Example
Tech Documentation

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Introduction

The Example HTML5 Web SDK is a collection of client-side JavaScript functionalities designed to allow web users to access and use the Example App Store. It is compatible with most modern mobile, tablet, and desktop browsers.

Namespace: example.

This is the `example.` namespace page and its model page. See them in the navigation bar or use the following links.

- Namespace
- Namespace Model: example.js

JavaScript Classes

JavaScript Classes *(https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes)* were introduced in ECMAScript 6. The `example.` namespace contains the following JavaScript Classes. View them under the **Classes** tab in the navigation bar or use the following links.

- `Class_1()`
- `Class_2()`
- `Class_3()`

JavaScript Models

JavaScript Models are the heart of a JavaScript application, containing interactive data and much of the logic surrounding it: conversions, validations, computed properties, and access control. The `example.` namespace and classes are linked to their respective JavaScript Models. View them under the **Models** tab in the navigation bar or use the following links.

- Model_1.js
- Model_2.js
- Model_3.js

Integration examples

Integration examples with editable sample code are provided for you. See them under the **HTML5 Examples** tab in the navigation bar, or use the following links.

- Example_1.html
- Example_2.html
- Example_3.html

General Concepts

Example App Store

Lorem ipsum.

Interpreting results

Lorem ipsum.

Rendering results

Lorem ipsum.

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Integrating the Example Web SDK

Contents

Overview

The Example Web SDK makes it super easy to integrate a simple app search UI into your page. Here is how you do it.

Installing dependencies

The Web SDK's JavaScript dependencies are managed with Node.js *(https://nodejs.org/download/)* and npm *(https://www.npmjs.com/)* which is bundled with recent versions of node. If you're running a recent version of Ubuntu, you should be able to `apt install nodejs`; on OS X, `brew install node`.

Old versions of Ubuntu (e.g. 12.04) may ship with old versions of Node; if that's you, you should consider directly downloading it *(https://nodejs.org/download/)* . At least Node.js 0.11+ and npm 2.1+ is recommended; older versions may mostly be OK but parts of the build process may not work as desired.

Once Node and npm are working, clone the repository, and from the repository root:

```javascript
$ npm install
```

You may find it useful to install the project's command-line Node tools globally:

```javascript
$ sudo npm install -g eslint jasmine jsdoc webpack webpack-dev-server
```

Descriptions

- jasmine *(https://jasmine.github.io/)* is our JavaScript testing framework.
- eslint *(http://eslint.org/)* is our JavaScript static analysis tool.
- webpack *(https://webpack.github.io/)* builds the project. It compiles, minifies, and bundles the JavaScript and assets.
- webpack-dev-server *(https://webpack.github.io/docs/webpack-dev-server.html)* is a convenient web server for development that integrates with webpack.

Linking to the SDK JavaScript

We recommend you refer to our hosted version of the SDK so that we can deliver updates for new kinds of deep views.

Include the following HTML `<script>` element in your HTML page, just before the closing `</body>` tag so it does not block page loading:

```html
<!DOCTYPE html>
    <html lang="en">
        <head>...</head>
        <body>
        ...
            <script src="https://www.example.com/js/sdk/stable/example.min.js.gz"></script>
        </body>
    </html>
```

Initializing the SDK

The Example Web SDK needs to be initialized before any searches are performed. You'll need to do the following:

1. **Find your Example partner credentials**. They need to be included in order to talk to the Example API.
2. Configure the SDK to recognize when it should perform a search. We provide two easy options for this:
    1. It can search when the page is loaded using an URL parameter, or
    2. It can search whenever a form is submitted using the value of a form input. For more complicated integration scenarios, see the other examples.
3. Create a HTML element to contain the Example SERP and configure the SDK to point at it.

Once you're all set, initialize the SDK like this:

```javascript
example.init({
    // fill in your partner credentials
    credentials: {
        partnerId: Enter your partnerID here,
        partnerSecret: 'Enter your partnerSecret here'
    },
    // the container into which search results will be rendered
    container: '#search-results',
    // a selector pointing to a search box, if you have one
    searchbox: 'input.searchbox',
    // the name of a URL parameter to use when running searches on page load
    queryParam: 'query'
})
```

Here's what a whole HTML working page should look like:

```html
<!DOCTYPE html>
    <html>
        <body>
            <form>
                <input class='searchbox' name='query' type="text" spellcheck="false"
                    placeholder="Type your query here">
                <input type="submit" value="Search">
            </form>

            <section id="search-results"></section>

            <script src="https://www.example.com/js/sdk/stable/example.min.js.gz"></script>
            <script>
                example.init({
                    credentials: {
                        partnerId: Enter your partnerID here,
                        partnerSecret: 'Enter your partnerSecret here'
                    },
                    container: "#search-results",
                    searchbox: "input.searchbox",
                    queryParam: "query"
                });
            </script>
        </body>
    </html>
```

Building and running

You can host the project locally using `webpac-dev-server`:

```javascript
$ npm run webpack-dev-server
```

You should see a bunch of output. Once it's running, visit http://localhost:xxxx to see the index page. If you have a modern browser, it should update automatically after any changes to the bundled JavaScript or assets.

Testing

You can run eslint and jasmine with the included configuration via:

```javascript
$ npm run lint
$ npm test
```

Tests and static analysis run automatically on every GitHub PR via Jenkins. Static analysis problems will also be displayed on the command line when running `webpack-dev-server`.

Rendering

The Example Web SDK `lookup()` and `search()` functions are your interface to talk to the Example API. Each one of them returns a promise *(https://www.promisejs.org/)* with either a single `SearchResult` or a list of SearchResults, which you can then do with as you please.

To render after you look up results, you can use the `xxxxxxCard()` and `xxxxxxSERP()` functions.

- **`xxxxxxCard()`** renders a single into the given container.
- **`xxxxxxSERP()`** renders a page full of cards, along with other functionality related to serving a search result page, like the UI when no results were present.

See the JavaScript examples in the following code snippet.

```javascript
// look up and render for a single known FURL
var furl = "func://allthecooks.com/showRecipeByRecipeId/292773";
example.lookup(furl).then(function(searchResult) {
    if (searchResult) {
        console.log("Got a result: ", searchResult);
        example.renderCard(searchResult);
    } else {
        console.log("No result for that FURL, try another.")
    }
}).catch(function(err) {
    console.error("Maybe this was a network error or something: ", err);
})

// print out the FURLs of the top 10 search results for a query
var query = "thai food";
example.search(query).then(function(searchResults) {
    searchResults.forEach(function(result) {
        console.log(result.deepState.furl);
    });
}).catch(function(err) {
    console.error("Maybe this was a network error or something: ", err);
})
```

Releasing and deploying

The Example Web SDK is deployed to S3, which is mirrored to CloudFront. The SDK winds up at www.example.com/--/-- under the following subdirectories:

- /latest - automatically deployed from develop.
- /stable - automatically deployed from master.
- /X.Y.Z - deployed from the corresponding release branch.

To deploy a named version, use deploy.sh:

```javascript
$ ./deploy.sh foo/bar/baz
```

Everything will get uploaded to https://www.example.com/js/sdk/foo/bar/baz.

To perform a new release, here's how it goes. Suppose the last release was 2.3.3, you're currently on 2.4.0-alpha and you want to cut the first candidate for 2.4.0, 2.4.0-rc0.

1. Create and check out a new release branch named release/2.4.0-rc0.
2. In that release branch, bump the version in package.json to 2.4.0-rc0.
3. `$ ./deploy.sh 2.4.0-rc0`.
4. Merge release/2.4.0-rc0 into develop. If further bug fixes happen on the release branch, also merge those into develop.

Later, perhaps after more release candidates, you'll be ready to release 2.4.0:

1. Repeat steps 1-4 above with the new version 2.4.0.
2. Merge release/2.4.0 into master. Jenkins will automatically deploy master into the /stable S3 directory.
3. Back in develop, bump the version in package.json to 2.5.0-alpha.
4. Party!

Contributing

We use git-flow *(http://nvie.com/posts/a-successful-git-branching-model/)* with the develop branch being trunk and the master being stable. Other conventions:

- Feature branches should just have whatever descriptive name you please.
- Release branches should have the prefix `release/`.
- The version in package.json should be kept up-to-date with the version of the latest release.

Issues are tracked in the JIRA WEBSDK project.

Integration examples

Open the Examples tab in the navigation bar at the top of any of these pages. Read the code in the:

- Example_1.html
- Example_2.html
- Example_3.html

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Example_1.html

HTML5, CSS and JavaScript

The following code block includes the HTML, CSS, and JavaScript for the Example_1.html page.

html

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=r
        <title>example.lookup</title>
        <style>
         html, body {
             font-family: 'Roboto', 'Arial', sans-serif;
             margin: 0;
             padding: 10px;
             height: 100%;
             -webkit-font-smoothing: antialiased;
             -moz-osx-font-smoothing: grayscale;
             background: #f5f6fa;
         }
         body>header>h2 {
             margin-top: 0;
         }
         .args { color: steelblue; }
        </style>
    </head>
    <body>
        <header>
            <h2>example.lookup(<i class='args'>furl, format, apiParams, delay, maxTries</i>) </h2>
            <p>
                Pass a 'furl' url param to this page and it will call example.lookup() to return a fully populat
                You can pass in the 'env' url param as 'stage', 'canary', or 'prod'.<br>
                You can pass in the 'format' url param as 'info', 'place', 'mrec1', etc...
            </p>
            <p>
                <a href="lookup.html?env=canary&furl=func://askme.com/cloudSearch-showProductById?q=toaster&form
            </p>
        </header>
        <section id="main-content"></section>
        <script src="../example.js"></script>
        <script>
            var urlParams;
            (window.onpopstate = function () {
             var match,
                pl     = /\+/g,
                search = /([^&=]+)=?([^&]*)/g,
                decode = function (s) { return decodeURIComponent(s.replace(pl, " ")); },
                query  = window.location.search.substring(1);
                urlParams = {};
                while (match = search.exec(query))
                    urlParams[decode(match[1])] = decode(match[2]);
                example.init({
                    env: urlParams["env"] || "stage",
                    credentials: {
                        partnerId: 2892812238,
                        partnerSecret: 'ypzpzt2gux3mscdwkksp3mdgxngxqr4h'
                    },
                    container: "#main-content"
                });
                if(urlParams["furl"]){
                    var furl = urlParams["furl"];
                    example.lookup(furl, urlParams['format'] || 'info', null, 1000, 3)
                    .then(function(searchResult){
                        // This searchResult should have all the data need to render a <Card/> component.
                        var output = '<h3>Populated SearchResult</h3>';
                        output += JSON.stringify(searchResult, null, 4);
                        document.getElementById('main-content').innerHTML = output;
                    }, function(err){
                        var output = '<h3>Failed to get content for furl: ' + err + '</h3>';
                        document.getElementById('main-content').innerHTML = output;
                    })
                }
            })();
        </script>
    </body>
</html>
```

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Class: `class_1()`

This Class provides all of the information about an "access method" that may be needed to link to a native or web deepstate.

```javascript
syntax                                                                                   javascript
  AccessMethod(accessUrl, packageName, action, category, component, editions, extra, method)
```

Parameter	Type	Description
`accessUrl`	String	A native or web link to access the deepstate.
`packageName`	String	Unique identifier for Android apps, for example: "com.yelp.android".
`action`	String	**Android specific**. Should default to 'android.intent.action.VIEW' on Chrome.
`category`	String	**Android specific**. Supported in Chrome Intent URIs.
`component`	String	**Android specific**. Supported in Chrome Intent URIs.
`editions`	Array	**Example defined**. Our Example-defined structure of an app 'edition'.
`extra`	Array	**Android specific**. Not currently supported in Chrome Intent URIs but could be later.
`method`	String	**Example defined**. Tells us how we should launch the deeplink. For example, launchIntentViaJSON means we should use Intent params but not AccessUrl if Chrome.

Source: models/model_1.js, line 23

Method: `(static) fromStateAccessInfoObject()`

Creates an `AccessMethod` from one of the objects in the `'accessMethods'` array returned by the Example API. The API response includes `'jar'` which is only relevant to the Example Android SDK.

```javascript
syntax                                                                                   javascript
  (static) fromStateAccessInfoObject(stateAccessInfoObj)
```

Parameter	Type	Description
`stateAccessInfoObj`	Object	The API object representing this AccessMethod.

Source: Model 1, line 39

Method: `selectAccessMethod()`

Specifies which access method we should use, or null if there are no access methods this client can use.

```
syntax                                                                 javascript
  (static) selectAccessMethod(client, accessMethods)
```

Parameter	**Type**	**Description**
`client`	Object	This is the `client` object created by `redirector.detectClient`.
`accessMethods`	Array	This is an array of `accessMethods` instances.
Source: Model 1, line 74		

Method: `isApplicable()`

Specifies whether this client (given the information we parsed out of their user agent) can apply a given access method.

```
syntax                                                                 javascript
  isApplicable(client)
```

Parameter	**Type**	**Description**
`client`	Object	This is the `client` object created by `redirector.detectClient`.
Source: Model 1, line 50		

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Model: model_1.js

This is the JavaScript Model for the `class_1()` Class.

javascript

```javascript
/**
 * All of the information about a example "access method" may be needed to link to a native or web deepstate.
 *
 * @param {string} accessUrl - Native or web link to access the deepstate.
 *
 * @param {string} packageName - Unique identifier for Android apps. ex. 'com.yelp.android'
 *
 * @param {string} action - Android specific. Should default to 'android.intent.action.VIEW' on Chrome
 *
 * @param {string} category - Android specific. Supported in Chrome Intent URIs.
 *
 * @param {string} component - Android specific. Supported in Chrome Intent URIs.
 *
 * @param {Array} editions - Our example-defined structure of an app 'edition'
 *
 * @param {Array} extra - Android specific. Not currently supported in Chrome Intent URIs but could be later.
 *
 * @param {string} method - example defined. Tells us how we should lauch the deeplink. ie. "launchIntentViaJSON
 *
 * @constructor
 * @memberof example
 */
function AccessMethod(accessUrl, packageName, action, category, component, editions, extra, method) {
    this.accessUrl = accessUrl;
    this.packageName = packageName;
    this.action = action;
    this.category = category;
    this.component = component;
    this.editions = editions;
    this.extra = extra;
    this.method = method;
}
/**
 * Creates an AccessMethod from one of the objects in the 'accessMethods' array returned by the example API.
 * The API response includes 'jar' is only relevant to example Android-sdk.
 *
 * @param {object} stateAccessInfoObj - The API object representing this AccessMethod.
 */
AccessMethod.fromStateAccessInfoObject = function(stateAccessInfoObj) {
    var packageName = (!stateAccessInfoObj.editions || !stateAccessInfoObj.editions.length)
        ? null
        : stateAccessInfoObj.editions[0].packageName;
    return new AccessMethod(stateAccessInfoObj.accessUrl, packageName, stateAccessInfoObj.action, stateAccessInf
};
/**
 * Whether this client (given the information we parsed out of their user agent) can apply a given access method
 *
 * @param {object} client - The client object created by redirector.detectClient
 */
AccessMethod.prototype.isApplicable = function(client) {
    if (!this.editions) {
        // if this access method doesn't specify editions, we can't use it
        return false;
    }
    var editionPlatforms = this.editions.map(function(edition) { return edition.platform; });
    if (client.os.name === "Android") {
        // if we're Android, use this access method only if it supports Android
        return editionPlatforms.indexOf("android") !== -1;
    }
    else if (client.os.name === "iOS"){
        return editionPlatforms.indexOf("ios") !== -1;
    }
    else {
        // if we're anything else, we can't use any access methods and we'll just open the fallback
        return false;
    }
};
/**
 * Which access method we should use, or null if there are no access methods this client can use.
 *
 * @param {object} client - The client object created by redirector.detectClient
 * @param {array}  accessMethods - Array of accessMethod instances.
 */
AccessMethod.selectAccessMethod = function(client, accessMethods) {
    accessMethods = Array.isArray(accessMethods) ? accessMethods : [accessMethods];
    var applicableMethods = accessMethods ? accessMethods.filter(function(am){ return am.isApplicable(client); }
    return applicableMethods.length ? applicableMethods[0] : {}; // arbitrarily pick the first applicable one, e
};
module.exports = AccessMethod;
```

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Namespace: `example.`

The `example.` namespace contains all of the Web SDK's functionality, and includes the following JavaScript Classes *(https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes)* :

`class_1()`, `class_2()`, `class_3()`

Namespace methods

```
(static) method_1()
(static) method_2()
(static) method_3()
```

Methods

`(static) method_1`

This method initializes the Web SDK with information that will be applicable to every subsequent search.

Parameter	Type	Description
params	Object	An object containing initialization parameters as described in the following list.
Source: namespace.js, line 341		

Generic initialization param options:

locale
The name of the locale intended for the Web SDK. Use either "en_US" (US) or "zh_CN" (China). Defaults to "en_US".

env
The name of an environment hosting the API and related resources like bindings. One of "prod", "canary", or "stage." Defaults to prod.

credentials
An object **`{ partnerId: 'your-id', partnerSecret: 'your-secret' }`** containing your Example credentials. These will be used when communicating with the Example API.

internalTrackers
An `{id: config}` mapping containing configuration for built-in analytics trackers. See the individual tracker documentation for more details.

Param options that control default integration behavior:

layout
Specifies the default layout for rendered search cards; use either "vertical-list" or "carousel".

searchbox
A DOM element or selector pointing to a form element that can contain a query. When the element's parent form is submitted, the value of the form element will be used as the search query sent to the Example API.

queryParam

The name of a URL parameter that may be passed to the hosting page. When initSDK is called, if this parameter is present, a search will run immediately with the value of this parameter as the query.

container
A DOM element or selector specifying the default container for search results. This container will be used when rendering results after e.g. run run is called.

searchCallback
A callback that will fire whenever a search has successfully been automatically performed (either on load or on form submit.)

`(static) method_2(furl, format, apiParams, delay, maxTries)`

Looks up content by FURL (Functional URL) from the Example API and returns a promised SearchResult containing that content (or null if the FURL was not found).
Note: Due to backend limitations this API currently only works with static states and app search states.

Parameter	Type	Description
furl	String	**Required**. A single furl string for static or dynamic dvcs.
format	String	One of our DVC formats. 'info', 'place', etc... Defaults to 'info'.
apiParams	Object	Optional. Overrides the default api params when hitting /dvcs.
delay	Integer	Delay in ms between queries to v4/dvcs. Defaults to 1000.
maxTries	Integer	The max number of attempts to query v4/dvcs. Defaults to 3.
Source: namespace.js, line 221		

`(static) method_3(furl, format, apiParams)`

A convenience function that looks up furls and then renders them in the container provided in `Example.init()`.

Parameter	Type	Description
furl	String	**Required**. A single furl string for static or dynamic dvcs.
format	String	One of our DVC formats. 'info', 'place', etc... Defaults to 'info'.
apiParams	Object	Optional. Overrides the default api params when hitting /dvcs.
Source: namespace.js, line 240		

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Model: `example.` namespace

This is the JavaScript Model for the `example.` namespace.

javascript

```javascript
// Shim to prevent Safari private browsing mode from breaking the entire page.
require('./vendor/localstorage_safari_private_shim.js');
var _ = require('lodash');
var RSVP = require('rsvp');
// Configuration
var configs = require('./configs.js');
var metrics = require('./metrics.js');
// SDK
var Controller = require('./Controller.js');
var User = require('./analytics/User.js');
// utils
var utils = require('./utils.js');
var Constants = require('./constants.js');
var http = require('./http.js');
// Analytics
var Analytics = require('./analytics/Analytics.js');
var CollectTracker = require('./analytics/CollectTracker.js');
var GaTracker = require('./analytics/GaTracker.js');
// set during init
var analytics, controller, defaultApiParams, globalBindingFile;
// set during render
var renderingDependencies;
// Default SDK params
var sdkParams = {
    locale: 'en_US', // or 'zh_CN'
    layout: 'vertical-list',  // or 'carousel' for swipe
    numDVs: 2,
    queryParam: null,
    searchbox: '#example-searchbox',
    container: '#example-content',
    searchCallback: null,
    clickHandling: true,
    bundle: 'default'
};
/**
 * Passes a blob of data to all of the registered analytics trackers.
 *
 * @param {Object} data - The data to send to the trackers. Data format is tracker-specific and undocumented so
 * luck with that.
 *
 * @memberof example
 */
var track = function(data) {
    return controller.track(data);
};
// 'Promisifies' require.ensure for our rendering dependencies (useful so that
// we can use promises all the way down for e.g. error handling.) The provided
// callback will get an object containing the dependencies.
var ensureRenderingDependencies = function() {
    return new RSVP.Promise(function(resolve) {
        if(__CODE_SPLITTING__){
            require.ensure([
                "react",
                "react-dom",
                './rendering/CardsContainer.jsx'
            ], function(require) {
                renderingDependencies = {
                    React: require('react'),
                    ReactDOM: require('react-dom'),
                    CardsContainer: require('./rendering/CardsContainer.jsx')
                };
                resolve(renderingDependencies);
            });
        } else {
            renderingDependencies = {
                React: require('react'),
                ReactDOM: require('react-dom'),
                CardsContainer: require('./rendering/CardsContainer.jsx')
            };
            resolve(renderingDependencies);
        }
    });
};
// Renders an HTML message into the given container.
var displayMessage = function(html, container) {
    container = utils.selectElement(container || sdkParams.container);
    if (!container)
        utils.error("Could not display message " + html + " (no container element specified.)");
    if (renderingDependencies)
        renderingDependencies.ReactDOM.unmountComponentAtNode(container);
    container.innerHTML = '' + html + '';
};
/**
 * Returns a Promise which resolves to an array of final deepview populated SearchResult instances.
 * Populated deepviews can be found in SearchResult.staticStates[i].deepState.dv.content.
 *
 * @param {example.SearchResult[]} results - Array of SearchResult instances
 * @param {string} bundle - Determines which format to use for each schema. One of 'default' or 'rightHalfImage'
 * @param {string} format - Determines which 'format' to render. Ex. 'mrec1', 'mrec2', 'info', 'wideImageWithGri
 *
 * @memberof example
 */
var populateResults = function(results, bundle, format){
    return globalBindingFile.then(function(gbf) {
        return controller.populate(results, bundle || 'default', format, gbf);
    });
};
// Returns a Promise which resolves to a final populated SearchResult
//
// {string} searchResult - SearchResult instance with deepviews.
// {string} format - Determines which 'format' to render. Ex. 'mrec1', 'mrec2', 'info', 'wideImageWithGrid'. See
var populateOneResult = function(searchResult, format){
    return globalBindingFile.then(function(gbf) {
        return controller.populateSearchResult(searchResult, format, gbf);
    });
};
/**
 * Renders a React template with a page of search results into the provided container DOM element, one  per sear
 * Returns a promise containing the rendered React container component.
 *
 * @param {example.SearchResult[]} populatedResults - The final array of Search Results with populated DeepViewD
 * @param {Object} options - Define specific option variables: container (DOMElement) and clickHandling (bool)
 * @param {DOMElement} options.container - The DOM element (or DOM element selector) to render the  into.
 * @param {boolean} options.clickHandling - Boolean flag to enable/disable handling of click events on rendered
 * @param {string} options.bundle - Determines which `format` to use per each `schema` returned in SERP experien
 *
 * @param {string} options.format - Determines which 'format' to render. Ex. 'mrec1', 'mrec2', 'info', 'wideImag
 * @param {string} options.layout - Controls how results are laid out in the container. One of either 'vertical-
 * 'carousel'.
 *
 * @memberof example
 */
var renderSERP = function(resultData, options) {
    options = options || {};
    return ensureRenderingDependencies().then(function(deps) {
        if(_.isUndefined(controller)) {
            utils.error("Controller was not initialized. Did you call example.init first?");
            return false;
        }
        metrics.recordStart("Rendering SERP");
        var container = utils.selectElement(_.has(options, 'container') ? options.container : sdkParams.containe
        var layout = _.has(options, 'layout') ? options.layout : sdkParams.layout;
        var clickHandling = _.has(options, 'clickHandling') ? options.clickHandling : sdkParams.clickHandling;
        if(!resultData.populatedResults || resultData.populatedResults.length === 0){
            displayMessage("No results found.", container);
        } else {
            var rootComponent = deps.ReactDOM.render(deps.React.createElement(deps.CardsContainer, {
                numDVs: sdkParams.numDVs,
                analytics: analytics,
                layout: layout,
                isInteractive: clickHandling
            }), container);
            rootComponent.resetSearch(resultData.populatedResults);
        }
        metrics.recordFinish("Rendering SERP");
        return {
            rootComponent: rootComponent,
            dynamicResults: resultData.dynamicResults
        };
    });
};
/**
 * Shortcut for rendering a single ; identical to calling {@link example.renderSERP|renderSERP} with the "single
 * Returns a promise containing the rendered React container component.
 *
 * @param {example.SearchResult} result - The search results to render into the container.
 * @param {Object} options - Define specific option variables: container (DOMElement) and clickHandling (bool)
 * @param {DOMElement} options.container - The DOM element (or DOM element selector) to render the card into.
 * @param {boolean} options.clickHandling - Boolean flag to enable/disable handling of click events on rendered
 * @param {string} options.bundle - Determines which `format` to use per each `schema` returned in SERP experien
 * See 'formatBundle' block in {@link https://github.com/example/trelleborg/blob/new_format/example/sdk/global/g
 * @param {string} options.format - Determines which 'format' to render. Ex. 'mrec1', 'mrec2', 'info', 'wideImag
 * @param {string} options.layout - Controls how results are laid out in the container. One of either 'vertical-
 * 'carousel'.
 *
 * @memberof example
 */
var renderCard = function(result, options) {
    // 'format' should only be specified when rendering 1 card (ie. for ads). If none provided use the one set i
    options = options || {};
    options.format = options.format || undefined;
    options.layout = "single-card";
    return renderSERP({ populatedResults: [result] }, options);
};
// Attaches an event handler that will perform a search and render results whenever the form of the
// given search box is submitted.
var attachOnSubmit = function(searchbox) {
    var searchboxElement = utils.selectElement(searchbox);
    if (searchboxElement && searchboxElement.form){
        searchboxElement.form.addEventListener("submit", function(e) {
            e.preventDefault();
            example.run(searchboxElement.value).then(sdkParams.searchCallback || _.noop);
        });
    }
};
/**
 * Returns a Promise that resolves to a populated SearchResult (ready for rendering)
 * or rejects and returns the furl passed in.
 *
 * @param {string}  furl - Required. A single furl string for static or dynamic s.
 * @param {string}  format - One of our  formats. 'info', 'place', etc... Defaults to 'info'.
 * @param {Object}  apiParams - Optional. Overrides the default api params when hitting /s.
 * @param {integer} delay - Delay in ms between queries to v4/s. Defaults to 1000.
 * @param {integer} maxTries - The max number of attempts to query v4/s. Defaults to 3.
 *
 * @memberof example
 */
var lookup = function(furl, format, apiParams, delay, maxTries){
    if(typeof furl !== 'string' || !/^func:/.test(furl)){
        utils.error('furl arg must be a valid furl string');
    }
    return new RSVP.Promise(function(resolve, reject) {
        // When using _.defaults, leftmost arguments override all others.
        return controller.fetchSearchResult(furl, _.defaults(apiParams || {}, defaultApiParams), delay, maxTries
            .then(function(searchResult){
                return searchResult
                    ? populateOneResult(searchResult, format || 'info').then(resolve)
                    : reject(furl);
            });
    });
};
/**
 * Convenience function that looks up furls and then renders them in the container provided in example.init.
 *
 * @param {string}  furl - Required. A single furl string for static or dynamic s.
 * @param {string}  format - One of our  formats. 'info', 'place', etc... Defaults to 'info'.
 * @param {Object}  apiParams - Optional. Overrides the default api params when hitting /s.
 *
 * @memberof example
 */
var lookupAndRender = function(furl, format, apiParams) {
    if(typeof furl !== 'string' || !/^func:/.test(furl)){
        utils.error('furl arg must be a valid furl string');
    }
    if(!utils.selectElement(sdkParams.container)){
        utils.error('Missing placeholder container ' + sdkParams.container);
        return;
    }
    return lookup(furl, format, _.defaults(apiParams || {}, defaultApiParams)).then(renderCard);
};
/**
 * Performs a search using the example API and returns a promised ordered list of {@link
 * example.SearchResult|SearchResults} matching the query.
 *
 * @param {string} q - The query text to search for.
 * @param {Object} apiParams - Other parameter overrides to send to the example API.
 *
 * @memberof example
 */
var search = function(q, apiParams) {
    return controller.search(_.defaults({q: q}, apiParams || {}, defaultApiParams));
};
// Handles the lookup of dynamic furls by hitting v4/s for each furl,
// then replaces loading card with full content, or removes the card from SERP.
// #lookup will hit v4/s once every 1000 ms up to 3 times. This can be configured in configs.js.
//
// {Object} postRenderData - All the info needed to render dynamic s.
// {example.SearchResult[]} postRenderData.dynamicResults - Search Results missing content (ie. dynamic s)
// {ReactComponent} postRenderData.rootComponent - The root React Component (usually CardsContainer.jsx) which m
var handleDynamics = function(postRenderData){
    var rootComponent = postRenderData.rootComponent;
    if(postRenderData.dynamicResults && postRenderData.dynamicResults.length){
        postRenderData.dynamicResults.forEach(function(pr){
            lookup(pr.furl, pr.format, null, Constants.S_DELAY_MS, Constants.S_MAX_TRIES)
            .then(rootComponent.updateResult, rootComponent.removeResult);
        });
    }
};
/**
```

```
 * Convenience method for performing a search and rendering. Uses the preconfigured SDK settings to perform a se
 * and render the results into the default container. On any error, displays a message in the default container.
 *
 * @param {string} q - The query text to search for.
 *
 * @memberof example
 */
var run = function(q) {
    var params = _.defaults({q: q}, defaultApiParams);
    return controller.search(params)
        .then(populateResults)
        .then(renderSERP)
        .then(handleDynamics)
        .catch(function(err) {
            if (err === 'Network timeout') {
                displayMessage("No network connection available.", sdkParams.container);
            } else {
                utils.error(err);
            }
            throw err;
        });
};
// Returns a list of Tracker objects which should record analytics tracking events
// as per the settings object passed in by the client.
var initTrackers = function(trackerSettings, credentials) {
    var trackers = [new CollectTracker(credentials)];
    if(trackerSettings){
        if(trackerSettings.ga){ // Google Analytics
            trackers.push(new GaTracker(trackerSettings.ga[0], trackerSettings.ga[1]) );
        }
    }
    return trackers;
};
// Reads any existing user tracking information and returns an Analytics object
// ready to record tracking events for the current user.
var initAnalytics = function(trackers, environment, existingUser) {
    var currentUser = existingUser || new User(utils.guid());
    var analyticsInterface = new Analytics(trackers, environment, currentUser, defaultApiParams);
    if (!existingUser) { // save the new user and send a First Time User tracking event recording their existenc
        currentUser.save();
        analyticsInterface.record('firsttimeuser');
    }
    return analyticsInterface;
};
/**
 * Initializes the SDK with some information which will be applicable to every subsequent search.
 * Generic options:
 *
 * - `locale`: The name of the locale web-sdk is intended for.  One of "en_US" (US) or "zh_CN" (China).
 *   Defaults to "en_US".
 *
 * - `env`: The name of an environment hosting the example API and related resources like bindings.  One of "prc
 *   "canary", or "stage."  Defaults to prod.
 *
 * - `credentials`: An object `{ partnerId: 'your-id', partnerSecret: 'your-secret' }` containing your example
 *   credentials. These will be used when communicating with the example API.
 *
 * - `internalTrackers`: An {id: config} mapping containing configuration for built-in analytics trackers. See t
 *   individual tracker documentation for more details.
 *
 * Options that control default integration behavior:
 *
 * - `layout`: Specifies the default layout for rendered search cards. Either "vertical-list" or "carousel".
 *
 * - `searchbox`: A DOM element or selector pointing to a form element that can contain a query. When the elemen
 *   parent form is submitted, the value of the form element will be used as the search query sent to the exampl
 *
 * - `queryParam`: The name of a URL parameter that may be passed to the hosting page. When init is called, if
 *   this parameter is present, a search will run immediately with the value of this parameter as the query.
 *
 * - `container`: A DOM element or selector specifying the default container for search results. This container
 *   be used when rendering results after e.g. {@link example.run|run} run is called.
 *
 * - `searchCallback`: A callback that will fire whenever a search has successfully been performed automatically
 *   on load or on form submit.)
 *
 * @param {Object} params - An object containing initialization parameters as described above.
 *
 * @function init
 * @memberof example
 */
var init = function(initParams) {
    ensureRenderingDependencies(); // this eager-loads the templates chunk
    if (!initParams.credentials || !initParams.credentials.partnerId) {
        utils.print('No partner credentials provided.');
        return;
    }
    if(initParams.locale){
        if(_.has(configs.environments, initParams.locale)){
            sdkParams.locale = initParams.locale;
        } else {
            utils.print("Invalid locale in .init \'" + initParams.locale + "\'. Defaulting to \'" + sdkParams.lc
        }
    }
    var environment = configs.environments[sdkParams.locale][initParams.env || "prod"];
    if (!environment) {
        utils.print("Invalid environment: " + initParams.env);
        return;
    }
    // Default api params
    defaultApiParams = {
        geoloc: utils.storeNewGeoloc(),
        user_agent: navigator.userAgent,
        platform_ids: utils.getPlatformIds(),
        limit: 10,
        dv_limit: 3,
        custom_id: 75675980,
        auto_filter: 1,
        filter_empty_results: 0,
        fix_spelling: 0,
        include_descriptions: 0,
        include_screenshots: 0,
        include_editions: 1,
        include_authors: 1,
        include_snippets: 0,
        display_config: { platform: 'all' },
        version: process.build ? process.build.version : null
    };
    var trackers = initTrackers(initParams.internalTrackers, initParams.credentials);
    analytics = initAnalytics(trackers, environment, User.load());
    controller = new Controller(analytics, environment, initParams.credentials, sdkParams.locale);
    metrics.recordStart("Fetching global binding file");
    globalBindingFile = http.get(environment.cdnHost + "/" + Constants.GLOBAL_BINDING_FILE_PATH).then(function(g
        metrics.recordFinish("Fetching global binding file");
        return gbf;
    });
    sdkParams.bundle = initParams.bundle || sdkParams.bundle;
    sdkParams.container = initParams.container || sdkParams.container;
    sdkParams.searchbox = initParams.searchbox || sdkParams.searchbox;
    sdkParams.queryParam = initParams.queryParam || sdkParams.queryParam;
    sdkParams.searchCallback = initParams.searchCallback || sdkParams.searchCallback;
    sdkParams.layout = initParams.layout || sdkParams.layout;
    sdkParams.clickHandling = _.has(initParams, 'clickHandling') ? initParams.clickHandling : sdkParams.clickHar
    if (sdkParams.queryParam) {
        var query = utils.parseQS(location.search)[sdkParams.queryParam];
        if (query) {
            example.run(query).then(sdkParams.searchCallback || _.noop);
        } else {
            utils.print('"' + sdkParams.queryParam + '"' + ' was not found in the URL query string.');
        }
    }
    if (sdkParams.searchbox){
        attachOnSubmit(sdkParams.searchbox);
    }
};
/**
 * The namespace containing all SDK functionality. To use, call {@link example.init|init} followed by e.g. {@lir
 * @namespace example
 */
module.exports = {
    init: init,
    run: run,
    track: track,
    search: search,
    lookup: lookup,
    lookupAndRender: lookupAndRender,
    renderCard: renderCard,
    renderSERP: renderSERP,
    populateResults: populateResults,
    DeepView: require('./models/DeepView.js'),
    DeepViewData: require('./models/DeepViewData.js'),
    SearchResult: require('./models/SearchResult.js'),
    EditionMetadata: require('./models/EditionMetadata.js'),
    metrics: metrics,
    build: process.build || {}
};
```

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Product FAQ

Overview

The following frequently asked questions cover Example's products available to our partner community.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer finibus tincidunt gravida. Morbi nibh est.?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

2. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer finibus tincidunt gravida. Morbi nibh est.?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

3. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer finibus tincidunt gravida. Morbi nibh est.?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

4. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer finibus tincidunt gravida. Morbi nibh est.?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

5. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer finibus tincidunt gravida. Morbi nibh est.?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

6. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer finibus tincidunt gravida. Morbi nibh est.?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

7. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer finibus tincidunt gravida. Morbi nibh est.?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

8. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer finibus tincidunt gravida. Morbi nibh est.?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

9. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer finibus tincidunt gravida. Morbi nibh est.?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eu elementum ante, quis pellentesque erat. Nulla consequat magna a erat bibendum, in pellentesque ipsum malesuada. Suspendisse egestas non metus quis tempus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi volutpat ante vitae elit sollicitudin suscipit. Nulla venenatis efficitur pulvinar. Nulla vel est non neque finibus ullamcorper. Quisque tristique neque ornare neque dapibus interdum.

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Release Notes

Contents

New in this Web SDK version

- ------
- ------
- ------
- ------
- ------

Known issues

- -----
- -----
- -----
- -----
- -----

Other issues

- -----
- -----
- -----
- -----
- -----

Changelog

Changes are posted here to keep you aware of the latest developments in the Example Web SDK.

Fixed:

- -----
- -----
- -----
- -----
- -----

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Technical and Product Glossary

Overview

The Example Technical and Product Glossary provides terms with definitions related to Example's technologies. Terms are followed by definitions, code samples (when relevant) and references for further study. This is a living document and new data will be added to it when available.

A

AAR (Android Archive Library)
The Example Android SDK Android Archive Library (.aar) is a key ingredient of the Example SDK integration process. Android library packages allow developers to bundle Android libraries, resource files, etc., into binary packages that can be used across multiple projects without providing source code.
Reference: Example Android SDK Android Archive Library

Access Coarse Location
An Android system user permission. Example technologies require this to access approximate location functionality.
Reference: Access Coarse Location *(http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_COARSE_LOCATION)*

Access Fine Location
An Android system user permission. Example technologies require this to access precise location functionality.
Reference: Access Fine Location *(http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_FINE_LOCATION)*

Access Network State
An Android system user permission. Example technologies require this to access information about networks.
Reference: Access Network State *(http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_NETWORK_STATE)*

Access Wifi State
An Android system user permission. Example technologies require this to access information about wifi networks.
Reference: Access Wifi State *(http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_WIFI_STATE/)*

activity
A parameter of the init() Method that belongs to the Example Class. Its Java data type is Activity which is an Android class representing a user experience workflow.
Reference: init()

amount
A parameter of the deposit() Method that belongs to the Example Class. Its Java data type is float. This is the actual amount to initiate the deposit with as a preset.
Reference: deposit()

Android APK Workflow
To develop apps for Android, you use a set of tools that are included in Android Studio. In addition to using the tools from Android Studio, you can also access most of the SDK tools from the command line. Developing with Android Studio is the preferred method because it can directly invoke the tools that you need while developing applications.
Reference: Android APK Workflow *(http://developer.android.com/tools/workflow/index.html)*

Android Operating Systems support
The Example Android SDK provides support for the following Android OS versions.

- *Minimum supported version*: Android 4.0 (API level 14)
- *Reference:* Android Codenames, Tags, and Build Numbers *(https://source.android.com/source/build-numbers.html)*

Android SDK

The Example Android SDK is integrated into your individual casino games. It provides the integration with the Example Store Application and payment services for your applications.
Reference: See the latest version of the Example Android SDK.

Android Studio
Android Studio is the official IDE for Android app development and is based on IntelliJ IDEA. The Example Android SDK provides integration instructions and tools for use with the Android SDK.
Reference:

- Google Android Studio *(http://developer.android.com/sdk/index.html)*
- Getting Started with Android Studio *(http://developer.android.com/develop/index.html)*
- Android APK Workflow *(http://developer.android.com/tools/workflow/index.html)*
- Android User Interface (UI) *(http://developer.android.com/guide/topics/ui/overview.html)*

Android User Interface
All user interface elements in an Android app are built using View and ViewGroup objects. A View is an object that draws something on the screen that the user can interact with. A ViewGroup is an object that holds other View (and ViewGroup) objects in order to define the layout of the interface.
Reference: Android User Interface *(http://developer.android.com/guide/topics/ui/overview.html)*

appcompat-v7 Library
This Android library adds support for the Action Bar user interface design pattern. It includes support for material design user interface implementations.
Reference: appcompat-v7 Library *(http://developer.android.com/tools/support-library/features.html#v7-appcompat)*

App History
This is an Android System user permission that retrieves the list of installed apps.
Reference: App History
(http://developer.android.com/reference/android/content/pm/PackageManager.html#getInstalledPackages(int))

App Information Page
An app information page promotes your app, provides players with information, and drives downloads. To effectively drive consumers to your app, you'll need to supply the following items:

- A featured image
- A featured app preview video
- An app icon
- An app description
- App screenshots

Reference: Example App Information Page *(http://sb.dicemedia.com/Example.com/developer/web/app-information-page/)*

App integration into the Example Store
This consists of the following steps as described in this document:

1. Sign up for an account in the Example Early Access Program.
2. Develop your gambling app and integrate it with the latest Example Android SDK.
3. Submit your gambling app to be certified by an authorized agency.
4. Submit your app to Example

Reference: Integrating your app into the Example Store *(http://sb.dicemedia.com/Example.com/developer/web/integrating-your-app-into-the-Example-store/)*

App submission preparation
Pay close attention to preparing your app with Example's technologies and having it placed in the Example Store. There are three tasks to complete:

- Phase 1. Integrate the latest Example Android SDK with your Android gaming app.
- Phase 2. Complete the initial checklist presented in this document.
- Phase 3. Go over the final checkpoints and make sure you are compliant with all items.

Reference: Get your app ready for submission *(http://sb.dicemedia.com/Example.com/developer/web/get-your-app-ready-for-submission/)*

App testing
Testing the release version of your application helps ensure that your application runs properly under realistic device and network conditions. Ideally, you should test your application on at least one handset-sized device and one tablet-sized device to verify that your user interface elements are sized correctly and that your application's performance and battery efficiency are acceptable.
Reference: Test your app *(http://sb.dicemedia.com/Example.com/developer/web/test-your-app/)*

Architecture
The foundation of Example's system architecture is comprised of key back-end services and client distributables.

- *Back-end services* are the Example Core System and the Example Transaction System.
- *Client distributables* are the Example Store Application (.apk) and the Example Android SDK.

Audience

The Example Android SDK is intended for use by Example Partner developers, engineers, and integrators who work with the Google Android APK application build process.
Reference: Example Android SDK Introduction: Getting Started

B

Back-end services
Back-end services are part of the Example System Architecture consisting of:

- **Example Core System:** This is the main controller managing user activities and provides the integration with the casino's back-end servers.
- **Example Transaction System:** This is responsible for all payment-related activities, including management of payment processing interactions and user payment methods.
- *Reference:* Example Android SDK Getting Started > System architecture overview

C

checkUpdates()
A Example Java Method of the `Example` Class that checks for new updates to a game app.
Reference: `checkUpdates()`

Class (Java)
In object-oriented programming terms, a class is the blueprint from which individual objects are created. A class can also be defined as a template/blue print that describes the behaviors/states that objects of its type support.
Reference: What Is a Class? *(https://docs.oracle.com/javase/tutorial/java/concepts/class.html)*

Client distributables
Client distributables are part of the Example System Architecture consisting of:

- **Example Store Application:** This app will be installed by consumers onto their Android devices and allows a consumer to browse, search for, install, and update casino gaming applications.
- **Example Android SDK:** This is integrated into your individual casino games. It provides the integration with the Example Store Application and payment services for your applications.

Reference: Example Android SDK Introduction: Getting Started

Codenames, Tags, and Build Numbers
At a high level, Android development happens around families of releases, which use code names ordered alphabetically after tasty treats.
Reference: Codenames, Tags, and Build Numbers *(https://source.android.com/source/build-numbers.html)*

Confidential Information
Means any information that is proprietary or confidential to the Discloser or that the Discloser is obligated to keep confidential (e.g., pursuant to a contractual or other obligation owing to a third party). Confidential Information may be of a technical, business or other nature (including, but not limited to, information which relates to the Discloser's technology, research, development, products, services, pricing of products and services, customers, employees, contractors, marketing plans, finances, contracts, legal affairs, or business affairs).
Reference: Example EAPA

Consumers
Example consumers are end-users of the Example Store and the gambling game apps within it.
Reference: Gamer Demographics that Every Developer Should Know *(http://blog.apptopia.com/game-demographics-that-every-developer-should-know/)*

Core System
The Example Core System is the main controller managing user activities and provides the integration with the casino's back-end servers.
Reference: Example Android SDK Introduction: Getting Started

Customers
Example customers are gambling casinos and gambling game operators.
Reference: David Chang – Example – The Android Challenge *(http://www.sbcnews.co.uk/europe/2016/03/17/david-chang-Example-the-android-challenge/#ixzz439kGg4Tr)*

D

data
A parameter of the onActivityResult() Method which belongs to the Example Class. Its Java data type is Intent. It is the standard container used to communicate any extra data in a response from an Intent call (used by the SDK to capture external responses).
Reference: onActivityResult()

deposit - Parameter
A parameter of the onExampleDepositSuccess(double deposit) Method which belongs to the Example.DepositListener Interface. Its Java data type is Double. It is called by the SDK when a deposit is successful and returns the amount successfully deposited to the player's account as parameter deposit.
Reference: onExampleDepositSuccess()

deposit(float amount) - Method
A Example Java Method with Parameters of the `Example` Class that allows a user to start an in-app payment form and make a deposit.
Reference: `deposit(float amount)`

depositListener
A parameter of the setDepositListener() Method which belongs to the Example Class. Its Java data type is Example.DepositListener. It is a Class instance, provided by the developer at runtime. It is any Class that implements the Interface Example.DepositListener.
Reference: setDepositListener()

Design Support Library
The Android Design Library package provides APIs to support adding material design components and patterns to your apps. It adds support for various material design components and patterns for app developers to build upon, such as navigation drawers, floating action buttons (FAB), snackbars, and tabs.
Reference: Design Support Library *(http://developer.android.com/tools/support-library/features.html#design)*

destroy()
A Example SDK Java Method of the `Example` Class that recycles all held instances.
Reference: `destroy()`

Discloser
Means a Party that discloses any of its Confidential Information to the other Party.
Reference: Example EAPA

Documentation
Means the user manuals, technical manuals, specifications and other documentation relating to any Example Services furnished by Example to Customer under the Example EAPA Agreement.
Reference: Example EAPA

E

errorMessage
A parameter of the onExampleMarketError() Method which belongs to the Example.MarketListener Interface. Its Java data type is String. It is called by the SDK when an error has occurred during any interactions, returning an error message to the developer stating the reason.
Reference: onExampleMarketError()

Evaluation Period
Means the period commencing on the date the Example Services are made available to Customer and ending December 30, 2016.
Reference: Example EAPA

Example Java Class
A Example SDK Java utility Class that enables and allows communication with the SDK, and contains the following eight Methods:

- `checkUpdates()` - Method that checks new updates for a game app.

- `deposit(float amount)` - Method that allows a user to start an in-app payment form and make a deposit.
- `destroy()` - Method that recycles all held instances.
- `init(Activity activity)` - Method that initializes the singleton instance of Example.
- `onActivityResult(int requestCode, int resultCode, Intent data)` - Method for the results of activities.
- `onResume()` - Method for the resumption of a game.
- `setDepositListener(Example.DepositListener depositListener)` - Method that sets the deposit listener.
- `setMarketListener(Example.MarketListener marketListener)` - Method that sets the market listener.
- `verifyMarketAuthorization()` - Method that verifies the Example app is installed and a user is logged in.

Reference: `Example` Java Class

Example.DepositListener Java Interface

This is an Interface which enables and allows listening to deposit events, and contains the following three Methods:

- `onExampleDepositCancel()` - Method that signals a canceled deposit.
- `onExampleDepositFail()` - Method that signals a failed deposit.
- `onExampleDepositSuccess(double deposit)` - Method that signals a successful deposit.

Reference: `Example.DepositListener`

Example.MarketListener Java Interface

This is an Interface used to listen for general failures. It contains the following Method:
`onExampleMarketError(java.lang.String errorMessage)`
Reference: `Example.MarketListener`

Example, LLC

Launching in July 2016, Example is the first dedicated Android app store for the gaming industry and the premier destination for players discover, download and manage the best gaming apps. Example provides a trusted and secure branded app store that makes it easy for operators to distribute and promote their real money gaming apps, and for players to discover, download and play all of the games they love. Once players install Example and download their favorite real-money apps, management, updates and deposits are seamless. Founded to create the first Android ecosystem specifically for gaming, Example's executive team has deep roots in gaming and mobile industries. The company is privately held with offices in Los Angeles and Silicon Valley.
Reference: Example LLC homepage *(http://www.Example.com)*

Example Marks

Means any trademarks, service marks, service or trade names, logos, and other designations of Example and its affiliates.
Reference: Example EAPA

Example Parties

Means Example and its affiliates, independent contractors and service providers, and each of their respective members, directors, officers, employees and agents.
Reference: Example EAPA

Example Services

Means Example's app store, together with any Software provided or made available by Example.
Reference: Example EAPA

F

Feedback

Means information and feedback (including, without limitation, questions, comments, suggestions, or the like) regarding the performance, features, functionality and your overall experience using the Example Services.
Reference: Example EAPA

G

Geolocation

An Android system user permission. Example requires geolocation to provide location-relevant search results.
Reference: Geolocation *(http://developer.android.com/reference/android/webkit/GeolocationPermissions.html)*

Get Accounts

An Android system user permission. This authenticates and synchronizes Example account information between apps and devices.
Reference: Get Accounts *(http://developer.android.com/reference/android/Manifest.permission.html#GET_ACCOUNTS)*

Google Play Services

Google Play services enables your app to take advantage of the latest, Google-powered features such as Maps, Google+, etc. with automatic platform updates distributed as an APK through the Google Play store.
Reference: Overview: Google Play Services *(https://developers.google.com/android/guides/overview)*

Gradle

Gradle is the official build system for Android Studio.
Reference: Getting Started with Gradle *(https://gradle.org/getting-started-android/)*

Gson

Gson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object.
Reference: Gson User Guide *(https://github.com/google/gson/blob/master/UserGuide.md#TOC-Overview)*

H

I

init(Activity activity)

A Example SDK Java Method with Parameters of the `Example` Class that initializes the singleton instance of Example.
Reference: `init(Activity activity)`

Integration (SDK)

This Example SDK document provides steps to perform a Example Android SDK integration using the Android Studio IDE.
Reference: SDK Integration using Android Studio

Intellectual Property Rights

Means any patent, copyright, trademark, service mark, trade name, trade secret, know-how, moral right or other intellectual property right under the laws of any jurisdiction, whether registered, unregistered, statutory, common law or otherwise (including any rights to sue, recover damages or obtain relief for any past infringement, and any rights under any application, assignment, license, legal opinion or search).
Reference: Example EAPA

Interface (Java)

An interface is a reference type in Java, it is similar to class and is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface. Along with abstract methods an interface may also contain constants, default methods, static methods, and nested types.
Reference: What Is an Interface? *(https://docs.oracle.com/javase/tutorial/java/concepts/interface.html)*

Internet

An Android system user permission. Example technologies require this to open network sockets.
Reference: Internet *(http://developer.android.com/reference/android/Manifest.permission.html#INTERNET)*

J

Java Class Library

The Example Java Class and its Interfaces, Methods, and Parameters are of primary importance and use during the integration process. They are documented in this page.
Reference: Example Android SDK Java Class Library

K

L

Location Services

The Example Android SDK requires the Android OS Location Services package. Android gives your applications access to the location services supported by the device through classes in the `android.location` package. The central component of the location framework is the `LocationManager` system service, which provides APIs to determine location and bearing of the underlying device (if available).
Reference: Location Services *(http://developer.android.com/guide/topics/location/index.html#location)*

M

marketListener
A parameter of the setMarketListener() Method which belongs to the Example Class. Its Java data type is Example.MarketListener. It is a Class instance provided by the developer at runtime. It is any Class that implements the Interface Example.MarketListener
Reference: setMarketListener()

Materials
Means content of any data or information that Customer provides to the Example Services.
Reference: Example EAPA

Method (Java)
A method is a set of code which is referred to by name and can be called (invoked) at any point in a program simply by utilizing the method's name. Think of a method as a subprogram that acts on data and often returns a value. Each method has its own name. When that name is encountered in a program, the execution of the program branches to the body of that method. When the method is finished, execution returns to the area of the program code from which it was called, and the program continues on to the next line of code.
Reference: Defining Methods *(https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html)*

N

O

onActivityResult(int requestCode, int resultCode, Intent data)
A Example SDK Method of the `Example` Class with Parameters used for the results of activities.
Reference: `onActivityResult(int requestCode, int resultCode, Intent data)`

onExampleDepositCancel()
A `Example.DepositListener` Method that signals a canceled deposit.
Reference: `onExampleDepositCancel()`

onExampleDepositFail()
A `Example.DepositListener` Method that signals a failed deposit.
Reference: `onExampleDepositFail()`

onExampleDepositSuccess(double deposit)
A `Example.DepositListener` Method that signals a successful deposit.
Reference: `onExampleDepositSuccess(double deposit)`

onResume()
A Example Java Method of the `Example` Class for the resumption of a game.
Reference: `onResume()`

Operator information
The operator information page promotes your brand and all of the gaming apps you're offering on the Example store. It should consist of the following items:

- Your featured image
- Your operator logo
- Your operator description
- Your operator apps

Reference: Example Operator Information *(http://sb.dicemedia.com/Example.com/developer/web/operator-info-page/)*

P

Parameter (Java)
In Java there are two types of parameters, implicit parameters and explicit parameters. Explicit parameters are the arguments passed into a method. The implicit parameter of a method is the instance that the method is called from. Arguments are simply one of the two types of parameters.
Reference: Passing Information to a Method or a Constructor *(https://docs.oracle.com/javase/tutorial/java/javaOO/arguments.html)*

Partners
Example partners are gambling game app developers and operators.
Reference: How To Make Money With Casino Apps *(http://www.bluecloudsolutions.com/source-codes/party_slots_b/#)*

Party
Means Example or Customer.
Reference: Example EAPA

Policies
Means Example’s acceptable use policy and Example’s privacy policy as found on the Site, all restrictions described on the Site, and any other policy or terms incorporated by reference into this Example EAPA Agreement.
Reference: Example EAPA

Q

R

Read External Storage (Read/Write)
This an Android user permission that is required to read and write from an external storage.
Reference: Read External Storage (Read/Write) *(http://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE)*

Read Phone State
This is an Android System user permission that is required to get Device specific details such as device id, OS, etc.
Reference: Read Phone State *(http://developer.android.com/reference/android/Manifest.permission.html#READ_PHONE_STATE)*

Recipient
Means a Party that receives any Confidential Information of the other Party.
Reference: Example EAPA

recyclerview-v7 Library
The recyclerview library adds the RecyclerView class. This class provides support for the RecyclerView widget, a view for efficiently displaying large data sets by providing a limited window of data items.
Reference: recyclerview-v7 Library *(http://developer.android.com/tools/support-library/features.html#v7-recyclerview)*

requestCode
A parameter of the onActivityResult() Method which belongs to the Example Class. Its Java data type is integer. It is a unique request code, numeric, provided by the developer when the call was started (unused in the SDK, default for platform signature).
Reference: onActivityResult()

resultCode
A parameter of the onActivityResult() Method which belongs to the Example Class. Its Java data type is integer. It is a unique result code returned to the developer to indicate the operation status (used by the SDK to capture external responses).
Reference: onActivityResult()

S

SDK (Android)
The Example Android SDK is integrated into your individual casino games. It provides the integration with the Example Store Application and payment services for your applications.
Reference: See the latest version of the Example Android SDK.

setDepositListener(Example.DepositListener depositListener)
A Example SDK Java Method of the `Example` Class with an Interface and Parameter that sets the deposit listener.
Reference: `setDepositListener(Example.DepositListener depositListener)`

setMarketListener(Example.MarketListener marketListener)
A Example SDK Java Method of the `Example` with an Interface and Parameter that sets the market listener.
Reference: `setMarketListener(Example.MarketListener marketListener)`

Site
Means www.Example.com *(http://www.Example.com)* , and any successor or related site designated by Example.
Reference: Example EAPA

Software
Means any down-loadable software made available by Example and required for use of the Example Services.
Reference: Example EAPA

Store Application
The Example Store Application (.apk) will be installed by consumers onto their Android devices. It allows a consumer to browse, search for, install, and update casino gaming applications. It requires the consumer to create a user account which is registered with the Example Core System, and provide a payment method which is registered with the Example Transaction System. The payment method is validated through the Bank System and stored in the Transaction System.
Reference: See the Example Android SDK "Getting Started" page System Architecture overview section.

support-v4
The Android Support Library package is a set of code libraries that provide backward-compatible versions of Android framework APIs as well as features that are only available through the library APIs.
Reference: Android Support Libraries *(http://developer.android.com/tools/support-library/index.html#)*

Support Library Setup
How you setup the Android Support Libraries in your development project depends on what features you want to use and what range of Android platform versions you want to support with your application. This document guides you through downloading the Support Library package and adding libraries to your development environment.
Reference: Support Library Setup *(https://developer.android.com/tools/support-library/setup.html)*

T

Trademarks

As between Example and Customer, Example owns all right, title and interest in and to the Example Marks and any goodwill arising out of the use of the Example Marks will remain with and belong to Example and its licensors. The Example Marks may not be copied, imitated or used without the prior written consent of Example or the applicable trademark holder.
Reference: Example EAPA

Transaction System

The Example Transaction System is responsible for all payment-related activities, including management of payment processing interactions and user payment methods.
Reference: Example Android SDK Getting Started > System architecture overview

U

Unauthorized Use

Means any use, reproduction, modification, distribution, disposition, possession, examination, inspection, viewing, disclosure or other activity involving the Example Services, Documentation or Confidential Information of Example that is not expressly authorized under the Agreement or otherwise in writing by Example.
Reference: Example EAPA

User Permissions

The Example Android SDK requires specific user permissions to operate effectively. Android applications should add the following permissions to their user permission model.

- Geolocation *(http://developer.android.com/reference/android/webkit/GeolocationPermissions.html)* : Example requires geolocation to provide location-relevant search results.
- Internet *(http://developer.android.com/reference/android/Manifest.permission.html#INTERNET)* : Required to open network sockets.
- Access Network State *(http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_NETWORK_STATE)* : Required to access information about networks.
- Access Wifi State *(http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_WIFI_STATE)* : Required to access information about Wifi networks.
- Read Phone State *(http://developer.android.com/reference/android/Manifest.permission.html#READ_PHONE_STATE)* : Required to get Device specific details such as device id, OS, etc.
- Read External Storage (Read/Write) *(http://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE)* : Required to read and write from an external storage.
- Access Fine Location *(http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_FINE_LOCATION)* : Required to fetch to access precise location functionality.
- Access Coarse Location *(http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_COARSE_LOCATION)* : Required to access approximate location functionality.
- App History *(http://developer.android.com/reference/android/content/pm/PackageManager.html#getInstalledPackages(int))* : Retrieve the list of installed apps.
- Get Accounts *(http://developer.android.com/reference/android/Manifest.permission.html#GET_ACCOUNTS)* : Authenticate and synchronize Example account information between apps and devices.
- Wake Lock *(http://developer.android.com/reference/android/Manifest.permission.html#WAKE_LOCK)* : Required for certain synchronization events to be handled in the background.

V

verifyMarketAuthorization()

A Example SDK Method of the `Example` Class that verifies the Example app is installed and a user is logged in.
Reference: `verifyMarketAuthorization()`

W - Z

Wake Lock

An Android system user permission. Example technologies require this for certain synchronization events to be handled in the background.
Reference: Wake Lock *(http://developer.android.com/reference/android/Manifest.permission.html#WAKE_LOCK)*

HTML5 Web SDK v1.0 beta 1

Updated: 2011.01.10

Security classification: PUBLIC

Early Adopter Program Agreement

> **Note:** This is an example of an Early Adopter Program Agreement / End User License Agreement.

The Early Adopter Program Agreement (the "Agreement") between Example, Inc. ("Example") and the Customer.

Example and Customer agree as follows:

Section 1. Definitions

Whenever used in this Agreement with initial letters capitalized, the following terms will have the following specified meanings:

"Example Marks"
Means any trademarks, service marks, service or trade names, logos, and other designations of Example and its affiliates.

"Example Parties"
Means Example and its affiliates, independent contractors and service providers, and each of their respective members, directors, officers, employees and agents.

"Example Services"
Means Example's app store, together with any Software provided or made available by Example.

"Confidential Information"
Means any information that is proprietary or confidential to the Discloser or that the Discloser is obligated to keep confidential (e.g., pursuant to a contractual or other obligation owing to a third party). Confidential Information may be of a technical, business or other nature (including, but not limited to, information which relates to the Discloser's technology, research, development, products, services, pricing of products and services, customers, employees, contractors, marketing plans, finances, contracts, legal affairs, or business affairs). However, Confidential Information does not include any information that: (a) was known to the Recipient prior to receiving the same from the Discloser in connection with this Agreement; (b) is independently developed by the Recipient; (c) is acquired by the Recipient from another source including any end user of the Example Services without restriction as to use or disclosure; or (d) is or becomes part of the public domain through no fault or action of the Recipient.

"Discloser"
Means a Party that discloses any of its Confidential Information to the other Party.

"Documentation"
Means the user manuals, technical manuals, specifications and other documentation relating to any Example Services furnished by Example to Customer under this Agreement.

"Evaluation Period"
Means the period commencing on the date the Example Services are made available to Customer and ending December 30, 2016.

"Feedback"
Means information and feedback (including, without limitation, questions, comments, suggestions, or the like) regarding the performance, features, functionality and your overall experience using the Example Services.

"Intellectual Property Rights"

Means any patent, copyright, trademark, service mark, trade name, trade secret, know-how, moral right or other intellectual property right under the laws of any jurisdiction, whether registered, unregistered, statutory, common law or otherwise (including any rights to sue, recover damages or obtain relief for any past infringement, and any rights under any application, assignment, license, legal opinion or search).

"Materials"

Means content of any data or information that Customer provides to the Example Services.

"Party"

Means Example or Customer.

"Policies"

Means Example's acceptable use policy and Example's privacy policy as found on the Site, all restrictions described on the Site, and any other policy or terms incorporated by reference into this Agreement.

"Recipient"

means a Party that receives any Confidential Information of the other Party.

"Site"

Means www.Example.com *(http://www.Example.com)* , and any successor or related site designated by Example.

"Software"

Means any downloadable software made available by Example and required for use of the Example Services.

"Unauthorized Use"

Means any use, reproduction, modification, distribution, disposition, possession, examination, inspection, viewing, disclosure or other activity involving the Example Services, Documentation or Confidential Information of Example that is not expressly authorized under the Agreement or otherwise in writing by Example.

Section 2. License

2.1 License to Use Example Services.

Subject to the terms of this Agreement, Example hereby grants to Customer a limited, nonexclusive, nontransferable, nonsublicensable, revocable, royalty free license to use the Example Services solely for Customer's own internal testing and evaluation of the Example Services for the Evaluation Period.

2.2 Software.

If Customer receives Software from Example, including, without limitation Software downloaded separate from this Agreement or from a third party marketplace, its use is governed in one of two ways: if Customer is presented with license terms that it must accept in order to use the Software, those terms apply; if no license is presented, this Agreement applies. Example reserves all other rights to the Software. Any Software is licensed, not sold. Unless Example notifies Customer otherwise, the Software license ends when Customer's Example Services end. Customer must then promptly uninstall the Software from all of Customer's computers or devices, or Example may disable it. Customer must not work around any technical limitations in the Software.

2.3 Restrictions; Limitations.

Customer may not use the Example Services in any manner or for any purpose other than as expressly permitted by this Agreement. Without limitation of the foregoing, the license granted under this Section 2 does not include or authorize: (a) publicly performing or publicly displaying any of the Example Services; (b) modifying or otherwise making any derivative use of any of the Example Services; (c) using any data mining, robots or similar data gathering or extraction methods; (d) reverse engineering or accessing any of the Example Services to build a competitive product or service; or (e) using any of the Example Services other than for their intended use. During and after the Evaluation Period, Customer will not assert, nor authorize, assist or encourage any third party to assert, against any of the Example Parties, any patent infringement or other intellectual property infringement claim regarding any Example Services Customer has used. The license granted under this Section 2 is conditioned on Customer's continued compliance this Agreement (including, without limitation, its compliance with the Policies), and will immediately and automatically terminate if Customer does not comply with any material term or condition of this Agreement.

Section 3. Confidential Information

Each Party reserves any and all right, title and interest (including, without limitation, any Intellectual Property Rights) that it may have in or to any Confidential Information that it may disclose to the other Party under this Agreement. The Recipient will protect Confidential Information of the Discloser against any unauthorized use or disclosure to the same extent that the Recipient protects its own Confidential Information of a similar nature against unauthorized use or disclosure, but in no event will use less than a reasonable standard of care to protect such Confidential Information; provided that the Confidential Information of the Discloser is conspicuously marked or otherwise identified as confidential or proprietary upon receipt by the Recipient or the Recipient otherwise knows or has reason to know that the same is Confidential Information of the Discloser. The Recipient will use any Confidential Information of the Discloser solely for the purposes for which it is provided by the Discloser. This paragraph will not be interpreted or construed to prohibit: (a) any use or disclosure which is necessary or appropriate in connection with the Recipient's performance of its obligations or exercise of its rights under this Agreement or any other agreement between the Parties; (b) any use or disclosure required by applicable law (e.g., pursuant to applicable securities laws or legal process), provided that the Recipient uses reasonable efforts to give the Discloser reasonable advance notice thereof (e.g., so as to afford the Discloser an opportunity to intervene and seek an order or other appropriate relief for the protection of its Confidential Information from any unauthorized use or disclosure); or (c) any use or disclosure made with the consent of the Discloser. In the event of any breach or threatened breach by the Recipient of its obligations under this paragraph, the Discloser will be entitled to injunctive and other equitable relief to enforce such obligations.

Section 4. Proprietary Rights

4.1 The Example Services.

As between Example and Customer, Example owns all right, title, and interest in and to the Example Services. Except as otherwise specified in Sections 2.1 and 2.2 of this Agreement, Customer does not obtain any rights under this Agreement from Example to the Example Services, including any related Intellectual Property Rights.

4.2 Feedback.

Customer will at its risk and expense use reasonable efforts to provide Example with reasonable Feedback. Customer will not provide any such Feedback to any third party without Example's prior written consent in each instance. Customer hereby grants to Example an exclusive, royalty-free, irrevocable, perpetual worldwide right and license to reproduce, use, disclose, exhibit, display, transform, create derivative works and distribute any such Feedback without limitation. Further, Example will be free to use any ideas, concepts, know-how or techniques contained in such Feedback for any purpose whatsoever, including, without limitation, developing, making, marketing, distributing and selling products and services incorporating such Feedback. Customer should not expect any compensation of any kind from Example with respect to Feedback. Example will exclusively own any improvements or modifications to the Example services based on or derived from any Feedback, including all Intellectual Property Rights therein or thereto.

4.3 Trademarks.

As between Example and Customer, Example owns all right, title and interest in and to the Example Marks and any goodwill arising out of the use of the Example Marks will remain with and belong to Example and its licensors. The Example Marks may not be copied, imitated or used without the prior written consent of Example or the applicable trademark holder.

4.4 Additional Protection of Proprietary Rights.

Customer will not infringe or violate, and will take appropriate steps and precautions for the protection of, the Example Services and related Intellectual Property Rights referred to in Section 4.1. Without limiting the generality of the foregoing, Customer will not: (a) make any Example Services or Documentation available to any third party without the prior written consent of Example; (b) reverse engineer, disassemble or decompile the Example Services or otherwise attempt to discover or recreate the source code to the Example Services; (c) interfere or attempt to interfere in any manner with the functionality or proper working of the Example Services; (d) except as permitted by Example, create, build, or otherwise develop a commercial website utilizing the Example Services; (e) remove, obscure or alter any notice of copyright or other Intellectual Property Right appearing in or as part of the Example Services; or (f) engage in or permit any Unauthorized Use. Customer will immediately notify Example of any Unauthorized Use that comes to Customer's attention. In the event of any Unauthorized Use relating to the activities of Customer, or any employees, agents, or representatives of Customer, Customer will take all steps reasonably necessary to terminate such Unauthorized Use. Customer will immediately notify Example of any legal proceeding initiated by Customer in connection with any such Unauthorized Use. Example may, at its option and expense, assume control of such proceeding. If Example assumes such control, Example will have exclusive control over the prosecution and settlement of the proceeding, and Customer will provide such assistance related to such proceeding as Example may reasonably request. Customer will assist Example in enforcing any settlement or order made in connection with such proceeding.

Section 5. Termination.

Either Party may terminate this Agreement at any time by written notice of termination to the other Party. Unless otherwise agreed to by the Parties in writing, upon the expiration of the Evaluation Period or the termination of this Agreement by either Party as provided above, the license granted to Customer in this Agreement will terminate. Sections 1, 2.3, 3, 4, 6, 7 and 8 shall survive the termination or expiration of this Agreement.

Section 6. Limited Warranties and Remedies

6.1 DISCLAIMER.

Customer acknowledges that the Example Services are a beta test version, are not ready for general commercial release and may contain bugs, errors and defects. Accordingly, the Example Services are provided by Example and accepted by Customer "AS IS" and "WITH ALL FAULTS, DEFECTS AND ERRORS." Example WILL HAVE NO LIABILITY FOR ANY ERROR, OMISSION OR DEFECT IN THE Example SERVICES, OR ANY INABILITY TO ACCESS OR USE THE Example SERVICES. Example MAKES NO, AND HEREBY DISCLAIMS, ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, ARISING BY LAW OR OTHERWISE, REGARDING THE Example SERVICES, ANY SERVICE PROVIDED IN CONNECTION WITH THE Example SERVICES OR ANY OTHER ITEMS, INFORMATION, MATERIALS OR DATA PROVIDED BY Example IN CONNECTION WITH THE EVALUATION OF THE Example SERVICES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT.

Section 7. Limitations of Liability

IN NO EVENT SHALL ANY OF THE Example PARTIES BE LIABLE FOR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY OTHER DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO LOSS OF USE, LOSS OF PROFITS OR LOSS OF DATA, WHETHER IN AN ACTION IN CONTRACT, TORT (INCLUDING BUT NOT LIMITED TO NEGLIGENCE) OR OTHERWISE, ARISING OUT OF OR IN ANY WAY CONNECTED WITH THE USE OF OR INABILITY TO USE THE Example SERVICES OR THE Example SITE, INCLUDING THE INFORMATION, CONTENT AND MATERIALS CONTAINED THEREIN, INCLUDING WITHOUT LIMITATION ANY DAMAGES CAUSED BY OR RESULTING FROM RELIANCE BY CUSTOMER ON ANY INFORMATION OBTAINED FROM Example, OR THAT RESULT FROM MISTAKES, OMISSIONS, INTERRUPTIONS, DELETION OF FILES, ERRORS, DEFECTS, VIRUSES, DELAYS IN OPERATION OR TRANSMISSION OR ANY FAILURE OF PERFORMANCE, WHETHER OR NOT RESULTING FROM ACTS OF GOD, COMMUNICATIONS FAILURE, THEFT, DESTRUCTION OR UNAUTHORIZED ACCESS TO Example'S RECORDS, PROGRAMS, Example SERVICE OR THE Example SITE. IN NO EVENT SHALL THE AGGREGATE LIABILITY OF ANY OF THE Example PARTIES, WHETHER IN CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE, WHETHER ACTIVE, PASSIVE OR IMPUTED), PRODUCT LIABILITY, STRICT LIABILITY OR OTHER THEORY, ARISING OUT OF OR RELATING TO THIS AGREEMENT EXCEED $100.

Section 8. Miscellaneous

8.1 Independent Contractors.

Each Party is an independent contractor and not a partner or agent of the other. This Agreement will not be interpreted or construed as creating or evidencing any partnership or agency between the Parties or as imposing any partnership or agency obligations or liability upon either Party. Further, neither Party is authorized to, and will not, enter into or incur any agreement, contract, commitment, obligation or liability in the name of or otherwise on behalf of the other Party.

8.2 Assignment.

Customer may not assign this Agreement or any right, interest or benefit under this Agreement without prior written consent of Example. Any attempted assignment in violation of the foregoing will be void. Subject to the foregoing, this Agreement will be fully binding upon, inure to the benefit of and be enforceable by any permitted assignee.

8.3 Dispute Resolution Procedures.

(a) The Parties will attempt to resolve through good faith discussion any dispute that arises under this Agreement. Any such dispute may at any time, at the election of either Party, be referred to a senior executive of each Party for discussion and possible resolution. If the senior executives are unable to resolve the dispute within sixty (60) days after delivery of written notice of the dispute, then either Party may, by notice to the other Party, demand mediation under the mediation rules of JAMS in Los Angeles, California. The Parties give up their right to litigate their disputes and may not proceed to arbitration without first attempting mediation, except that the Parties are NOT required to arbitrate any dispute in which either Party seeks equitable and other relief from the alleged unlawful use any Intellectual Property Rights by the other Party. Whether the dispute is heard in arbitration or in court, the Parties will not commence against the other a class action, class arbitration or other representative action or proceeding.

(b) If settlement is not reached within sixty (60) days after service of a written demand for mediation, any unresolved controversy or claim shall be resolved by arbitration in accordance with the rules of JAMS before a single arbitrator in Los Angeles, California. The language of all proceedings and filings shall be English. The arbitrator shall render a written opinion including findings of fact and law and the award and/or determination of the arbitrator shall be binding upon the Parties, and their respective administrators and assigns, and shall not be subject to appeal. Judgment may be entered upon the award of the arbitrator in any court of competent jurisdiction. The expenses of the arbitration shall be shared equally by the Parties unless the arbitration determines that the expenses shall be otherwise assessed and the prevailing Party may be awarded its attorneys' fees and expenses by the arbitrator. It is the intent of the Parties that, barring extraordinary circumstances, arbitration proceedings shall be concluded within ninety (90) days from the date the arbitrator is appointed. The arbitrator may extend this time limit only if failure to do so would unduly prejudice the rights of the Parties. Failure to adhere to this time limit shall not constitute a basis for challenging the award. Consistent with the expedited nature of arbitration, pre-hearing information exchange shall be limited to the reasonable production of relevant, non-privileged documents, carried out expeditiously.

8.4 Severability.

If any provision of this Agreement shall be deemed unlawful, void or for any reason unenforceable, then that provision shall be deemed severable from this Agreement and shall not affect the validity and enforceability of any remaining provisions.

8.5 Applicable Law.

This Agreement will be interpreted, construed and enforced in all respects in accordance with the laws of the State of Delaware, U.S.A., without reference to its choice of law principles to the contrary. Subject to Section 8.3, Customer hereby consents to the jurisdiction and venue of the state and federal courts located in Los Angeles County, State of California, U.S.A. with respect to any claim arising under or by reason of this Agreement.

8.6 Entire Agreement.

This Agreement, together with any agreement, policy or guideline referenced in this Agreement, constitutes the complete and exclusive statement of all mutual understandings between the parties with respect to the subject matter hereof, superseding all prior or contemporaneous proposals, communications and understandings, oral or written.