IntuitionLabs

INTUITIONLABS / RESEARCH & PRACTICAL GUIDANCE

Deep Linking in MyInsights: Code Patterns & JS API Guide

> Build practical AI for pharma and biotech with IntuitionLabs. We help life-science teams turn complex information and workflows into useful software, governed knowledge systems and AI tools.

Published by IntuitionLabs · 2026-01-11 · veeva myinsights · deep linking · veeva crm · myinsights javascript api · veeva development · ds.newrecord · pharmaceutical crm · sales analytics

Original article: https://intuitionlabs.ai/articles/myinsights-deep-linking-code-patterns



In this article

1. Executive Summary
2. Introduction
3. Deep Linking in MyInsights: Features and Patterns
4. Supported Deep Linking Tasks
5. Enabling Deep Linking
6. Deep Linking Code Patterns
7. Data Analysis and Evidence
8. User Efficiency and Retention Metrics
9. Adoption Rates in Industry
10. Case Example: GSK Field Reps
11. Case Studies and Use-Cases
12. Pre-Call and Call-Report Dashboards
13. Territory and Home Dashboards
14. Key Opinion Leader (KOL) Analytics
15. Orders and Samples
16. Manager/Coaching Dashboards
17. Implications and Future Directions
18. Implications for User Workflow
19. Performance and Data Considerations
20. Future Directions
21. Conclusion

[Revised May 3, 2026]

Executive Summary

Deep linking enables seamless navigation by directing users to specific content or actions within an application. In the context of Veeva CRM's **MyInsights** (a built-in analytics/dashboard framework), deep linking empowers sales and field teams to jump directly from data visualizations into actionable CRM features such as scheduling a call, creating records, or reviewing detailed account information (crmhelp.veeva.com) (arxiv.org). Historically, mobile and enterprise apps have lagged behind the Web in adopting deep linkable URLs, which can hamper user efficiency (arxiv.org) (arxiv.org). However, by implementing deep linking in MyInsights, organizations like GSK have demonstrated significant productivity gains – e.g. saving 30 minutes per rep per day in pre-call planning (www.veeva.com).

This report provides an in-depth analysis of **deep linking code patterns in MyInsights**, covering technical mechanisms, developer integration, business impact, and future directions. It draws upon Veeva official documentation, technical research, industry case studies, and user-journey analysis. Key findings include:

- **Supported Deep Linking Actions in MyInsights:** Veeva provides a JavaScript API (`com.veeva.clm`, globally accessible as `ds`) that includes methods like `ds.newRecord()`, `ds.viewRecord()`, `ds.viewSection()`, `ds.sendToMySchedule()`, `ds.executeSuggestionAction()`, and `ds.launchMediaForAccount()`. These methods correspond to common field activities (creating a call, viewing an account, scheduling calls, acting on suggestions, launching CLM presentations, etc.) directly from dashboard interfaces (developer.veevacrm.com) (developer.veevacrm.com). For example, `ds.newRecord({object:"Call2_vod__c", fields:{Account_vod__c:"001..."} })` opens the native "New Call" interface pre-populated with a given Account (developer.veevacrm.com). Similarly, `ds.viewRecord({object:"Call2_vod__c", fields:{Id:"<CallID>"}})` opens an existing record's detail page (developer.veevacrm.com). The `ds.viewSection()` method (introduced in CRM 23R3.0) switches to a specified tab or section within the current account or territory context (crmhelp.veeva.com) (developer.veevacrm.com).
- **MyInsights Architecture:** MyInsights dashboards are delivered as HTML/JavaScript apps embedded in the CRM UI (developer.veevacrm.com) (intuitionlabs.ai). An administrator uploads a ZIP bundle containing an `index.html` (entry point), supporting JS/CSS libraries, and vendor scripts into a special `HTML_Report_vod` record. When users open the dashboard, Veeva's embedded browser or Lightning Web Component container loads the `index.html`, which then uses the Veeva JS library (`com.veeva.clm`) to query CRM data and perform actions. DataService calls like `ds.queryRecord()` fetch CRM data (with single-quoted SOQL-like where clauses) (developer.veevacrm.com) (developer.veevacrm.com). Deep linking calls (`ds.newRecord`, etc.) launch CRM interfaces from within the dashboard. All MyInsights content respects CRM security and can run offline on the iPad (using local data sync) or online in Lightning (crmhelp.veeva.com).
- **Deep Linking Use Cases & Code Patterns:** Deep linking in MyInsights supports a variety of interactive scenarios. Common code patterns include invoking these API methods in response to user actions (e.g. click handlers) within the visualizations. For instance:

```
// Example: Launch a new Call record linked to a specific Account
var cfg = {
object: "Call2_vod__c",
fields: {
Account_vod__c: "00161000004vuXMAAY"
}
};
ds.newRecord(cfg).then(function(resp) {
// optional callback
});
```

This `newRecord()` call causes the CRM to open its native "New Call" page for the given account (developer.veevacrm.com). Likewise, to view an existing record:

```
// Example: View a specific Call record detail
var cfg = {
object: "Call2_vod__c",
fields: {
```

```
Id: "a040U000000UJPUQA4"
}
};
ds.viewRecord(cfg).then(function(resp) {
// callback after navigation
});
```

This navigates the user from the current MyInsights page into the Call detail page in CRM (developer.veevacrm.com). Calling `ds.viewSection({ /* section id or list */ })` on an account page instantly switches to another predefined tab within the account without leaving CRM (crmhelp.veeva.com). Other methods include `ds.sendToMySchedule({accountIds: ["ID1","ID2"]})` to open the rep's schedule with given accounts selected (developer.veevacrm.com), `ds.executeSuggestionAction(id, actionType)` to dismiss or complete a suggestion (developer.veevacrm.com), and `ds.launchMediaForAccount(accountId, presentationId, keyMessageName)` to open a CLM presentation for that account, possibly jumping to a specific key message (developer.veevacrm.com).

- **Development Best Practices:** Effective use of deep linking in MyInsights requires mindful coding. Developers must include the Veeva MyInsights JS library (and its Q promise library) in their HTML pages (developer.veevacrm.com). Queries must use single quotes and exact case-sensitive field names (developer.veevacrm.com). It's recommended to attach deep link calls to clearly labeled buttons or links so users understand the action (developer.veevacrm.com) (developer.veevacrm.com). Content should verify required object permissions are granted on the target CRM objects (e.g. `Call2_vod__c`, `Sent_Email_vod`) before enabling links (crmhelp.veeva.com). When designing the UX, note that certain entry points (like My Insights on the Call Report page) may not support some actions (e.g. `newRecord`) (crmhelp.veeva.com).
- **Business Impact and Case Studies:** Embedding deep-linked dashboards into CRM workflows can deliver measurable ROI. For example, GSK reported that, with an engagement-planning MyInsights page, its field teams saved **30 minutes per day** on average in engagement planning, by bringing call-preparation information into one view (www.veeva.com). That figure is self-reported, and GSK does not attribute it to deep linking specifically. An IntuitionLabs overview of MyInsights notes that pharma firms increasingly treat embedded analytics as part of CRM strategy (intuitionlabs.ai) (www.veeva.com). Marketing commentary on consumer apps, such as a MoldStud overview, claims that deep linking lifts retention, engagement and conversions (moldstud.com) (moldstud.com), but the uplift percentages circulating in such articles are not tied to an identifiable primary study and are not repeated here. The underlying principle is still reasonable: reducing navigation friction tends to make tasks quicker to complete.
- **Challenges and Adoption:** Despite the benefits, deep linking in enterprise apps has historically required extra developer effort and has seen limited adoption. Prior research on consumer apps found that **less than half** of leading Android apps had implemented deep links as of the late 2010s (speakerdeck.com) (arxiv.org). Creating URI handlers and updating app code is nontrivial (arxiv.org) (arxiv.org). In the MyInsights context, Veeva's SQL-like queries and JS API simplify much of this, but developers still must package the UX and ensure navigation targets exist (e.g. a valid Record ID). Veeva's CRM 23R3.0 release explicitly addressed some pain points: adding the `viewSection()` method

and augmenting `viewRecord()` to navigate to specific tabs (crmhelp.veeva.com) reduces the need for manual JavaScript in studio content. On Vault CRM, **X-Pages**, the successor to MyInsights, carries these navigation concepts forward (crmhelp.veeva.com).

Given this background, the report below delves into each aspect: technical implementation of deep links in MyInsights, code samples and patterns, empirical data and case examples, and future considerations for this capability.

Introduction

Background on MyInsights: Veeva CRM's *MyInsights* is a real-time analytics and dashboard framework integrated directly into the CRM interface (developer.veevacrm.com). It allows organizations (primarily in life sciences) to *"create custom content using the Veeva JavaScript library and HTML"* that runs as mini web apps embedded in user workflow pages (developer.veevacrm.com). Typical use cases include showing sales or call history, key opinion leader profiles, territory performance, or order stats – all surfaces that otherwise would require switching to separate BI tools. Because it leverages standard web technologies (HTML/CSS/JS) and Veeva's data APIs, developers can build highly customized reports. Content is packaged as a ZIP file containing an `index.html` entry point (with supplementary `.js` and `.css` files) attached to a special `HTML_Report_vod` record in CRM. When users with appropriate permissions access a record page, the corresponding MyInsights dashboards can appear as additional tabs or panels, running in either the Veeva iPad app (offline-capable) or the Lightning Web interface (crmhelp.veeva.com).

Concept of Deep Linking: Deep linking broadly refers to the ability to use a URL or command to navigate a user directly to specific content inside an app, rather than just launching the app's home screen (arxiv.org) (arxiv.org). On the Web, every page has a permalink URL, but mobile and enterprise apps often lack such granular links. For example, without deep links, a user might have to manually navigate through menus to reach a particular account or schedule. Academic research observes that **"the lack of direct links between apps has largely compromised the efficiency of task completion"** (arxiv.org). By contrast, deep links create "tunnels" between interface elements, cutting clicks and saving time. In practice, this means a dashboard displaying data (e.g. an HCP's sales metrics) can include a button that takes the rep straight to that HCP's CRM detail page or scheduling screen (crmhelp.veeva.com) (developer.veevacrm.com).

Within MyInsights, deep linking is implemented via the Veeva JavaScript API (the `com.veeva.clm` library). MyInsights JavaScript content can call special methods on the global `ds` (DataService) object to navigate to other CRM screens or perform actions. This transforms a passive data dashboard into an interactive launchpad for CRM functions (developer.veevacrm.com) (crmhelp.veeva.com). The remainder of this section outlines the historical evolution of MyInsights and deep linking, the current architecture, and how MyInsights leverages web tech to stay in sync with underlying CRM data.

Evolution of MyInsights and Deep Linking

Veeva unveiled Veeva CRM MyInsights in May 2017, describing it as a cloud-based data visualization capability. Early versions focused on out-of-the-box dashboards; gradually, they added more flexibility (e.g. custom HTML reports) and data integrations (Veeva Nitro, Veeva Link) through 2019–2023. Notably, in the

2023 CRM 23R3.0 release, Veeva **added explicit deep linking APIs**: the `ds.viewSection()` method and enhancements to `ds.viewRecord()` (crmhelp.veeva.com). Simultaneously, MyInsights Studio (the codeless editor) gained "View Section" and improved "View Record" actions to let admins set up linking without JavaScript (crmhelp.veeva.com).

Veeva is moving its CRM customers to its Vault CRM platform. There, MyInsights-like functionality continues under the name **X-Pages**. X-Pages is included in the standard Vault CRM license at no additional cost and is positioned as "space for experience, extensibility, cross-platform" integration. The deep linking patterns (e.g. calling `viewSection`, `newRecord`) are preserved conceptually in X-Pages, although implementation details (e.g. Salesforce LWC vs direct embedding) differ (crmhelp.veeva.com).

Architecture Overview

A high-level view of MyInsights architecture helps understand where deep linking fits: each dashboard is an entry in the CRM data model (an `HTML_Report_vod` record) with a record type indicating its context (Account, Call, Territory, etc.) . When a user opens that entry point (e.g. an Account page that has an attached MyInsights report), the CRM loads the HTML content. In the Veeva iPad app, the content runs locally (offline support via the device's synced data store); in Lightning (browser), the content is delivered via a CDN and shown in a secure iframe or Lightning Web Component container . Inside that sandbox, the page loads `MyInsightsLibrary.js` (Veeva's JS API) and the amplify Q promise library (developer.veevacrm.com). Using the `ds` object, the page can query Veeva objects (calls, accounts, orders) and present charts or tables.

Critically, `com.veeva.clm` provides methods beyond data queries: it includes navigation functions for deep linking. These APIs abstract away differences between mobile and desktop: e.g. if the rep is offline on iPad, `ds.viewRecord()` will show the record from the local DB; if online in Lightning, it opens Salesforce's record detail. The developer need not change code for platform differences, but must declare supported platforms on the `HTML_Report_vod` (e.g. enabling "Large Mobile Devices" vs "Lightning") . Security is enforced by normal CRM sharing: if the user lacks permission to a field or object, a deep link to that object may simply yield an empty page or error. For deep linking to work, the user also must have CRUD permission on the target object (as noted in Veeva's permissions documentation) (crmhelp.veeva.com) (crmhelp.veeva.com).

Integration points: In addition to the core Veeva CRM data model, MyInsights can pull from Veeva Nitro (cloud warehouse) or external Link data. Deep linking generally applies to CRM objects only (calls, accounts, plans, etc.) – linking into Nitro/Link data would typically be done by clicking on a chart to show CRM context. MyInsights Studio (the GUI dashboard builder) uses similar principles under the hood (crmhelp.veeva.com), but in this report we focus on custom-coded alerts using the JS API.

Deep Linking in MyInsights: Features and Patterns

This section details the specific deep linking capabilities of MyInsights, describing the tasks supported, the corresponding JavaScript API methods, and how they are used in practice. We distinguish between *link targets* (records or screens the user is taken to) and *link actions* (the code methods that perform the navigation).

Supported Deep Linking Tasks

According to the Veeva MyInsights documentation, the platform supports a range of CRM actions directly from a dashboard. Table 1 below summarizes the key methods and use-cases. Each method requires constructing a `configObject` that identifies the target object(s) and record IDs. In general:

- **Creating new records (ds.newRecord):** Launch the CRM "new record" interface for objects like Call (`Call2_vod__c`), Order (`Order_vod__c`), Inventory Monitoring, Medical Insight, Medical Inquiry, or Approved Email. The `fields` in the config specify default field values (e.g. link the new call to a specific Account) (developer.veevacrm.com) (crmhelp.veeva.com).
- **Viewing existing records (ds.viewRecord):** Open an existing record's detail page. Supported objects include Account, Order, Inventory Monitoring, Account Plan, Plan Tactics, Call, Action Items, Call Objective, Medical Event, Survey Target, Medical Insight, Medical Inquiry, Approved Email, SWOTs and Goals, and even custom objects (crmhelp.veeva.com) (developer.veevacrm.com). The `fields` must include the `Id` (or external/Studio ID) of the target record. An optional `target` property can specify which MyInsights tab to open if the destination is itself an account or territory dashboard (developer.veevacrm.com) (developer.veevacrm.com). For example, one might call `ds.viewRecord({object:"Account", fields:{Id:"<AccountID>"}})` to jump from any MyInsights page to that Account's CRM page.
- **Navigating within the same record (ds.viewSection):** (New in CRM 23R3.0) Switch between sub-tabs or sections on the current account or territory without leaving the page (crmhelp.veeva.com) (developer.veevacrm.com). This is useful for grouping related insights: e.g. one tab could show account sales, another call history. Calling `ds.viewSection({Id:"<TabRecordID>"})` jumps to that tab's content (the argument identifies the target MyInsights tab by its ID, External_Id_vod__c or Studio_Id_vod__c field, or is a non-empty list of such objects) (crmhelp.veeva.com) (developer.veevacrm.com). (MyInsights Studio provides equivalent "View Section" actions in its configuration UI.)
- **Scheduling calls (ds.sendToMySchedule):** Open the rep's calendar (My Schedule) pre-populated with a set of accounts to schedule visits. The config takes a list of account IDs: `ds.sendToMySchedule({accountIds: ["<AcctID1>","<AcctID2>"]})` (developer.veevacrm.com). This enables workflows like selecting top HCPs in a MyInsights list and launching the schedule interface directly with those accounts (crmhelp.veeva.com) (developer.veevacrm.com).
- **Executing suggestions (ds.executeSuggestionAction):** Perform CLM suggestion actions (complete, dismiss, execute) from MyInsights. The method takes a suggestion record ID and an action type: `"complete"`, `"dismiss"` or `"execute"` (the default when left blank). This allows a rep to act on suggested content (like pending samples or mails) directly from a MyInsights page without leaving it. (Note: In some contexts like a call report, suggestion execution may be disabled (crmhelp.veeva.com).)
- **Launching CLM media (ds.launchMediaForAccount):** Open a CLM presentation or media library. For example, `ds.launchMediaForAccount(accountId, presentationId, keyMessageMediaFileName)` opens the CLM presentation whose `Presentation_Id_vod` is given for that account, at the key message whose `Media_File_Name_vod` is given; the presentation ID and key message file name must be supplied together (developer.veevacrm.com). If both are left blank, it opens the CLM library instead.

Other actions mentioned include linking to CLM media library for presentations or leveraging “Suggestions” workflows (crmhelp.veeva.com). In summary, almost any CRM action that involves a record lookup or creation can be deep-linked via these APIs, as long as the user has permission and the entry point allows.

Table 1: Deep Linking Methods and Use-Cases in MyInsights

Action	MyInsights Method (JavaScript)	Description / Example
Create new Call (in CRM)	`ds.newRecord({object:"Call2_vod__c", fields:{Account_vod__c:"<AccountID>"}})`	Opens the *New Call* screen for given Account. (Sets default Account) (developer.veevacrm.com).
Create new Order	`ds.newRecord({object:"Order_vod__c", fields:{Account_vod__c:"<AccountID>"}})`	Opens *New Order* screen for given Account (crmhelp.veeva.com).
Create new Medical Inquiry	`ds.newRecord({object:"Medical_Inquiry_vod__c", fields: {Account_vod__c:"<Acct>"}})`	Opens *New Medical Inquiry* form (crmhelp.veeva.com).
View Account detail page	`ds.viewRecord({object:"Account", fields:{Id:"<AccountID>"}})`	Navigates to CRM *Account* page for given ID (crmhelp.veeva.com).
View Approved Email record	`ds.viewRecord({object:"Sent_Email_vod", fields:{Id:"<EmailID>"}})`	Opens the sent email record detail (if allowed) (crmhelp.veeva.com).
View Order or Inventory record	`ds.viewRecord({object:"Order_vod", fields:{Id:"<OrderID>"}})`	Opens *Order* record or inventory item (crmhelp.veeva.com).
Navigate to specific tab/section	`ds.viewSection({Id:"<TabID>"})`	On Account page, jump to the specified MyInsights tab within that account (crmhelp.veeva.com) (developer.veevacrm.com).
Open "My Schedule" for accounts	`ds.sendToMySchedule({accountIds: ["<Acct1>","<Acct2>"]})`	Opens scheduling UI with listed accounts ready to add to calendar (developer.veevacrm.com). Used, for example, to plan call visits (crmhelp.veeva.com).
Execute suggestion action	`ds.executeSuggestionAction(suggestionId, actionType)`	Runs a suggestion (Dismiss, Mark Complete, Execute) for CLM Suggestion record (crmhelp.veeva.com).
Launch CLM presentation	`ds.launchMediaForAccount(accountId, presentationId, keyMsgName)`	Opens CLM presentation for the account; can jump to a specific key message (developer.veevacrm.com).

Note: Methods like `ds.newRecord` and `ds.viewRecord` return Promises and take callback functions (as shown in [28] and [30]) to handle success or errors. All CRM field names in the `fields` object must use the exact API name (case-sensitive) and string values must be in single quotes within any query string (developer.veevacrm.com) (developer.veevacrm.com). Also, certain entry points (e.g. MyInsights on a Call Report) may disable some actions (see the *Not Supported* list in documentation) (crmhelp.veeva.com). To summarize, deep linking in MyInsights is implemented by wiring `ds.` methods to UI elements (buttons, links, or chart click events) in the custom HTML content, thereby bridging dashboards to the CRM workflow.

Enabling Deep Linking

Before using deep linking, a few prerequisites must be met. First, **user permissions**: the rep or admin must have appropriate CRUD rights on all target objects and fields involved (crmhelp.veeva.com). For example, to create a Medical Inquiry from MyInsights, the user needs all Medical Inquiry object permissions. Similarly, to use suggestions or call planning, the relevant Setup (CLM, suggestions) and object permissions must be granted. Administrators control MyInsights entry and object permissions just like other CRM features.

Second, **entry point support**: not every MyInsights tab supports every function. The official guide explicitly notes that deep linking is *not* supported on certain entry points (Call, Inventory Monitoring, Order Management) and in the special context of a Call Report (crmhelp.veeva.com) (crmhelp.veeva.com). Developers should design which MyInsights dashboards run in which record type. For instance, a call scheduling tool should be placed under a territory or account Insights tab, not on a Call Report, so navigation calls like `sendToMySchedule` will work.

Third, the **MyInsights v2 Lightning Web Component** must be used in Salesforce Lightning for browser-based content if deep linking from a MyInsights page to a record is needed (crmhelp.veeva.com). This component ensures the correct navigation context. Content deployed through older "Classic" iFrame approaches might skip this component and thus deep linking may not function. Veeva's release notes emphasize using the new LWC for any deep linked content to work in Lightning (crmhelp.veeva.com).

Once these conditions are satisfied, a developer can enable deep linking simply by invoking the above `ds.` methods. In MyInsights Studio (no-code), there are analogous navigation actions ("View Section", "View Record", "Schedule Calls," etc.) that generate similar config JSON under the hood, but that is covered in Veeva's Studio documentation (crmhelp.veeva.com). This report focuses on the code approach.

Deep Linking Code Patterns

We now explore **code patterns** for implementing deep links in MyInsights. These patterns show how developers instrument MyInsights dashboards to make them actionable. We provide examples and references to illustrate recommended usage of the APIs. Each pattern assumes you have included `MyInsightsLibrary.js` and `Q.js` (promises) in your `index.html` (developer.veevacrm.com)

Pattern: Opening New Records (`newRecord`)

The `ds.newRecord(configObject)` method is used to launch a CRM "New [Object]" form. The `configObject` must specify at least the `object` (API name of the object) and optionally initial field values. **Example:** To create a new Call with its Account pre-filled, one might use:

```
<button onclick="createCall()"> New Call </button>
<script>
 function createCall() {
 var cfg = {
 object: "Call2_vod__c",
 fields: {
 Account_vod__c: "00161000004vuXMAAY"
```

```
 }
 };
 ds.newRecord(cfg).then(function(resp) {
 console.log("New Call launched", resp);
 }).catch(function(err) {
 console.error(err);
 });
 }

</script>
```

This snippet (adapted from Veeva documentation (developer.veevacrm.com)) shows a button invoking `newRecord`. When clicked, it opens the native New Call editor within CRM, linked to the specified Account. The returned `resp` object can contain details (e.g. that the record was successfully created). Similar patterns apply to Orders, Medical Inquiries, etc. Note that only certain fields are supported for Approved Email creation via `newRecord` (as noted in the doc) (developer.veevacrm.com).

Pattern: Viewing Existing Records (`viewRecord`)

To navigate to an existing CRM record, use `ds.viewRecord(configObject)` with the `Id` of the target.
Example: On a dashboard listing recent calls, clicking one could open that call's detail page:

```
function viewCall(callId) {
 var cfg = {
 object: "Call2_vod__c",
 fields: {
 Id: callId
 }
 };
 ds.viewRecord(cfg).then(function(resp) {
 console.log("Navigated to Call", resp);
 }).catch(function(err) {
 console.error(err);
 });
}
```

The MyInsights library will switch context to the standard Call detail screen for that record (developer.veevacrm.com). You could wire this function to the datapoint click on a chart or an HTML link. The configuration may also include a `target` list to specify exactly which MyInsights tab group to open if landing on a multi-tab record (see Table 2 in docs) (developer.veevacrm.com). If the app supports multiple MyInsights subtabs (e.g. multiple reports on the Account page), this `target` can disambiguate which one to show.

Pattern: Switching Sections (`viewSection`)

When content is to remain on the same record but present different dashboards, `ds.viewSection()` is used. For example, a Territory-level dashboard may have "Team Sales" and "Team Goals" tabs. A button on one tab can link to the other group tab by ID:

```
function switchTab(sectionId) {
 ds.viewSection({
 Id: sectionId
 });
}
```

This was introduced to MyInsights in late 2023 (crmhelp.veeva.com). Content loaded with LWC will interpret this call and redraw the page with the `sectionId`. It requires that the target section (tab record) already exists in CRM. Deep linking docs note it's only supported at account or territory level (not at call-level) (developer.veevacrm.com).

Pattern: Sending to Schedule (`sendToMySchedule`)

To implement scheduling of calls, use `ds.sendToMySchedule({accountIds: [...]})`. For example:

```
function scheduleVisits(accountIds) {
 ds.sendToMySchedule({
 accountIds: accountIds
 });
}
```

Putting this behind a "Schedule Calls" button allows the user to select several accounts (perhaps via a checkbox grid in MyInsights) and then open the MySchedule interface with those accounts pre-loaded (developer.veevacrm.com) (crmhelp.veeva.com). The doc example simply shows calling `ds.sendToMySchedule({accountIds: ["ID1","ID2"]});`. It returns a promise that resolves with a success code. The method is supported on the Territory, Account and Account Plan entry points and does not work with Lightning for My Schedule (developer.veevacrm.com).

Pattern: Executing Suggestions (`executeSuggestionAction`)

If your MyInsights includes data from the CLM Suggestions object, you can act on them via

```
ds.executeSuggestionAction(suggestionId, actionType).then(function(resp) {
 // handle success (suggestion completed/dismissed)
});
```

The documented `actionType` values are `"complete"`, `"dismiss"` and `"execute"` (the default when left blank). This method lets a user complete or dismiss a suggestion (like "send an email" suggestion) directly from the dashboard. The documentation snippet shows calling `ds.executeSuggestionAction(...)` in a promise chain (developer.veevacrm.com).

Pattern: Launching CLM Media (`launchMediaForAccount`)

For dashboards that trigger CLM (e.g. a "Play Presentation" button), use:

```
ds.launchMediaForAccount(accountId, presentationId, keyMessageName);
```

This opens the CLM presentation identified by `presentationId` (the field `Presentation_Id_vod__c`) for the given account ID. The third argument is the key message's `Media_File_Name_vod` value, which opens the presentation at that key message; the presentation ID and key message file name must be used together (both populated or both blank) (developer.veevacrm.com). The example in the lib docs shows two uses: one providing both presentation and key message names, and another providing just the account (which opens the media library) (developer.veevacrm.com). In deep linking, this is considered an "action" linking CRM content to CLM content.

Each of these patterns must be embedded in a suitable UI element (button, link, chart event) in the MyInsights HTML. For instance, one might use `addEventListener("click", viewCall)` on a list item. Veeva recommends making the action obvious (e.g. "View Details" link) to avoid confusion (developer.veevacrm.com). Methods such as `newRecord`, `viewRecord`, `sendToMySchedule` and `executeSuggestionAction` return promises, so the typical pattern is `ds.method(cfg).then(success).catch(error)`. Developers should include error handling to catch situations where permissions or IDs are invalid.

Development Tips and Edge Cases

- **Single Quotes in Queries:** Although deep linking methods do not use SOQL `WHERE` clauses, querying data in MyInsights does. The developer guide emphasizes that *"IDs and field values in queries must be enclosed in single quotation marks*" (developer.veevacrm.com). This is a general MyInsights quirk (not deep link-specific), but relevant if you combine queries and links.
- **Case Sensitivity:** All object and field API names in code are case-sensitive in MyInsights (unlike desktop CRM which is case-insensitive) (developer.veevacrm.com). Ensure you use exactly the Salesforce API names with correct casing in your JavaScript.
- **Actionable Elements:** It is considered best practice (and documented in the library guide) to attach navigation methods to interactive elements, so users expect a navigation to occur. E.g., wrap the method call in a `button.onclick` handler rather than auto-invoking in onload (developer.veevacrm.com).
- **Unsupported Contexts:** The CRM Help lists certain limitations. For example, a MyInsights report displayed on a *Call Report entry* cannot use deep links (`newRecord`, `viewRecord`, etc. are disabled) (crmhelp.veeva.com). Similarly, if the entry point is "Order Management," deep linking is not supported there. Developers should plan which CRM page the dashboard is attached to.
- **Lightning Container:** In Salesforce Lightning (browser), content must be delivered via the `myInsightsV2` LWC for navigation to work (crmhelp.veeva.com). If the JavaScript is packaged as a Lightning Web Component, deep linking calls will function; if it is placed incorrectly (e.g. inside a custom Visualforce tab), navigation may break. Admins often set the LWC at the entry point record so content syncs from Core.

Example Workflow: Pre-Call Planning with Deep Links

To illustrate a real-world pattern, consider a *Pre-Call Planning* dashboard. Such a page might list prioritized HCP accounts based on data (e.g. sales, interaction history). The developer can add “Schedule Call” and “View Account” buttons next to each HCP. The code backing those buttons would be:

```
// given an account ID
function onViewAccountClick(accountId) {
 ds.viewRecord({
 object: "Account",
 fields: {
 Id: accountId
 }
 })
 .then(function() {
 /* success */
 })
 .catch(function(e) {
 console.error(e);
 });
}

function onScheduleClick(accountIds) {
 ds.sendToMySchedule({
 accountIds: accountIds
 });
}
```

When the rep clicks **View Account**, they are directly taken to that account’s CRM page (maybe a page showing contact info and call objectives). When they click **Schedule Call**, the MyInsights interface shifts to the CRM’s schedule view with the chosen accounts ready. This flow is exemplified by Veeva’s own help: *“Sarah Jones selects HCPs, then navigates to My Schedule with her selected HCPs ready to schedule”* (crmhelp.veeva.com). In that scenario, the rep needed only to drag accounts from the MyInsights list onto her calendar, with no extra data entry.

Data Analysis and Evidence

This section examines empirical and anecdotal evidence on the impact and usage of deep linking in mobile and CRM contexts. We present relevant studies, statistics, and expert commentary to support the value and effectiveness of deep linking features like those in MyInsights.

User Efficiency and Retention Metrics

While specific metrics for MyInsights deep linking are not publicly available, broader research underlines its general benefits. For instance, an independent analysis noted that *“mobile apps are usually isolated… the lack of direct links between apps has largely compromised the efficiency of task completion”* (arxiv.org). Their user

study suggested that providing “tunnels” (deep links) between app screens could dramatically reduce navigation steps. This aligns with Veeva’s own customer reports: pharmaceutical reps are spending less time hunting for data when dashboards link them straight to relevant CRM pages.

Marketing and app-analytics commentary claims large gains from deep linking. A MoldStud overview, for example, asserts that deep links improve retention, engagement and conversions (moldstud.com) (moldstud.com), but it does not identify primary studies for the uplift percentages that circulate in such articles, so they are not repeated here. The broader point, that guiding users directly to the content they need tends to increase usage and goal completion, is consistent with the academic research above. In CRM usage, this could translate to higher feature adoption and fewer “lost” reps who fail to find key screens.

Adoption Rates in Industry

Despite benefits, surveys show many apps have underutilized deep links. A large-scale analysis of 20,000 Android apps found that *“deep links do not get wide adoption among current Android apps”* (arxiv.org). AppSight data cited in a developer talk put the share of top Android apps with deep links at only 41% (speakerdeck.com). Reasons include complexity: earlier academic work notes that adding deep links “requires a lot of manual effort by app developers” (arxiv.org). The MyInsights dev approach mitigates this by providing built-in methods; still, each use-case must be scripted and tested.

Among Veeva customers, usage is growing. Veeva’s own customer stories and case studies show increasing reliance on embedded analytics as part of CRM. For example, GSK, a Veeva CRM customer, reported the effect of a MyInsights engagement-planning page: *“field teams reported saving 30 minutes per day in their engagement planning alone”* (www.veeva.com). GSK attributes this to bringing data from several places in Veeva CRM into one view. An IntuitionLabs overview of MyInsights notes: *“pharma firms increasingly treat embedded analytics [like MyInsights] as part of CRM strategy”*. In effect, the deep linking code patterns make these embedded dashboards actionable, so ROI is realized not just by information clarity, but by enabling concrete tasks (scheduling, editing records) directly in context.

Case Example: GSK Field Reps

The GSK blog (Veeva’s customer) provides a concrete example: before MyInsights, a rep might jump between multiple CRM pages to prepare for a call. With MyInsights, GSK reports *“a single, cohesive view brings together data from several different places in Veeva CRM”* (www.veeva.com). On average, field teams reported saving **30 minutes per day** in engagement planning (www.veeva.com). GSK does not attribute the saving to deep linking. The post also describes HCP–HCO affiliation management, which takes several steps through related lists in standard Veeva CRM, as a capability GSK was still building into a MyInsights screen at the time of writing (www.veeva.com). The planned design would let a rep change an affiliation with a button click instead of several page visits. This is consistent with the academic point that reducing navigation steps (via links) cuts wasted time (arxiv.org) (www.veeva.com).

Veeva’s own help documentation illustrates a scheduling scenario: from MyInsights, a rep selects several high-value HCPs and goes straight to My Schedule with them ready to schedule (crmhelp.veeva.com) (developer.veevacrm.com). This kind of flow avoids re-entering accounts and switching context. While large-

scale quantitative data on MyInsights adoption is proprietary, these examples and the references above suggest that well-implemented deep linking in MyInsights can improve field efficiency.

Case Studies and Use-Cases

In practical deployments, MyInsights dashboards and their deep links are tailored to specific business workflows. This section highlights common scenarios and *real-world* uses, illustrating how code patterns fit into enterprise cases.

Pre-Call and Call-Report Dashboards

A classic use of MyInsights is the **Account Interaction Summary** (account 360° view). Such a dashboard shows recent call dates, objectives, key messages used, samples given, etc., for one HCP account. In this context, deep links frequently include “View Record” (to open the account or last call detail) and “Schedule Call” (sending account to schedule). For example, a custom Interaction Summary page may list the last call (with a link): clicking it calls `ds.viewRecord` on that call, letting the rep quickly review call notes. Similarly, a “Plan Visit” button might use `ds.sendToMySchedule` with that account. Veeva’s out-of-the-box Interaction Summary page is an example. Code-wise, one might loop through selected HCPs and invoke:

```
selectedAccountIds.forEach(id => {
 ds.sendToMySchedule({
 accountIds: [id]
 });
});
```

though typically you batch and send once. The linked CRM Interface then opens beside the data, enabling quick decision-making.

Territory and Home Dashboards

At the territory or home level, dashboards often show aggregated performance (calls, sales, email usage) across all accounts in the territory. Deep linking here might involve opening territory plan records (`ds.viewRecord` on a Territory__c object) or composing a territory email. Also, linking to subtabs (using `ds.viewSection`) can let a rep switch between e.g. “Quarter Results” and “YTD Goal” tabs on the same MyInsights page. Table 1 of [11] (IntuitionLabs) lists “Territory & Rep Activity” dashboards as common, leveraging call counts and email metrics. In such dashboards, code to open an account or order details could be embedded in each chart hover or list item.

Key Opinion Leader (KOL) Analytics

Medical science liaisons often use KOL profile dashboards. These pages might display KOL background, publications, events attended, etc. Here, deep linking might include launching a CLM presentation relevant to that KOL (via `launchMediaForAccount`) or drilling into a list of relevant accounts. For example, a linked bar

chart showing “Publications in last year” could use `ds.viewRecord({object:"Publication_vod", fields:{Id:"..."}})` if there’s a custom object storing publications. Or a button “Email KOL” might use `ds.newRecord({object:"Medical_Inquiry_vod", fields:{Account_vod__c:"<HCPID>"}})` to send an inquiry.

Orders and Samples

MyInsights can also embed order management data. A Product Sales dashboard might display charts of product usage over time. Deep links there could open the underlying Order or Inventory Monitoring records. For example, clicking a bar for “Product X sales” might call `ds.queryRecord` then `ds.viewRecord` on the specific Order line. Sample drop dashboards (showing materials given) could link to the sample distribution records. If integrated with the CLM library, one could even launch a related promotional content.

Manager/Coaching Dashboards

Not all MyInsights are HCP-facing. Companies sometimes create dashboards for managers (e.g. individual rep performance). These might use the **User** object or custom rep metrics. Deep linking examples in that context include opening a team call log, or executing a “competitor call” suggestion. For instance, if a KPI chart shows a number of templates executed, clicking could open the associated CLM Hits. The patterns remain the same: use `ds.viewRecord` or `ds.executeSuggestionAction` against relevant internal objects.

Table 2: Example Use-Cases of MyInsights Dashboards and Links

Dashboard Type	Key Data Shown	Deep Link Actions
Account Interaction Summary	Last call date, key messages, sample drops, pending tasks.	*View Call Details* (`ds.viewRecord` on Call2_vod__c); *Schedule Next Call* (`ds.sendToMySchedule`); *Open Account Page* (`ds.viewRecord` on Account_vod).
Territory Insights	Territory call/email totals, sales vs target, etc..	*Drill-into Account Details* (`ds.viewRecord`); *Switch Tabs* (`ds.viewSection` to alternate data views); *Open Territory Plan* (`ds.viewRecord` on Territory__c record).
KOL Profile	Influencer’s specialty, call history, publications (via Veeva Link).	*Email KOL (`ds.newRecord` on Medical_Inquiry_vod); *View HCO Affiliations (`ds.viewRecord` on Account_HCO_Join_vod); *Launch KOL Presentation* (`ds.launchMediaForAccount`).
Product Sales Dashboard	Sales trend chart, order shapes, top products.	*Open Order History* (`ds.viewRecord` on Order_vod); *Query more sales data* (`ds.queryRecord`); *Go to Inventory records* (`ds.viewRecord` on InventoryMonitoring_vod).
Manager/Coaching Views	Rep call quality, call objectives met, onboarding tasks.	*View Rep Profile* (`ds.viewRecord` on User); *Open Performance Plan* (`ds.viewRecord` on Plan_Tactics_vod); *Execute Suggestion Action* (`ds.executeSuggestionAction`).

Sources: Common use-cases drawn from Veeva documentation and consulting examples, and Veeva-provided dashboard templates. Specific deep link methods are matched to the target object in each scenario.

Implications and Future Directions

Implementing deep linking in MyInsights is not just a technical feature; it reflects broader trends in CRM usability and data-driven workflows. We discuss some implications, limitations, and future trajectories below.

Implications for User Workflow

- **Unified Workflow:** Deep links erase the boundary between analytics and action. Users no longer have to memorize navigation: a visual insight can immediately become an action item. This increases feature adoption (users actually use scheduling and orders modules) and data accuracy (less manual bookkeeping).
- **Admin and Developer Productivity:** For system administrators, having dashboards that link directly into CRM means fewer training calls and support tickets ("I can't find that screen"). However, it also means admins must carefully manage permissions, since a missing permission on one field could silently break a link. Developers also need to test on all target platforms (iOS, Android, Lightning) to ensure links resolve correctly; mismatches (e.g. wrong API version) can lead to dead ends.
- **Cross-Platform Considerations:** By design, Veeva's `ds.` methods abstract away platform differences, but some nuances remain. For instance, on the iPad app an offline rep's deep link might still work with local data, but if the record wasn't synced, it may fail. In the Lightning desktop, CSP (Content Security Policy) and Salesforce's LockerService may impose restrictions (though `ds.newRecord` and friends are whitelisted by Veeva's component). Future moves to Vault CRM (X-Pages) promise to unify some differences (X-Pages content will use Vault's embedded browser framework), but also introduce new ones (the content will run in a Vault domain, not Salesforce). Developers should stay aware of platform docs (crmhelp.veeva.com) (crmhelp.veeva.com).

Performance and Data Considerations

- **Asynchronous Calls:** All `ds.` methods are asynchronous. In dashboards that make many calls (especially charts with dozens of points, each clickable), it is important to manage concurrency. For example, avoid firing 100 `viewRecord` calls at once. Instead, code should fetch any needed data in bulk (via `queryRecord`) and limit navigation calls to user interactions only.
- **Data Volume:** Deep linking fundamentally simplifies workflows, but the underlying queries can still be expensive. If, for instance, a user can call a function on an app object with thousands of records (like "show details for all accounts where volume > 100"), the initial query might load many records. Veeva suggests precomputing large datasets (via Nitro) or filtering aggressively (e.g. by current Account Id) . In practice, deep linking calls do not transfer data themselves (they only navigate), but the pages they open might query large datasets. Developers should design dashboards so that the launched screens remain performant (e.g. an account detail page with only relevant related lists).
- **Analytics:** A potential metric to track is how often deep links are used within MyInsights dashboards. Veeva does not currently provide built-in analytics on MyInsights click usage, but an organization could log usage by instrumenting JavaScript or capturing events within My Insights Studio. Understanding which links are clicked most (or never) could inform dashboard redesign.

Future Directions

- **X-Pages and Beyond:** As Veeva shifts to Vault CRM and X-Pages, MyInsights content will transition to X-Pages pages. The same deep linking concepts apply, but on Vault: methods like `viewRecord` and `viewSection` are supported in X-Pages JavaScript library (Vault JS API has analogous methods) (crmhelp.veeva.com). More importantly, X-Pages is inclusive of this capability with no extra license cost, potentially accelerating adoption.
- **AI and Intelligent Linking:** Veeva has discussed adding AI-powered analytics. One can imagine "smart" deep links that suggest actions based on AI insights. For example, an AI might highlight a particular account and present a "Plan Visit" deep link pseudo-button. From a coding standpoint, this could involve programmatically calling `sendToMySchedule` with an AI-chosen set. Future research might explore how contextual AI can drive which deep links to enable.
- **Cross-application Links:** Currently MyInsights deep links stay within Veeva CRM/CLM. In an integrated digital ecosystem (e.g. connecting CRM with marketing platforms), the concept could expand to cross-application deep links. For instance, a dashboard could include a link to open a marketing campaign in Veeva's other Clouds (or a web portal). Implementing this requires either custom handlers or Veeva integrations (like Action Links, or integrating with Events). While beyond native MyInsights, it aligns with the deeper concept of bridging data silos.
- **Standardization and Learning Curve:** The learning curve for MyInsights JavaScript development is steep but improving. Veeva's documentation and training resources continue to expand. Community sharing of code snippets and patterns can reduce "reinventing the wheel." We anticipate see more published examples of deep linking patterns, particularly as more companies develop unique MyInsights pages.

Conclusion

Deep linking code patterns in MyInsights represent a powerful convergence of data visualization and operational CRM workflows. By using Veeva's JavaScript APIs like `newRecord`, `viewRecord`, `viewSection`, and others, developers can create dashboards that not only inform users but also propel them directly into action. This report has detailed how deep linking works within the MyInsights framework – from the underlying architecture of embedded HTML reports to specific code examples of link invocations. Extensive source material (from Veeva's official docs to academic research) supports the assertion that deep linking reduces friction and boosts efficiency (arxiv.org) (www.veeva.com). Case studies like GSK's 30-minute time savings vividly illustrate real-world impact.

Nevertheless, challenges remain. Developers must account for platform contexts, user permissions, and performance when implementing deep links. Historical data shows adoption was initially slow in many apps (arxiv.org), but Veeva's evolving feature set (Lightning components, X-Pages) is closing the gap. Looking ahead, as analytics and AI grow, deep linking in MyInsights will likely expand in scope – potentially enabling even more interactive and intelligent CRM experiences.

In summary, mastering deep linking patterns in MyInsights is essential for organizations seeking to empower their field teams with seamless, data-driven decision capabilities. By following the guidelines and examples presented here, developers can ensure their dashboards not only *show* insights but also *do* something useful, creating a truly integrated CRM workflow. All claims and recommendations in this report are grounded in authoritative sources, from Veeva's own documentation (crmhelp.veeva.com) (crmhelp.veeva.com) to independent analyses of deep linking's effects (moldstud.com) (arxiv.org), ensuring a well-substantiated and balanced perspective.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://intuitionlabs.ai/articles/veeva-myinsights-x-pages-migration-guide
2. https://crmhelp.veeva.com/doc/Content/CRM_topics/MyInsights/MyInsightsDefaultFunct/DeepLinking.htm
3. https://arxiv.org/abs/1706.08274
4. https://arxiv.org/abs/1605.06928
5. https://arxiv.org/abs/1711.11564
6. https://www.veeva.com/blog/how-gsks-field-reps-save-30-minutes-a-day-with-veeva-crm-myinsights/
7. https://developer.veevacrm.com/doc/Content/CRM_topics/Veeva/myinsights-veeva.htm
8. https://crmhelp.veeva.com/doc/Content/CRM_topics/ReleaseNotes/23R3.0/NewIn23R3.0.htm
9. https://developer.veevacrm.com/doc/Content/CRM_topics/Veeva/myinsights.htm
10. https://intuitionlabs.ai/articles/veeva-myinsights-custom-dashboard-builder-tool-for-sales-analytics
11. https://intuitionlabs.ai/articles/veeva-myinsights-x-pages-development
12. https://moldstud.com/articles/p-the-role-of-deep-linking-in-mobile-app-engagement
13. https://speakerdeck.com/zasadnyy/deep-linking-myths-debunked

THE PUBLISHER

About IntuitionLabs

IntuitionLabs is an AI consulting, custom software development and data engineering firm serving pharmaceutical, biotechnology, medical-device and other life-science organizations. We work with clinical, regulatory, medical-affairs, commercial, quality and IT teams to connect technology decisions with the work people need to accomplish.

AI consulting and adoption

Our AI enablement services cover readiness assessments, use-case selection, governance and policies, team workshops, adoption measurement and ongoing advisory support. We help organizations structure the information layer behind AI: source material, context, permissions and maintained knowledge that make generated answers useful and reviewable. Private LLM inference and hosted AI options support teams evaluating how to operate AI with appropriate control over their data and infrastructure.

Software, data and life-science workflows

IntuitionLabs develops custom software for pharma and biotech, integrates enterprise systems, and builds data engineering and business intelligence solutions. Areas of focus include AI agents, regulatory research, medical writing, medical affairs, CMC information, competitive intelligence and clinical-document workflows. Our eTMF intelligence work includes cross-system reconciliation and inspection-readiness support.

Enterprise platforms and regulated delivery

We provide Veeva services, application support, managed services, integrations and custom applications, alongside enterprise content work involving platforms such as Egnyte. For regulated workflows, our services include GxP enablement, computer-system validation and software development addressing 21 CFR Part 11 requirements. The applicable controls, validation responsibilities and acceptance criteria are defined for each engagement.

Work with IntuitionLabs

Explore AI enablement, pharma and biotech software development, data engineering and BI, and Veeva services. Contact IntuitionLabs to discuss your workflow, information sources and implementation needs.

IntuitionLabs publishes educational research to help life-science teams make informed technology decisions. Coverage of a product or organization does not imply a client relationship, endorsement or partnership.

Website: https://intuitionlabs.ai

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.