>
    <snx:orgId>20051314</snx:orgId>
    <snx:orgName>Renovations Inc.</snx:orgName>
  </atom:author>
  <atom:link
    href="https://apps.lotuslive.com/files/app/file/a041af41-d2c0-4bec-a106-4e16a4f15195"
    rel="alternate" type="text/html"/>
  <atom:content
    src="https://apps.lotuslive.com/files/basic/cmismessaging/repository/p%2120527378/object/snx%3Afile%21a04e16a4f15195/smcl4645780839489271795tmp" type="application/octet-stream"/>
  <atom:link
    href="https://apps.lotuslive.com/files/basic/cmismessaging/repository/p%2120527378/object/snx%3Afile%21a04e16a4f15195/smcl4645780839489271795tmp"
    rel="edit-media" type="application/octet-stream"/>
  <atom:link
    href="https://apps.lotuslive.com/files/basic/cmismessaging/repository/p%2120527378/object/snx%3Afile%21a04e16a4f15195/smcl4645780839489271795tmp"
    hreflang="" length="96" rel="enclosure"/>
```

Benefits

You deal with this:

```
require([ "sbt/smartcloud/FileService", "sbt/dom" ], function(FileService, dom) {
  var fileService = new FileService();
  fileService.getFileEntry({
    loadIt : true,
    id : "%{sample.smartcloud.fileEntryId}",
    load : function(file) {
      var content = "";
      content = content + file.getId() + " : " + file.getName();
      dom.setText("content", content);
    },
    error : function(error) {
      dom.setText("content", "Error received. Error Code = " + error.code + ". Error Message = " + error.message);
    }
  });
});
```

Low Level APIs - JavaScript

- The low level APIs allow you to consume any REST service not directly supported by the high level APIs. This is an example in Javascript.

```
require(["sbt/connections/core","sbt/xml","sbt/xpath","sbt/dom"],function(conn,xml,xpath,dom) {  
  
    var ep = sbt.Endpoints['connections'];  
    ep.xhrGet({  
        serviceUrl: "/activities/service/atom2/activities",  
        handleAs: "text",  
        content: {  
            email: "FrankAdams@renovations.com",  
            ps: 5  
        },  
        load: function(response) {  
            var displayStr = "";  
            var activities = xpath.selectNodes(xml.parse(response), "/a:feed/a:entry", conn.namespaces);  
            for(var count = 0; count < activities.length; count++){  
                var activityName = xpath.selectText(activities[count], "a:title", conn.namespaces);  
                displayStr += activityName + ((count == activities.length -1) ?" ":" , ");  
            }  
            if (displayStr.length == 0) {  
                displayStr = "You are not a member of any activity.";  
            }  
            dom.setText("content",displayStr);  
        },  
        error: function(error){  
            dom.setText("content","Error received. Error Code = " + error.code + ". Error Message = " + error.message);  
        }  
    });  
});
```

Low Level APIs - Java

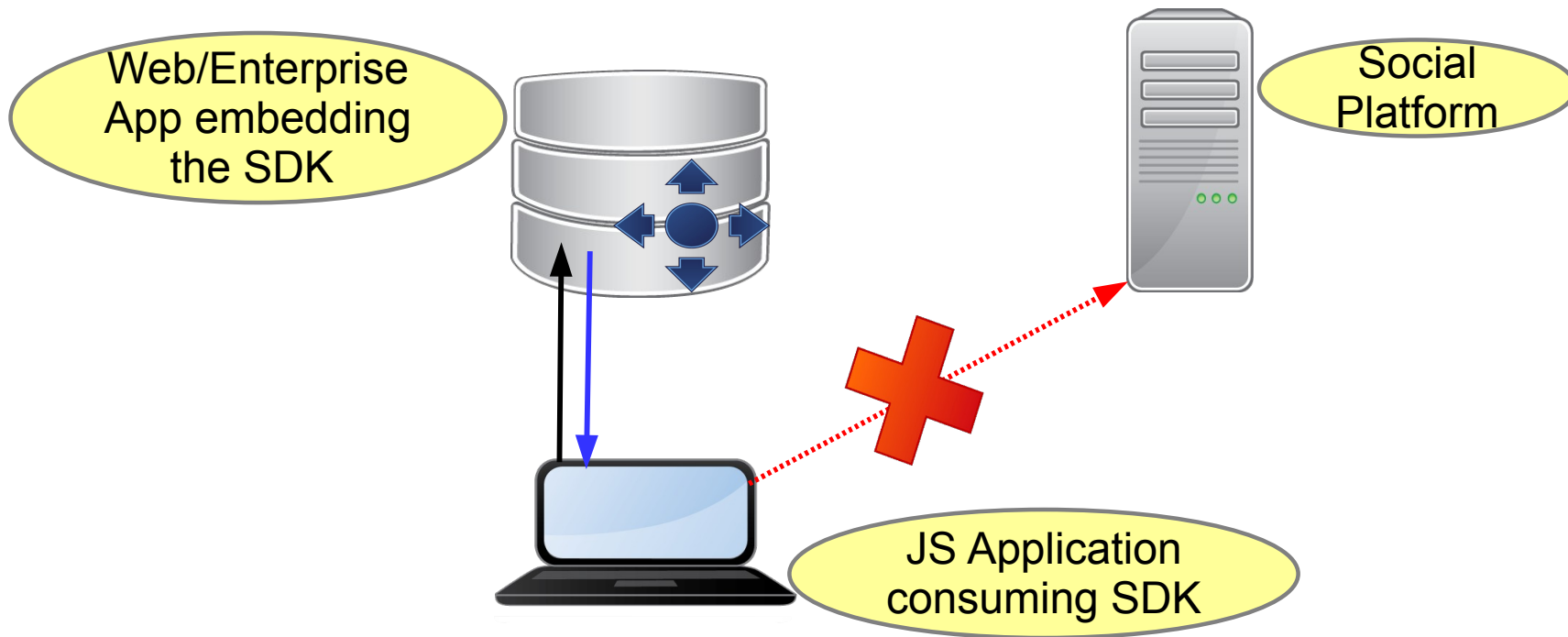
- The low level APIs allow you to consume any REST service not directly supported by the high level APIs. This is an example in Java.

```
<%  
    String email = "FrankAdams@renovations.com";  
  
    Endpoint ep = EndpointFactory.getEndpoint("connections");  
    System.out.println("ep authtype: " + ep.getAuthType());  
  
    String profileUrl = "profiles/atom/profile.do";  
    Map<String, String> parameters = new HashMap<String, String>();  
    parameters.put("email", email);  
  
    Object result = ep.xhrGet(profileUrl, parameters, ClientService.FORMAT_XML);  
%>  
  
<%= (new XmlNavigator((Node)result)).stringValue("feed/entry/contributor/name")%>
```

Server Assets

- **Proxy Server**
- **Library Servlet**
- **SBT Filter**
- **Endpoints**
- **Configuration**
- **Augment existing services**
- **Functional services (i.e. send email)**
- **Authentication/Authorization and Security**

Proxy Server



Proxy Server

In web.xml

```
<servlet>
  <description>The service servlet handles requests from the toolkit to access external resources.</description>
  <display-name>Social Business Toolkit Service Servlet</display-name>
  <servlet-name>ServiceServlet</servlet-name>
  <servlet-class>com.ibm.sbt.service.core.servlet.ServiceServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>ServiceServlet</servlet-name>
  <url-pattern>/service/*</url-pattern>
</servlet-mapping>
```

Library Servlet

- Serves the core JavaScript Modules of the SDK

In web.xml

```
<servlet>
  <description>This servlet initializes the specified JavaScript library for use by the Social Business Toolkit.</description>
  <display-name>Social Business Toolkit Library Servlet</display-name>
  <servlet-name>LibraryServlet</servlet-name>
  <servlet-class>com.ibm.sbt.jslibrary.servlet.LibraryServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>LibraryServlet</servlet-name>
  <url-pattern>/library/*</url-pattern>
</servlet-mapping>
```

- Include a script tag pointing to the library servlet in order to start consuming social SDK

```
<script type="text/javascript" src="/library?ver=1.8.0"></script>
```

Library Servlet

- Once LibraryServlet is included in script tag, JavaScript samples are all set to consume Social SDK capabilities

```
require([ "sbt/connections/core", "sbt/dom", "sbt/xml", "sbt/xpath", "sbt/Endpoint" ], function(conn, dom, xml, xpath, Endpoint) {
  var endpoint = Endpoint.find("connections");
  endpoint.xhrGet({
    serviceUrl : "/profiles/atom/profile.do",
    handleAs : "text",
    content : {
      userid : "%{sample.id1}"
    },
    load : function(response) {
      var doc = xml.parse(response);
      var name = xpath.selectText(doc, "/a:feed/a:entry/a:contributor/a:name", conn.namespaces);
      dom.setText("content", name);
    },
    error : function(error) {
      dom.setText("content", error);
    }
  });
});
```

```
require([ "sbt/connections/ProfileService", "sbt/dom" ], function(ProfileService, dom) {
  var id = "%{sample.id1}";
  var profileService = new ProfileService();
  profileService.getProfile({
    id: id,
    load: function(profile){
      if(profile.getDisplayName()){
        dom.setText("content",profile.getDisplayName());
      }else{
        dom.setText("content","No Result");
      }
    }
  },
```

Library Servlet

- **Registers the core JavaScript modules path:**
 - String Utilities
 - XML / XPath helpers
 - Json helpers
 - Proxy
 - Endpoint
 - JS library Bridge
 - Dojo / Dojo AMD/ JQuery libraries
 - Transport
 - Authenticators
 - SBT Configuration
 - Connections and SmartCloud APIs
- **Returns Endpoint definitions requested by JavaScript application**





















Library Servlet parameters

| <i>Parameter</i> | <i>Description</i> | <i>Default Value</i> |
|------------------|--|---------------------------------|
| toolkitUrl | Override location of the toolkit | %local_server%/sbt. |
| javaScriptPath | Override path to the JavaScript library | /js/sdk |
| serviceUrl | Override location of the service servlet | %local_application %/service |
| environment | Override name of the default environment | defaultEnvironment |

SBT Filter

```
<filter>
  <description>This filter is responsible for creating the toolkit application and context objects
  | for every servlet within this web application.</description>
  <display-name>Social Business Toolkit Filter</display-name>
  <filter-name>SBTFilter</filter-name>
  <filter-class>com.ibm.sbt.util.SBTFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>SBTFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Social SDK: Endpoints

-  AbstractEndpoint.java
-  AnonymousEndpoint.java
-  ApplicationEndpoint.java
-  BasicEndpoint.java
-  ConnectionsBasicEndpoint.java
-  ConnectionsOAuth2Endpoint.java
-  ConnectionsSSOEndpoint.java
-  DominoBasicEndpoint.java
-  DominoOAuth2Endpoint.java
-  DominoSSOEndpoint.java
-  Endpoint.java
-  EndpointFactory.java
-  GadgetEndpoint.java
-  GadgetOAuthEndpoint.java
-  OAuth2Endpoint.java
-  OAuthEndpoint.java
-  SametimeBasicEndpoint.java
-  SmartCloudOAuth2Endpoint.java
-  SmartCloudOAuthEndpoint.java
-  SSOEndpoint.java

Social SDK: Endpoint (managed bean)

```
<!-- Connections -->
<managed-bean>
  <managed-bean-name>connections</managed-bean-name>
  <managed-bean-class>com.ibm.sbt.services.endpoints.ConnectionsBasicEndpoint</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
  <managed-property>
    <property-name>url</property-name>
    <value>#{connections.url}</value>
  </managed-property>
  <managed-property>
    <property-name>passwordStore</property-name>
    <value>PwdStore</value>
  </managed-property>
  <managed-property>
    <property-name>authenticationPage</property-name>
    <value>/sbt/loginForm.html?endpoint=connections</value>
  </managed-property>
  <managed-property>
    <property-name>authenticationService</property-name>
    <value>communities/service/atom/communities/my</value>
  </managed-property>
  <!-- Trust the connection -->
  <managed-property>
    <property-name>forceTrustSSLCertificate</property-name>
    <value>>true</value>
  </managed-property>
</managed-bean>
```

Social SDK: Endpoint (managed bean) using OAuth

```
<!-- SmartCloud OAuth 1.0 -->
<managed-bean>
  <managed-bean-name>smartcloud</managed-bean-name>
  <managed-bean-class>com.ibm.sbt.services.endpoints.SmartCloudOAuthEndpoint</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
  <!-- Endpoint URL -->
  <managed-property>
    <property-name>url</property-name>
    <value>#{smartcloud.url}</value>
  </managed-property>
  <!-- Trust the connection -->
  <managed-property>
    <property-name>forceTrustSSLCertificate</property-name>
    <value>>true</value>
  </managed-property>
  <!-- Fixed values for the registered application -->
  <managed-property>
    <property-name>consumerKey</property-name>
    <value>#{smartcloud.consumerKey}</value>
  </managed-property>
  <managed-property>
    <property-name>consumerSecret</property-name>
    <value>#{smartcloud.consumerSecret}</value>
  </managed-property>
  <managed-property>
    <property-name>requestTokenURL</property-name>
    <value>#{smartcloud.requestTokenURL}</value>
  </managed-property>
  <managed-property>
    <property-name>authorizationURL</property-name>
    <value>#{smartcloud.authorizationURL}</value>
  </managed-property>
  <managed-property>
    <property-name>accessTokenURL</property-name>
    <value>#{smartcloud.accessTokenURL}</value>
  </managed-property>
</managed-bean>
```

Social SDK: Configuration

In context.xml

```
<!-- Configures the location of a sbt.properties file which  
| contains the configuration proeprties for use with the Social Business Toolkit -->  
<Resource name="url/ibmsbt-sbtproperties" auth="Container" type="java.net.URL"  
| factory="com.ibm.commons.runtime.naming.UrlFactory" url="sbt.properties" />
```

In sbt.properties

```
# Connections Basic Endpoint Parameters  
connections.url=https://qs.renovations.com:444  
connections.authenticationPage=/sbt/loginForm.html?endpoint=connections  
connections.authenticationService=communities/service/atom/communities/my  
  
# Connections OAuth 2.0 Endpoint Parameters  
connectionsOA2.url=https://qs.renovations.com:444  
connectionsOA2.serviceName=SBTK  
connectionsOA2.appId=SBTK  
connectionsOA2.consumerKey=SBTK
```

Social SDK: Environment (managed bean)


```
<!-- Default Environment -->
<managed-bean>
  <managed-bean-name>defaultEnvironment</managed-bean-name>
  <managed-bean-class>com.ibm.sbt.jslibrary.SBTEnvironment</managed-bean-class>
  <managed-bean-scope>application</managed-bean-scope>
  <managed-property>
    <property-name>endpoints</property-name>
    <value>connections,connections0A2,smartcloud,smartcloud0A2,domino,sametime</value>
  </managed-property>
  <managed-property>
    <property-name>properties</property-name>
    <value><![CDATA[
      sample.email1=
      sample.email2=
    ]]>
  </value>
</managed-property>
</managed-bean>

<!-- OpenSocial Environment -->
<managed-bean>
  <managed-bean-name>openSocial</managed-bean-name>
  <managed-bean-class>com.ibm.sbt.jslibrary.SBTEnvironment</managed-bean-class>
  <managed-bean-scope>application</managed-bean-scope>
  <managed-property>
    <property-name>endpoints</property-name>
    <value>connectionsOS:connections,connectionsOS0A:connections0A2</value>
  </managed-property>
</managed-bean>
```


Email service


```
document.getElementById('sendBtn').onclick = function(e) {
  var subject = document.getElementById('subjectInput').value;
  var emailJson = {
    to: to,
    cc: cc,
    bcc: bcc,
    subject: subject,
    mimeParts: mimeParts
  };
  email.send(emailJson, {
    callback: function(response) {
      var successElement = document.getElementById('success');
      successElement.setAttribute('style', 'display:none;');
      var errorElement = document.getElementById('error');
      errorElement.setAttribute('style', 'display:none;');
      if(response.message || (response.error && response.error.length != 0)) {
        if(response.message) {
          errorElement.innerHTML = response.message;
        } else {
          errorElement.innerHTML = response.error[0].message;
        }
        errorElement.setAttribute('style', '');
      } else {
        successElement.setAttribute('style', '');
      }
    }
  });
};
```



Services

- **Connections** 
 - ActivityStreams
 - Bookmarks
 - Communities
 - Files
 - Profiles (with caching support)

- ... and any REST API

- **Smartcloud** 
 - Communities
 - Files
 - Profiles (with caching support)

- More to come... 
 - Sametime

 - others...   



APIs

- **Java API for tight server-side integration.**



- **Javascript API (library-agnostic approach)**
 - Currently supported (more to come)



- **OpenSocial support**
 - Gadget Endpoint



- **Portlets, iWidgets, Web Parts...**



Security

- Endpoint configuration
- Oauth 1.0/2.0 support



Developing the Social Business SDK

- Open Source Project



- Community



OpenNTF release

- Preconfigured Tomcat



- Run samples in



Playground

- XPages based application
- Available on Greenhouse



The screenshot displays the Social Business Toolkit Playground interface. At the top, there are tabs for "Home" and "Playground". Below the tabs, the breadcrumb path is "Social Business Toolkit Playground > Connections/Communities / Get Community Title".

On the left side, there is a file tree structure:

- Authentication
- Connections
 - Activities
 - ActivityStreams
 - Bookmarks
 - Communities
 - Add Community Member
 - Create Community
 - Delete Community
 - Get Community Member Name
 - Get Community Members
 - Get Community Tags
 - Get Community Title

On the right side, there is a "New Snippet" editor with tabs for "HTML", "JavaScript", and "CSS". The "JavaScript" tab is active, showing the following code:

```
1 require(["sbt/connections/CommunityService"  
2     var communityService = new CommunitySer  
3     var communityId = "%${sample.communityId  
4     communityService.getCommunity({  
5         id: communityId,  
6         load: function(community) {  
7             if(community.getTitle()){  
8                 dom.setText("content", commu  
9             }else{  
10                dom.setText("content", "No R  
11            }  
12        },  
13        error: function(error) {  
14            dom.setText("content", "Error re  
15        }  
16    });  
17 });
```

Code snippets

- Self-documented samples
- Copy, paste and run samples in your own application
- No need to Read The Funky Manual before getting some code running.

```
require(["sbt/connections/ActivityStreamService", "sbt/connections/ActivityStreamConstants", "sbt/dom"], function(ActivityStreamService, ASConstants, dom) {
    dom.byId("loading").style.visibility = "visible";
    var activityStreamService = new ActivityStreamService();
    activityStreamService.getUpdatesFromCommunitiesIFollow({
        load : function(as) {
            var aEntries = as.getEntries();//get array of all activityEntry Objects
            if(aEntries.length == 0){
                dom.setText("content", "No Results");
            }
            for(var i=0; i<aEntries.length; i++){
                var thisEntry = aEntries[i];
                var liElement = document.createElement("li");
                liElement.setAttribute("id", "li"+i);
                document.getElementById("content").appendChild(liElement);
                dom.setText("li"+i, thisEntry.connections.plainTitle);
            }
            dom.byId("loading").style.visibility = "hidden";
        },
        error : function(error){
            dom.setText("content", "Error received. Error Code = " + error.code + ". Error Message = " + error.message);
            dom.byId("loading").style.visibility = "hidden";
        }
    });
});
```

Documentation

- **Java**
 - JavaDoc API documentation
- **Javascript**
 - YUIDoc API Documentation
- **Wikis**
 - <http://www-10.lotus.com/ldd/appdevwiki.nsf/xpDocViewer.xsp?lookupName=IBM+Social+Business+Toolkit+SDK+documentation#action=openDocument&content=catcontent&ct=prodDoc>
- **Coming soon:**
 - Playground API explorer

QSI – QuickStart Image

- **Preconfigured VM with preinstalled and preconfigured software and servers**
 - Connections, Domino, Sametime
 - All SDK configuration completed i.e. OAuth Keys, ...

Pre-integrated software appliance

IBM Social Business Toolkit SDK

IBM Social Business Toolkit Playground

IBM Social Business Toolkit APIs

IBM Social Business Core Modules

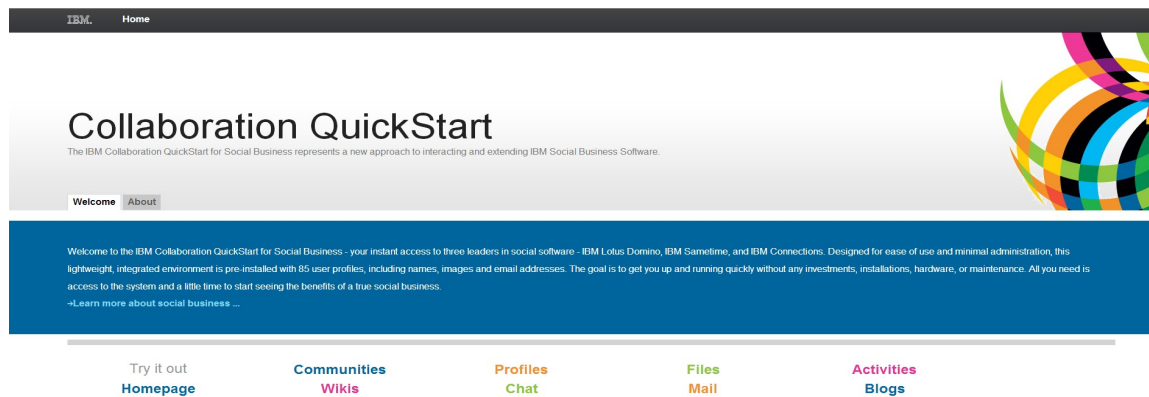
Workflow and forms
Lotus Domino
8.5.3 FP2

Social collaboration
Lotus Connections
C4 + CR2

Unified
communications
Lotus Sametime
8.5.2

QSI – QuickStart Image

- In process of being cleared by the legal department to be downloadable



IBM Collaboration QuickStart for Social Business

Revision: 20121025.01

IBM Lotus Domino Enterprise Server 8.5.3 FP2
IBM Connections 4.0 CR1
IBM Sametime 8.5.2 IFR1

Licensed Materials - Property of IBM. Copyright IBM Corporation and by other(s) 2012. All Rights Reserved.

IBM Social Business SDK - Next!

Quick Start Image



Mobile Support



Reusable Dijits



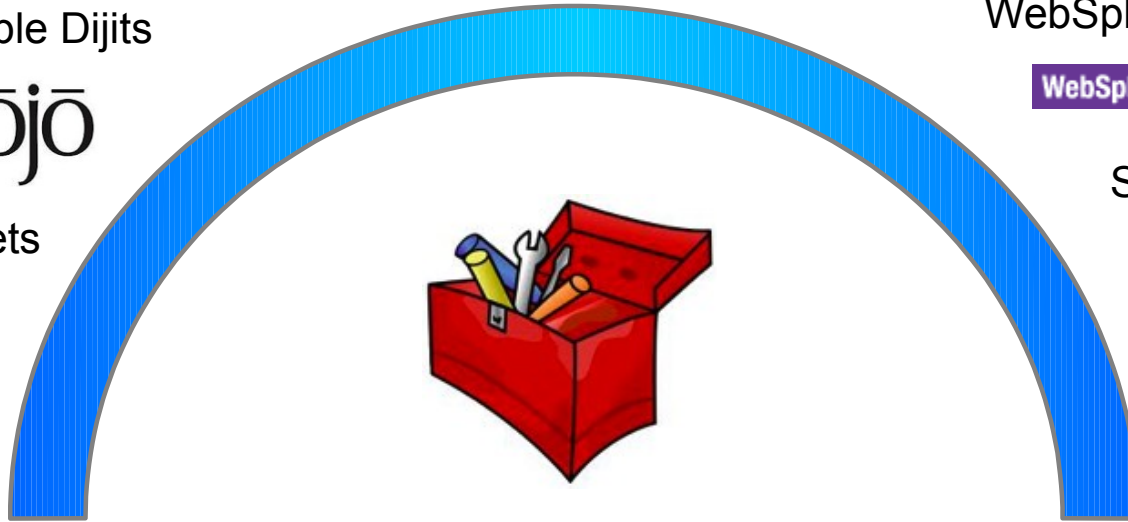
WebSphere Portal



Server Assets



SmarterWorkforce



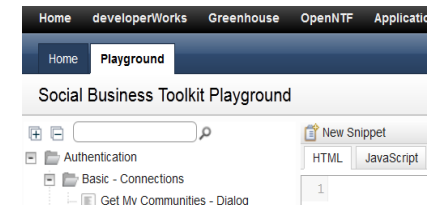
Patterns



Language Bindings



Developer Playground



IBM Social Business SDK - Next

- Language/library Bindings (with help from the Open Source community)



IBM Social Business SDK - Next

- **Reusable Dijits**

- Business Card
- Add to ActivityStream
- ...



- **Controls**

- GridRenderer
- ...



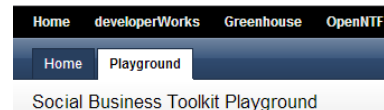
IBM Social Business SDK - Next

- **Mobile Support**



Getting Involved

- **Join the Community**
 - <http://bit.ly/ibmsbtcommunity>
- **Download the SDK**
 - <http://ibmsbt.openntf.org/>
- **Ask and answer questions on Stack Overflow #ibmsbt**
 - <http://stackoverflow.com/questions/tagged/ibmsbt>
- **Get the code from GitHub**
 - <https://github.com/OpenNTF/SocialSDK>
- **Follow us on Twitter #ibmsbt**
 - <https://twitter.com/ibmsbt>
- **Experiment with latest samples on Greenhouse**
 - <http://bit.ly/sbtplayground>



Getting Involved

- **Early Adopters**
 - Talk to us about what you need in the SDK
- **Open Source Developers**
 - Contribute new projects e.g. new language bindings



Legal disclaimer

© IBM Corporation 2013. All Rights Reserved.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.

All references to [insert fictitious company name] refer to a fictitious company and are used for illustration purposes only.

