

# ▶ 日本語プレゼンテーション原稿

## Employee Management System Demo

For Senior Engineers (12-13分)

### 📝 Speech Structure / スピーチ構成

Total Time / 合計時間: 12-13分

- 自己紹介: 1.5分
- システムアーキテクチャ: 2.5分
- ライブデモ: 6分
- 技術的な説明: 2分
- Q&A: 1分

### 📖 三つのフォーマット / Three Formats

このドキュメントには、同じ内容が3つの形式で書かれています：

- ローマ字 (Romaji) - 読みやすい
- ひらがな (Hiragana) - 日本語学習者向け
- 漢字+ひらがな (Kanji with Hiragana) - 標準的な日本語

各セクションは短く、簡単な文で書かれています。N5レベルの方がN3レベルに聞こえるように作りました。

### 🎬 SECTION 1: 自己紹介 (Introduction - 1.5分)

#### ローマ字版 (Romaji Version)

Ohayou gozaimasu, mina san. Watashi no name wa [Your Name] desu. Watashi wa ima, InterSystems IRIS to

React.js no intaan desu.

Kyou wa, watashi ga tsukutta kantan na shain kanri shisutemu wo setsumei shimasu. Kono shisutemu wa, modaan na gijutsu wo tsukatta furu sutakku uebbu apurikeeshon desu. Furonto endo wa React.js to TypeScript wo tsukatte imasu. Bakkuendo wa InterSystems IRIS to ObjectScript wo tsukatte imasu.

[Pause, randingu peeji wo miseru]

Kono purezentēshon no mokuteki wa:

- CRUD sokushin no kanzen na waaku furō
- React to IRIS no tōgō
- REST API no jissō
- Modaan na furonto endo kaihatsu no purakutisu

wo shōkai suru koto desu.

Saisho kara saigo made, yūzā touroku kara shain kanri made, subete no nagare wo setsumei shimasu. Gijutsuteki na shitsumon ga areba, itsudemo kiite kudasai.

Soredewa, shisutemu ākitekucha kara hajimemasu.

---

## ひらがな版 (Hiragana Version)

おはようございます、みなさん。わたしのなまえは[Your Name]です。わたしはいま、InterSystems IRIS と React.js のインターンです。

きょうは、わたしがつくったかんたんなしゃいんかんりシステムをせつめいします。このシステムは、モダンなぎじゅつをつかったフルスタックウェブアプリケーションです。フロントエンドは React.js と TypeScript をつかっています。バックエンドは InterSystems IRIS と ObjectScript をつかっています。

[ポーズ、ランディングページをみせる]

このプレゼンテーションのもくてきは:

- CRUD そくしんのかんぜんなワークフロー
- React と IRIS のとうごう
- REST API のじっそう
- モダンなフロントエンドかいはつのプラクティス

をしようかいすることです。

さいしょからさいごまで、ユーザーとうろくからしゃいんかんりまで、すべてのながれをせつめいします。ぎじゅつてきなしつもんがあれば、いつでもきいてください。

それでは、システムアーキテクチャからはじめます。

---

## 漢字+ひらがな版 (Kanji with Hiragana Version)

おはようございます、皆(みな)さん。私(わたし)の名前(なまえ)は[Your Name]です。私(わたし)は今(いま)、InterSystems IRIS と React.js のインターナンスです。

今日(きょう)は、私(わたし)が作(つく)った簡単(かんたん)な社員管理(しゃいんかんり)システムを説明(せつめい)します。このシステムは、モダンな技術(ぎじゅつ)を使(つか)ったフルスタックウェブアプリケーションです。フロントエンドは React.js と TypeScript を使(つか)っています。バックエンドは InterSystems IRIS と ObjectScript を使(つか)っています。

[ポーズ、ランディングページを見(み)せる]

このプレゼンテーションの目的(もくてき)は:

- CRUD 操作(そうさ)の完全(かんぜん)なワークフロー
- React と IRIS の統合(とうごう)
- REST API の実装(じっそう)
- モダンなフロントエンド開発(かいはつ)のプラクティス

を紹介(しょうかい)することです。

最初(さいしょ)から最後(さいご)まで、ユーザー登録(とうろく)から社員管理(しゃいんかんり)まで、全(すべ)ての流(なが)れを説明(せつめい)します。技術的(ぎじゅつてき)な質問(しつもん)があれば、いつでも聞(き)いてください。

それでは、システムアーキテクチャから始(はじ)めます。

---

## E SECTION 2: システムアーキテクチャ (System Architecture - 2.5分)

---

### Part 2.1: 全体像 (High-Level Overview - 1分)

#### ローマ字版 (Romaji Version)

[Akitekucha zu wo miseru]

Kono apurikeeshon wa san-sō ākitekucha wo tsukatte imasu.

### Mazu, furonto endo sō:

- React 18 to TypeScript de tsukurarete imasu
- Material-UI wo tsukatte, konpōnento wo sutairu shimasu
- Vite wo birudo tsūru to shite tsukaimasu
- Axios de HTTP tsūshin wo okonaimasu

### Tsugi ni, API sō:

- RESTful API dezain desu
- Hyōjun-teki na HTTP mesoddo wo tsukaimasu
- JSON dēta fōmatto de tsūshin shimasu
- Nana-tsu no endpoint ga arimasu - ninshō to CRUD sōsa yō desu

### Saigo ni, bakkuento sō:

- InterSystems IRIS wo dētabēsu enjin to shite tsukaimasu
- ObjectScript de bijinesu rojikku wo kakimasu
- Futatsu no omona tēburu ga arimasu - tblAccount to tblEmployee desu

Furonto endo wa pōto 5173 de ugokimasu. Vite no purokishi kinou wo tsukatte, IRIS bakkuento to tsūshin shimasu. IRIS wa pōto 52773 desu. Kore wa kaihatsu-chū no CORS mondaι wo sakemasu.

---

### ひらがな版 (Hiragana Version)

[アーキテクチャをみせる]

このアプリケーションはさんそアーキテクチャをつかっています。

### まず、フロントエンドそ:

- React 18 と TypeScript でつくられています
- Material-UI をつかって、コンポーネントをスタイルします
- Vite をビルドツールとしてつかいます
- Axios で HTTP つうしんをおこないます

### つぎに、API そ:

- RESTful API デザインです
- ひょうじゅんてきな HTTP メソッドをつかいます
- JSON データフォーマットでつうしんします
- ななつのエンドポイントがあります - にんしょうと CRUD そうさようです

## さいごに、バックエンドそう:

- InterSystems IRIS をデータベースエンジンとしてつかいます
- ObjectScript でビジネスロジックをかきます
- ふたつのおもなテーブルがあります - tblAccount と tblEmployee です

フロントエンドはポート 5173 でうごきます。Vite のプロキシきのうをつかって、IRIS バックエンドとつうしんします。IRIS はポート 52773 です。これはかいはつちゅうの CORS もんだいをさけます。

---

## 漢字+ひらがな版 (Kanji with Hiragana Version)

[アーキテクチャ図(ず)を見(み)せる]

このアプリケーションは三層(さんそう)アーキテクチャを使(つか)っています。

## まず、フロントエンド層(そう):

- React 18 と TypeScript で作(つく)られています
- Material-UI を使(つか)って、コンポーネントをスタイルします
- Vite をビルドツールとして使(つか)います
- Axios で HTTP 通信(つうしん)を行(おこな)います

## 次(つぎ)に、API 層(そう):

- RESTful API デザインです
- 標準的(ひょうじゅんてき)な HTTP メソッドを使(つか)います
- JSON データフォーマットで通信(つうしん)します
- 七(なな)つのエンドポイントがあります - 認証(にんしょう)と CRUD 操作用(そうさよう)です

## 最後(さいご)に、バックエンド層(そう):

- InterSystems IRIS をデータベースエンジンとして使(つか)います
- ObjectScript でビジネスロジックを書(か)きます

- 二(ふた)つの主(おも)なテーブルがあります - `tblAccount` と `tblEmployee` です

フロントエンドはポート 5173 で動(うご)きます。Vite のプロキシ機能(きのう)を使(つか)って、IRIS バックエンドと通信(つうしん)します。IRIS はポート 52773 です。これは開発中(かいはつちゅう)の CORS 問題(もんだい)を避(さ)けます。

---

## Part 2.2: 技術詳細 (Technical Stack Details - 1.5分)

### ローマ字版 (Romaji Version)

#### Furonto endo gjutsu:

React wa konpōnento bēsu no ākitekucha ga arimasu. Kore wa UI wo mojuraa-teki ni shimasu. Watashitachi wa kansu konpōnento to fukku dake wo tsukaimasu. Omoni useState, useEffect, soshite useMemo wo tsukaimasu.

useState wa sutēto kanri yō desu. useEffect wa saido efekuto - tatoeba API kōru - yō desu. useMemo wa pafoomansu saitekika yō desu - firuta sōsa nado.

TypeScript wa taipu anzen-sei wo tsuikashimasu. Konpairu-ji ni erā wo mitsukemasu. Tatoeba, watashitachi no Employee intafeisu wa, shain obujekuto ga dono puropati wo motsu beki ka wo sadamemasu.

Material-UI wa, Google no Material Design wo shitagau puribiruto konpōnento wo teikyō shimasu. Kore wa konshistento de puropesshonaru na yūzā intafeisu wo kakuho shimasu.

#### Bakkuendo gjutsu:

Bakkuendo de wa, InterSystems IRIS wo tsukaimasu. IRIS wa maruchimoderu dētabēsu desu. Obujekuto pāshisutansu wo sapōto shimasu.

Watashitachi no REST API wa IRIS no %CSP.REST furemuwāku wo tsukatte tsukurarete imasu. API handorā kurasu wa kono furemuwāku wo kakuchō shimasu. URL rūto wa XML konfigyurēshon de teigi shimasu.

Jūyōna kino no hitotsu wa, ZCONVERT wo tsukatta UTF-8 henkan desu. Kore wa nihongo no moji wo tadashiku atsukaeru tame ni hitsuyō desu.

*[Demo e no iko]*

Dewa, kore kara, kono gjutsu ga dō renkei suru ka wo jissai ni o miseshimasu.

---

### ひらがな版 (Hiragana Version)

#### フロントエンドぎじゅつ:

React はコンポーネントベースのアーキテクチャがあります。これは UI をモジュラーにします。わたしたちはかんすうコンポーネントとフックだけをつかいます。おもに useState、useEffect、そして useMemo をつかいます。

useState はステートかんりようです。useEffect はサイドエフェクト - たとえば API コール - ようです。useMemo はパフォーマンスさいてきかようです - フィルターそうさなど。

TypeScript はタイプあんぜんせいをつかいます。コンパイルじにエラーをみつけます。たとえば、わたしたちの Employee インターフェイスは、しゃいんオブジェクトがどのプロパティをもつべきかをさだめます。

Material-UI は、Google の Material Design をしたがうプリビルトコンポーネントをていきょうします。これはコンシスティントでプロフェッショナルなユーザーインターフェイスをかくほします。

### バックエンドぎじゅつ:

バックエンドでは、InterSystems IRIS をつかいます。IRIS はマルチモデルデータベースです。オブジェクトパーシスタンスをサポートします。

わたしたちの REST API は IRIS の %CSP.REST フレームワークをつかってつくられています。API ハンドラークラスはこのフレームワークをかくちようします。URL ルートは XML コンフィギュレーションでていきします。

じゅうよな機能(きのう)のひとつは、ZCONVERT をつかった UTF-8 へんかんです。これはにほんごのもじをただしくあつかえるために必要(ひつよう)です。

### [デモへのいこう]

では、これから、このぎじゅつがどうれんけいするかをじっさいにおみせします。

---

## 漢字+ひらがな版 (Kanji with Hiragana Version)

### フロントエンド技術(ぎじゅつ):

React はコンポーネントベースのアーキテクチャがあります。これは UI をモジュラー的(てき)にします。私(わたし)たちは関数(かんすう)コンポーネントとフックだけを使(つか)います。主(おも)に useState、useEffect、そして useMemo を使(つか)います。

useState はステート管理用(かんりよう)です。useEffect はサイドエフェクト - 例(たと)えば API コール - 用(よう)です。useMemo はパフォーマンス最適化用(さいてきかよう)です - フィルター操作(そうさ)など。

TypeScript はタイプ安全性(あんぜんせい)を追加(ついか)します。コンパイル時(じ)にエラーを見(み)つけます。例(たと)えば、私(わたし)たちは Employee インターフェイスは、社員(しゃいん)オブジェクトがどのプロパティを持(も)つべきかを定(さだ)めます。

Material-UI は、Google の Material Design に従(したが)うプリビルトコンポーネントを提供(ていきょう)します。これはコンシスティントでプロフェッショナルなユーザーインターフェイスを確保(かくほ)します。

### バックエンド技術(ぎじゅつ):

バックエンドでは、InterSystems IRIS を使(つか)います。IRIS はマルチモデルデータベースです。オブジェクトパーシスタンスをサポートします。

私(わたし)たちの REST API は IRIS の %CSP.REST フレームワークを使(つか)って作(つく)られています。API ハンドラークラスはこのフレームワークを拡張(かくちよう)します。URL ルートは XML コンフィギュレーションで定義(ていぎ)します。

重要(じゅうよう)な機能(きのう)の一(ひと)つは、ZCONVERT を使(つか)った UTF-8 変換(へんかん)です。これは日本語(にほんご)の文字(もじ)を正(ただ)しく扱(あつか)えるために必要(ひつよう)です。

[デモへの移行(いこう)]

では、これから、この技術(ぎじゅつ)がどう連携(れんけい)するかを実際(じっさい)にお見(み)せします。

---

---

## SECTION 3: ライブデモ (Live Demo - 6分)

---

### Part 3.1: ユーザー登録 (User Signup - 1分)

#### ローマ字版 (Romaji Version)

[Sainappu peeji ni idō suru]

Mazu, atarashii yūzā akaunto wo tsukurimasu. Kore ga sainappu peeji desu.

[Fōmu ni nyūryoku shi nagara hanasu]

Tesuto akaunto wo iremasu:

- Namae: Tanaka Tarō
- Mēru adoresu: [tanaka.taro@example.com](mailto:tanaka.taro@example.com)
- Pasuwādo: password123

[Taipingu shi nagara baridēshon ni tsuite hanasu]

Furonto endo wa baridēshon wo okonaimasu. Tatoeba, mēru ni @ māku ga nakereba, shisutemu wa sabumitto wo yurusanai desu. Pasuwādo wa saikō hachi moji hitsuyō desu.

[Touroku botan wo kurikku]

Touroku wo kurikku suru to, furonto endo wa POST rikuesuto wo /sem/signup ni okurimasu. Bakkuedo wa kono dēta wo futatabi kensho shimasu. Watashitachi wa kuraiento-gawa no baridēshon dake ni wa tanominai desu.

Bakkuedo wa tblAccount tēburu ni atarashii rekōdo wo tsukurimasu.

[Seikō to ridairekuto wo miseru]

Touroku ga seikō shita ato, shisutemu wa sain'in peeji ni ridairekuto shimasu.

---

## ひらがな版 (Hiragana Version)

[サインアップページにいどうする]

まず、あたらしいユーザーアカウントをつくります。これがサインアップページです。

[フォームにニュウリヨクしながらはなす]

テストアカウントをいれます:

- なまえ: たなかたろう
- メールアドレス: tanaka.taro@example.com
- パスワード: password123

[タイピングしながらバリデーションについてはなす]

フロントエンドはバリデーションをおこないます。たとえば、メールに @ マークがなければ、システムはサブミットをゆるさないです。パスワードはさいてい8もじひつようです。

[どうろくボタンをクリック]

どうろくをクリックすると、フロントエンドは POST リクエストを /sem/signup におくります。バックエンドはこのデータをふたたびけんしょうします。わたしたちはクライアントがわのバリデーションだけにはたのみないです。

バックエンドは tblAccount テーブルにあたらしいレコードをつくります。

[せいこうとリダイレクトをみせる]

どうろくがせいこうしたあと、システムはサインインページにリダイレクトします。

---

## 漢字+ひらがな版 (Kanji with Hiragana Version)

[サインアップページに移動(いどう)する]

まず、新(あたら)しいユーザー アカウントを作(つく)ります。これがサインアップページです。

[フォームに入力(にゅうりょく)しながら話(はな)す]

テストアカウントを入(い)れます:

- 名前(なまえ): 田中太郎(たなかたろう)
- メールアドレス: [tanaka.taro@example.com](mailto:tanaka.taro@example.com)
- パスワード: password123

[タイピングしながらバリデーションについて話(はな)す]

フロントエンドはバリデーションを行(おこな)います。例(たと)えば、メールに @ マークがなければ、システムはサブミットを許(ゆる)さないです。パスワードは最低(さいてい)8文字(もじ)必要(ひつ)よう)です。

[登録(とうろく)ボタンをクリック]

登録(とうろく)をクリックすると、フロントエンドは POST リクエストを /sem/signup に送(おく)ります。バックエンドはこのデータを再度(ふたたび)検証(けんしょう)します。私(わたし)たちはクライアント側(がわ)のバリデーションだけには頼(たよ)りないです。

バックエンドは tblAccount テーブルに新(あたら)しいレコードを作(つく)ります。

[成功(せいこう)とリダイレクトを見(み)せる]

登録(とうろく)が成功(せいこう)した後(あと)、システムはサインインページにリダイレクトします。

---

## Part 3.2: ログイン (User Login - 40秒)

ローマ字版 (Romaji Version)

[Sain'in fōmu ni nyūryoku]

Ima tsukutta akaunto de roguin shimasu.

[Roguin wo sabumitto]

Shisutemu wa POST rikuesuto wo /sem/signin ni okurimasu. Bakkuedo wa dētabēsu ni kueri wo jikkō shite, mēru ga sonzai suru ka, soshite pasuwādo ga icchi suru ka wo kakunin shimasu.

[Roguin seikō go]

Ninshō ga seikō suru to, shisutemu wa sesshon furagu wo localStorage ni hozon shimasu. Soshite shain risuto peeji ni ridairekuto shimasu. Nabigēshon bā ni rogauto botan ga arawaremasu. Kore wa Layout konpōnento desu.

---

## ひらがな版 (Hiragana Version)

[サインインフォームにニュウリョク]

いまつくったアカウントでログインします。

[ログインをサブミット]

システムは POST リクエストを /sem/signin におくります。バックエンドはデータベースにクエリをじっこうして、メールがそんざいするか、そしてパスワードがいちつするかをかくにんします。

[ログインせいこうご]

にんしょうがせいこうすると、システムはセッションフラグを localStorage にほぞんします。そしてしゃいんリストページにリダイレクトします。ナビゲーションバーにログアウトボタンがあらわれます。これは Layout コンポーネントです。

---

## 漢字+ひらがな版 (Kanji with Hiragana Version)

[サインインフォームに入力(にゅうりょく)]

今(いま)作(つく)ったアカウントでログインします。

[ログインをサブミット]

システムは POST リクエストを /sem/signin に送(おく)ります。バックエンドはデータベースにクエリを実行(じっこう)して、メールが存在(そんざい)するか、そしてパスワードが一致(いちつ)するかを確認(かくにん)します。

[ログイン成功後(せいこうご)]

認証(にんしょう)が成功(せいこう)すると、システムはセッションフラグを localStorage に保存(ほぞん)します。そして社員(しゃいん)リストページにリダイレクトします。ナビゲーションバーにログアウトボタンが現(あらわ)れます。これは Layout コンポーネントです。

---

## Part 3.3: 新しい社員を追加 (Add New Employee - 1.5分)

### ローマ字版 (Romaji Version)

[*Shinki touroku botan wo kurikku*]

Atarashii shain wo tsuikashimasu. Kono botan wa /employees/new ni nabigeeto shimasu.

[*Fōmu ni nyūryoku hajimeru*]

Yamada Hanako to iu shain wo tsukurimasu:

- Shain bangō: 12345 - choudo go keta hitsuyō desu
- Namae: Yamada Hanako
- Kana namae: Yamada Hanako
- Seibetsu: Josei wo erabimasu
- Yūbin bangō: 100-0001
- Jūsho: Tokyo-to Chiyoda-ku Chiyoda 1-1
- Denwa: 090-1234-5678
- Busho: Eigyōbu
- Taishoku furagu: chekku shinai - zaishoku-chū

*[Nyūryoku shi nagara setsumei]*

Fōmu wa kontorooru konpōnento wo tsukaimasu. Kakku inputto no atai wa, useState fukku wo tsukatte, sutēto hensu ni tsunagaremasu.

*[Touroku botan wo kurikku]*

Sabumitto mae ni, shisutemu wa kakunin daiarogu wo hyōji shimasu.

*[Daiarogu de "Hai" wo kurikku]*

Furonto endo wa shain dēta to issho ni POST rikuesuto wo okurimasu. Bakkuedo de wa:

1. IRIS REST handorā ga JSON pērōdo wo ukemasu
2. Hitsuyōna firudo wo kensho shimasu
3. Jūfuku shain bangō wo chekku shimasu
4. Nihongo tekisuto wo ZCONVERT de UTF-8 ni henkan shimasu
5. Atarashii tblEmployee obujekuto wo tsukurimasu
6. Puropati wo setto shimasu - deleteFlg=0 to genzai no taimusutanpu
7. %Save() wo yonde, obujekuto wo dētabēsu ni hozon shimasu
8. Seikō respōnsu wo kaeshimasu

*[Seikō to ridairekuto wo miseru]*

Risuto ni ridairekuto saremasu. Atarashii shain ga tēburu ni arawaremasu.

---

## ひらがな版 (Hiragana Version)

### [新規登録ボタンをクリック]

あたらしいしゃいんをついついします。このボタンは /employees/new にナビゲートします。

### [フォームにニュウリョクはじめる]

やまだはなこというしゃいんをつくります:

- しゃいんばんごう: 12345 - ちょうど5けたひとつようです
- なまえ: やまだはなこ
- かななまえ: ヤマダハナコ
- せいべつ: じょせいをえらびます
- ゆうびんばんごう: 100-0001
- じゅうしょ: とうきょうとちよだくちよだ1-1
- でんわ: 090-1234-5678
- ぶしょ: えいぎょうぶ
- たいしょくフラグ: チェックしない - ざいしょくちゅう

### [ニュウリョクしながらせつめい]

フォームはコントロールコンポーネントをつかいます。かくインプットのあたいは、useState フックをつかって、ステートへんすうにつながれます。

### [新規登録ボタンをクリック]

サブミットまえに、システムはかくにんダイアログをひょうじします。

### [ダイアログで「はい」をクリック]

フロントエンドはしゃいんデータといっしょに POST リクエストをおくります。バックエンドでは:

1. IRIS REST ハンドラーが JSON ペーロードをうけます
2. ひとつようなフィールドをけんしょうします
3. じゅうふくしゃいんばんごうをチェックします
4. にほんごテキストを ZCONVERT で UTF-8 にへんかんします
5. あたらしい tblEmployee オブジェクトをつくります
6. プロパティをセットします - deleteFlg=0 とげんざいのタイムスタンプ
7. %Save() をよんで、オブジェクトをデータベースにほぞんします
8. せいこうレスポンスをかえします

[せいこうとリダイレクトをみせる]

リストにリダイレクトされます。あたらしいしゃいんがテーブルにあらわれます。

---

## 漢字+ひらがな版 (Kanji with Hiragana Version)

[新規登録(しんきとうろく)ボタンをクリック]

新(あたら)しい社員(しゃいん)を追加(ついか)します。このボタンは /employees/new にナビゲートします。

[フォームに入力(にゅうりょく)始(はじ)める]

山田花子(やまだはなこ)という社員(しゃいん)を作(つく)ります:

- ・ 社員番号(しゃいんばんごう): 12345 - 丁度(ちょうど)5桁(けた)必要(ひつよう)です
- ・ 名前(なまえ): 山田花子(やまだはなこ)
- ・ カナ名前(なまえ): ヤマダハナコ
- ・ 性別(せいべつ): 女性(じょせい)を選(えら)びます
- ・ 郵便番号(ゆうびんばんごう): 100-0001
- ・ 住所(じゅうしょ): 東京都千代田区千代田(とうきょうとちよだくちよだ)1-1
- ・ 電話(でんわ): 090-1234-5678
- ・ 部署(ぶしょ): 営業部(えいぎょうぶ)
- ・ 退職(たいしょく)フラグ: チェックしない - 在職中(ざいしょくちゅう)

[入力(にゅうりょく)しながら説明(せつめい)]

フォームはコントロールコンポーネントを使(つか)います。各(かく)インプットの値(あたい)は、useState フックを使(つか)って、ステート変数(へんすう)に繋(つな)がれます。

[登録(とうろく)ボタンをクリック]

サブミット前(まえ)に、システムは確認(かくにん)ダイアログを表示(ひょうじ)します。

[ダイアログで「はい」をクリック]

フロントエンドは社員(しゃいん)データと一緒に(いっしょ)に POST リクエストを送(おく)ります。バックエンドでは:

1. IRIS REST ハンドラーが JSON ペーロードを受(う)けます
2. 必要(ひつよう)なフィールドを検証(けんしょう)します

- 重複(じゅうふく)社員番号(しゃいんばんごう)をチェックします
- 日本語(にほんご)テキストを ZCONVERT で UTF-8 に変換(へんかん)します
- 新(あたら)しい tblEmployee オブジェクトを作(つく)ります
- プロパティをセットします - deleteFlg=0 と現在(げんざい)のタイムスタンプ
- %Save() を呼(よ)んで、オブジェクトをデータベースに保存(ほぞん)します
- 成功(せいこう)レスポンスを返(かえ)します

[成功(せいこう)とリダイレクトを見(み)せる]

リストにリダイレクトされます。新(あたら)しい社員(しゃいん)がテーブルに現(あらわ)れます。

---

#### Part 3.4: 検索機能 (Search - 30秒)

##### ローマ字版 (Romaji Version)

[Kensaku bokkusu ni taipu]

Kensaku kinō wa kuraiento-gawa firuta wo shimeji shimasu. Kensaku bokkusu ni "Yamada" to taipu shimasu.

[Firuta ga okoru no wo miseru]

Tēburu ga sugu ni apdēto saremasu - API kōru wa arimasen. Kore wa, subete no shain wo ichido rōdo shite, soshite ローカル ni firutā suru kara desu. Kono pataan wa chiisai dētasetto ni wa yoku hatarakimasu.

Kensaku wa shain bangō, namee, to kana namee wo chekku shimasu.

[Kensaku wo kuria]

Kensaku wo kuria suru to, subete no shain ga futatabi arawaremasu.

---

##### ひらがな版 (Hiragana Version)

[けんさくボックスにタイプ]

けんさくきのうはクライアントがわフィルターをしめします。けんさくボックスに「やまだ」とタイプします。

[フィルターがおこるのをみせる]

テーブルがすぐにアップデートされます - API コールはありません。これは、すべてのしゃいんをいちどロードして、そしてローカルにフィルターするからです。このパターンはちいさいデータセットにはよくはたらきます。

けんさくはしゃいんばんごう、なまえ、とかななまえをチェックします。

[けんさくをクリア]

けんさくをクリアすると、すべてのしゃいんがふたたびあらわれます。

---

## 漢字+ひらがな版 (Kanji with Hiragana Version)

[検索(けんさく)ボックスにタイプ]

検索機能(けんさくきのう)はクライアント側(がわ)フィルターを示(しめ)します。検索(けんさく)ボックスに「山田(やまだ)」とタイプします。

[フィルターが起(おこ)るのを見(み)せる]

テーブルがすぐにアップデートされます - API コールはありません。これは、全(すべ)ての社員(しゃいん)を一度(いちど)ロードして、そしてローカルにフィルターするからです。このパターンは小(ちい)さいデータセットにはよく働(はたら)きます。

検索(けんさく)は社員番号(しゃいんばんごう)、名前(なまえ)、とカナ名前(なまえ)をチェックします。

[検索(けんさく)をクリア]

検索(けんさく)をクリアすると、全(すべ)ての社員(しゃいん)が再(ふたた)び現(あらわ)れます。

---

## Part 3.5: 社員編集 (Edit Employee - 1.5分)

### ローマ字版 (Romaji Version)

[Mō hitotsu shain wo tsuika - Shain bangō 67890, Satō Jirō]

Henshū kinō wo shimesu tame, mō hitotsu shain wo tsuikashimasu.

[Tsuika shita ato, henshū aikon wo kurikku]

Kono shain no henshū aikon wo kurikku shimasu. /employees/67890 ni nabigeeto shimasu - shain no dētabēsu ID ga URL no bubun ni narimasu.

[Purifiru fōmu wo miseru]

Konpōnento wa henshū mōdo da to ninshiki shimasu. URL paramētā ga "new" de wa nai kara desu. GET rikuesuto wo shite shain dēta wo totte, fōmu wo purifiru shimasu.

Jūyōna ten: Shain bangō firudo wa henshū-chū ni mukou ni sarete imasu. Sonzai suru shain no shain bangō wo henkou shitaku nai desu.

## [Henkou wo okonau]

Ikutsu ka henkou shimasu:

- Busho: Kaihatsbu kara Gijutsubu ni kawemasu
- Taishoku furagu: Chekku shimasu - taishoku-zumi to shimasu

## [Kōshin botan wo kurikku]

Shisutemu wa PUT rikuesuto wo /sem/employee/67890 ni okurimasu. PUT wo tsukaimasu - POST de wa arimasen. Kore wa REST kiyakku ni shitagaimasu.

## [Daiarogu de kakunin]

Kakunin go, bakkudo wa ObjectScript no %OpenId mesoddo wo tsukatte sonzai suru shain obujekuto wo rōdo shi, puropati wo henkou shi, soshite dētabēsu ni modoshimasu.

## [Seikō wo miseru]

Kōshin sareta shain wa ima "Gijutsubu" to hyōji shi, taishoku furagu ga setto sarete imasu.

---

## ひらがな版 (Hiragana Version)

### [もうひとつしゃいんをつかい - しゃいんばんごう 67890、さとうじろう]

へんしゅうきのうをしめすため、もうひとつしゃいんをつかいます。

### [つかいたあと、へんしゅうアイコンをクリック]

このしゃいんのへんしゅうアイコンをクリックします。/employees/67890 にナビゲートします - しゃいんのデータベース ID が URL のぶぶんになります。

### [プリフィルフォームをみせる]

コンポーネントはへんしゅうモードだとにんしきします。URL パラメーターが「new」ではないからです。GET リクエストをしてしゃいんデータをとって、フォームをプリフィルします。

じゅうようなてん: しゃいんばんごうフィールドはへんしゅうちゅうにむこうにされています。そんざいするしゃいんのしゃいんばんごうをへんこうしたくないです。

### [へんこうをおこなう]

いくつかへんこうします:

- ぶしょ: かいはつぶからぎじゅつぶにかわります
- たいしょくフラグ: チェックします - たいしょくずみとします

## [こうしんボタンをクリック]

システムは PUT リクエストを /sem/employee/67890 におくります。PUT をつかいます - POST ではありません。これは REST きやくにしたがいます。

## [ダイアログでかくにん]

かくにんご、バックエンドは ObjectScript の %OpenId メソッドをつかってそんざいするしゃいんオブジェクトをロードし、プロパティをへんこうし、そしてデータベースにもどします。

## [せいこうをみせる]

こうしんされたしゃいんはいま「ぎじゅつぶ」とひょうじし、たいしょくフラグがセットされています。

---

## 漢字+ひらがな版 (Kanji with Hiragana Version)

[もう一(ひと)つ社員(しゃいん)を追加(ついか) - 社員番号(しゃいんばんごう) 67890、佐藤次郎(さとうじろう)]

編集機能(へんしゅうきのう)を示(しめ)すため、もう一(ひと)つ社員(しゃいん)を追加(ついか)します。

## [追加(ついか)した後(あと)、編集(へんしゅう)アイコンをクリック]

この社員(しゃいん)の編集(へんしゅう)アイコンをクリックします。/employees/67890 にナビゲートします - 社員(しゃいん)のデータベース ID が URL の部分(ぶぶん)になります。

## [プリフィルフォームを見(み)せる]

コンポーネントは編集(へんしゅう)モードだと認識(にんしき)します。URL パラメーターが「new」ではないからです。GET リクエストをして社員(しゃいん)データを取(と)って、フォームをプリフィルします。

重要(じゅうよう)な点(てん): 社員番号(しゃいんばんごう)フィールドは編集中(へんしゅうちゅう)に無効(むこう)にされています。存在(そんざい)する社員(しゃいん)の社員番号(しゃいんばんごう)を変更(へんこう)したくないです。

## [変更(へんこう)を行(おこな)う]

いくつか変更(へんこう)します:

- 部署(ぶしょ): 開発部(かいはつぶ)から技術部(ぎじゅつぶ)に変(か)わります
- 退職(たいしょく)フラグ: チェックします - 退職済(たいしょくす)みとします

## [更新(こうしん)ボタンをクリック]

システムは PUT リクエストを /sem/employee/67890 に送(おく)ります。PUT を使(つか)います - POST ではありません。これは REST 規約(きやく)に従(したが)います。

[ダイアログで確認(かくにん)]

確認後(かくにんご)、バックエンドは ObjectScript の %OpenId メソッドを使(つか)って存在(そんざい)する社員(しゃいん)オブジェクトをロードし、プロパティを変更(へんこう)し、そしてデータベースに戻(もど)します。

[成功(せいこう)を見(み)せる]

更新(こうしん)された社員(しゃいん)は今(いま)「技術部(ぎじゅつぶ)」と表示(ひょうじ)し、退職(たいしょく)フラグがセットされています。

## Part 3.6: 社員削除 (Delete Employee - 40秒)

ローマ字版 (Romaji Version)

*[Henshū mōdo de, sakujo botan wo kurikku]*

Saigo ni, kono shain wo sakujo shimasu. Sakujo botan wa henshū mōdo dake de riyō dekimasu.

*[Kakunin mae ni setsumei]*

Kono shisutemu wa sofuto sakujo wo jisshi shite imasu. Rekōdo wa jissai ni wa dētabēsu kara torinozokaremasan. Kawari ni, deleteFlg wo 1 ni setto shimasu.

Kore ni wa ikutsu ka no riten ga arimasu:

- Dēta kaifuku ga kanō desu
- Kansa shōseki wo iji dekimasu
- Dēta no seikō-sei ga tamotaremasu

*[Sakujo wo kakunin]*

Bakkuendo wa deleteFlg wo 1 ni setto shi, shain wa risuto kara kieremasu. Shikashi, dētabēsu ni chokusetsu kueri wo sureba, rekōdo wa mada soko ni arimasu.

*[Kara no risuto matawa nokotta shain wo miseru]*

Risuto kueri wa "WHERE deleteFlg = 0" de firutā shimasu. Dakara sakujo sareta shain wa jidō-teki ni jogai saremasu.

## ひらがな版 (Hiragana Version)

[へんしゅうモードで、さくじょボタンをクリック]

さいごに、このしゃいんをさくじょします。さくじょボタンはへんしゅうモードだけりようできます。

[かくにんまえにせつめい]

このシステムはソフトさくじょをじっししています。レコードはじっさいにはデータベースからとりのぞかれません。かわりに、deleteFlg を 1 にセットします。

これにはいくつかのりてんがあります:

- データかいふくがかのうです
- かんさしょうせきをいじできます
- データのせいこうせいがたもたれます

[さくじょをかくにん]

バックエンドは deleteFlg を 1 にセットし、しゃいんはリストから見えます。しかし、データベースにちよくせつクエリをすれば、レコードはまだそこにあります。

[からのリストまたはのこったしゃいんをみせる]

リストクエリは「WHERE deleteFlg = 0」でフィルターします。だからさくじょされたしゃいんはじどうてきにじょがいされます。

---

## 漢字+ひらがな版 (Kanji with Hiragana Version)

[編集(へんしゅう)モードで、削除(さくじょ)ボタンをクリック]

最後(さいご)に、この社員(しゃいん)を削除(さくじょ)します。削除(さくじょ)ボタンは編集(へんしゅう)モードだけで利用(りよう)できます。

[確認前(かくにんまえ)に説明(せつめい)]

このシステムはソフト削除(さくじょ)を実施(じっし)しています。レコードは実際(じっさい)にはデータベースから取(と)り除(のぞ)かれません。代(か)わりに、deleteFlg を 1 にセットします。

これにはいくつかの利点(りてん)があります:

- データ回復(かいふく)が可能(かのう)です
- 監査証跡(かんさしょうせき)を維持(いじ)できます
- データの整合性(せいごうせい)が保(たも)たれます

### [削除(さくじょ)を確認(かくにん)]

バックエンドは deleteFlg を 1 にセットし、社員(しゃいん)はリストから消(き)えます。しかし、データベースに直接(ちょくせつ)クエリをすれば、レコードはまだそこにあります。

### [空(から)のリストまたは残(のこ)った社員(しゃいん)を見(み)せる]

リストクエリは「WHERE deleteFlg = 0」でフィルターします。だから削除(さくじょ)された社員(しゃいん)は自動的(じどうてき)に除外(じょがい)されます。

---

## Part 3.7: ログアウト (Logout - 20秒)

### ローマ字版 (Romaji Version)

*[Rogauto botan wo kurikku]*

Saigo ni, rogauto wo shimeji shimasu. Rogauto botan wo kurikku suru to, shisutemu wa localStorage wo kuria shimasu.

*[Sain'in peeji e no ridairekuto wo miseru]*

Sain'in peeji ni ridairekuto saremasu. Ima, moshi /employees ni chokusetsu nabigeeto shiyō to sureba, ProtectedRoute konpōnento ga kore wo torae, roguin ni ridairekuto shimasu.

---

### ひらがな版 (Hiragana Version)

*[ログアウトボタンをクリック]*

さいごに、ログアウトをしめします。ログアウトボタンをクリックすると、システムは localStorage をクリアします。

*[サインインページへのリダイレクトをみせる]*

サインインページにリダイレクトされます。いま、もし /employees にちょくせつナビゲートしようとすれば、ProtectedRoute コンポーネントがこれをとらえ、ログインにリダイレクトします。

---

### 漢字+ひらがな版 (Kanji with Hiragana Version)

*[ログアウトボタンをクリック]*

最後(さいご)に、ログアウトを示(しめ)します。ログアウトボタンをクリックすると、システムは localStorage をクリアします。

*[サインインページへのリダイレクトを見(み)せる]*

サインインページにリダイレクトされます。今(いま)、もし /employees に直接(ちょくせつ)ナビゲートしようとすれば、ProtectedRoute コンポーネントがこれを捉(とら)え、ログインにリダイレクトします。

---

---

## 🔧 SECTION 4: 技術的な説明 (Technical Discussion - 2分)

---

### Part 4.1: 重要な技術的決定 (Key Technical Decisions - 1分)

#### ローマ字版 (Romaji Version)

Ikutsu ka no jūyōna gjutsu-teki kettei ni tsuite kantan ni setsumei shimasu:

**React Fukku:** Watashitachi wa kansu konpōnento to fukku dake wo tsukaimasu. Kore wa modaan na React no pataan desu. useState wa konpōnento sutēto wo kanri shimasu. useEffect wa saido efekuto - tatoeba API kōru - wo atskai masu. useMemo wa kosuto no takai keisan wo saitekika shimasu.

**TypeScript:** TypeScript wa ōku no bagu wo rantaimu mae ni toraemasu. Tatoeba, moshi watashi ga sū wo kitai suru tokoro ni mojiretsu wo watasou to sureba, konpairā wa sugu ni oshiete kuremasu.

**Sofuto sakujo pataan:** Rekōdo wo butsuriteki ni sakujo suru kawari ni, furagu wo tsukaimasu. Kore wa purodakushon shisutemu de wa ippan-teki desu. Dēta kaifuku to kansa no tame ni.

**Kuraiento-gawa tai sābā-gawa sōsa:** Kensaku to firuta wa kuraiento-gawa de okonawaremasu - yori yoi UX no tame. Shikashi baridēshon to dēta hozon wa tsuneni sābā-gawa de okonawaremasu - sekyuriti no tame.

---

#### ひらがな版 (Hiragana Version)

いくつかのじゅうようなぎじゅつてきけっていについてかんたんにせつめいします:

**React フック:** わたしたちはかんすうコンポーネントとフックだけをつかいます。これはモダンな React のパターンです。useState はコンポーネントステートをかんりします。useEffect はサイドエフェクト - たとえば API コール - をあつかいます。useMemo はコストのたかいけいさんをさいてきかします。

**TypeScript:** TypeScript はおおくのバグをランタイムまえにとらえます。たとえば、もしわたしがすうをきたいするところにもじれつをわたそうとすれば、コンパイラーはすぐにおしえてくれます。

**ソフトさくじょパターン:** レコードをぶつりてきにさくじょするかわりに、フラグをつかいます。これはプロダクションシステムではいっぱいんてきです。データかいふくとかんさのために。

**クライアントがわたいサーバーがわそうさ:** けんさくとフィルターはクライアントがわでおこなわれます - よりよい UX のために。しかしバリデーションとデータほぞんはつねにサーバーがわでおこなわれます - セキュリティのために。

---

## 漢字+ひらがな版 (Kanji with Hiragana Version)

いくつかの重要(じゅうよう)な技術的決定(ぎじゅつてきけってい)について簡単(かんたん)に説明(せつめい)します:

**React フック:** 私(わたし)たちは関数(かんすう)コンポーネントとフックだけを使(つか)います。これはモダンな React のパターンです。useState はコンポーネントステートを管理(かんり)します。useEffect はサイドエフェクト - 例(たと)えば API コール - を扱(あつか)います。useMemo はコストの高(たか)い計算(けいさん)を最適化(さいてきか)します。

**TypeScript:** TypeScript は多(おお)くのバグをランタイム前(まえ)に捉(とら)えます。例(たと)えば、もし私(わたし)が数(すう)を期待(きたい)するところに文字列(もじれつ)を渡(わた)そうとすれば、コンパイラーはすぐに教(おし)えてくれます。

**ソフト削除(さくじょ)パターン:** レコードを物理的(ぶつりてき)に削除(さくじょ)する代(か)わりに、フラグを使(つか)います。これはプロダクションシステムでは一般的(いっぽんてき)です。データ回復(かいふく)と監査(かんさ)のために。

**クライアント側(がわ)対(たい)サーバー側操作(がわそうさ):** 検索(けんさく)とフィルターはクライアント側(がわ)で行(おこな)われます - より良(よ)い UX のために。しかしバリデーションとデータ保存(ほぞん)は常(つね)にサーバー側(がわ)で行(おこな)われます - セキュリティのために。

---

## Part 4.2: 改善点 (Areas for Improvement - 1分)

### ローマ字版 (Romaji Version)

Kore wa gakushū purojekuto na node, purodakushon ni wa ikutsu ka no kaizen-ten ga arimasu:

Sekyuriti:

- Pasuwādo wa bcrypt de hasshu suru beki desu
- JWT tōkun wo jissō suru beki desu
- Purodakushon de wa HTTPS ga hitsuyō desu
- Rēto seigen wo tsuika suru beki desu

Pafoomansu:

- Ōkina dētasetto ni wa pējinēshon wo tsuika

- Kyasshingu wo jissō - tatoeba Redis
- Dētabēsu kueri wo indekusū de saitekika

## Kino:

- Rōru bēsu no akusesu kontorōru
- Dēta ekusupōto kinō
- Yori yoi erā messēji

## Tesuto:

- Konpōnento no yunitto tesuto
  - API no integrēshon tesuto
  - Endo-tsuetu tesuto
- 

## ひらがな版 (Hiragana Version)

これはがくしゅうプロジェクトなので、プロダクションにはいくつかのかいぜんてんがあります:

## セキュリティ:

- パスワードは bcrypt でハッシュするべきです
- JWT トークンをじっそうするべきです
- プロダクションでは HTTPS がひつようです
- レートせいげんをついかするべきです

## パフォーマンス:

- おおきなデータセットにはページネーションをついか
- キャッシングをじっそう - たとえば Redis
- データベースクエリをインデックスでさいてきか

## きのう:

- ロールベースのアクセスコントロール
- データエクスポートきのう
- よりよいエラーメッセージ

## テスト:

- コンポーネントのユニットテスト
  - API のインテグレーションテスト
  - エンドツーエンドテスト
- 

## 漢字+ひらがな版 (Kanji with Hiragana Version)

これは学習(がくしゅう)プロジェクトなので、プロダクションにはいくつかの改善点(かいぜんてん)があります:

## セキュリティ:

- パスワードは bcrypt でハッシュするべきです
- JWT トークンを実装(じっそう)するべきです
- プロダクションでは HTTPS が必要(ひつよう)です
- レート制限(せいげん)を追加(ついか)するべきです

## パフォーマンス:

- 大(おお)きなデータセットにはページネーションを追加(ついか)
- キャッシングを実装(じっそう) - 例(たと)えば Redis
- データベースクエリをインデックスで最適化(さいてきか)

## 機能(きのう):

- ロールベースのアクセスコントロール
- データエクスポート機能(きのう)
- より良(よ)いエラーメッセージ

## テスト:

- コンポーネントのユニットテスト
  - API のインテグレーションテスト
  - エンドツーエンドテスト
-

## ？ SECTION 5: Q&A (1分)

---

### ローマ字版 (Romaji Version)

Go-seichō arigatō gozaimashita. Shitsumon ga areba, yorokonde o-kotae shimasu:

- Ākitekucha no kettei ni tsuite
- Jissō no shōsai ni tsuite
- React ya IRIS no tokutei no koto ni tsuite
- Demo no dono bubun demo

Yoroshiku onegai shimasu.

---

### ひらがな版 (Hiragana Version)

ごせいちようありがとうございました。しつもんがあれば、よろこんでおこたえします:

- アーキテクチャのけっていについて
- じっそうのしょうさいについて
- React や IRIS のとくていのことについて
- デモのどのぶぶんでも

よろしくおねがいします。

---

### 漢字+ひらがな版 (Kanji with Hiragana Version)

ご清聴(せいちよう)ありがとうございました。質問(しつもん)があれば、喜(よろこ)んでお答(こた)えします:

- アーキテクチャの決定(けってい)について
- 実装(じっそう)の詳細(しょうさい)について
- React や IRIS の特定(とくてい)のことについて
- デモのどの部分(ぶぶん)でも

よろしくお願(ねが)いします。

---

## 練習のヒント (Practice Tips)

### ローマ字で読む練習 (Romaji Reading Practice)

#### 1. 最初は全部ローマ字で読んでください

- 自然な速さで読む
- わからない単語があれば、漢字版を見る

#### 2. 発音に注意してください

- 長音 (ō, ū) を正しく発音する
- 促音 (っ) に注意する
- アクセントは平坦でOK

#### 3. 区切りで練習してください

- 最初は1文ずつ
- 慣れたら段落ごと
- 最後に全体を通して

### プレゼンテーションのコツ (Presentation Tips)

#### 声の出し方:

- **はっきりと発音:** 日本語の各音節を明確に
- **ゆっくり話す:** N3レベルでもゆっくりなら理解しやすい
- **間を取る:** 文と文の間に少し停止

#### 体の動き:

- **アイコンタクト:** 聴衆を見る
- **手のジェスチャー:** ポイントを強調
- **画面を指す:** デモ中は画面を指して説明

#### トラブル対処:

- **言葉に詰まつたら:** 英語で補足してOK
- **デモが失敗したら:** 落ち着いて、英語で説明

- 質問が理解できなかったら: "もう一度お願ひします" と言う

## 重要フレーズ集 (Key Phrases)

### 始めるとき:

- それでは、始めます (Soredewa, hajimemasu) - "Now, let's begin"
- ご覧ください (Goran kudasai) - "Please look"
- こちらをご覧ください (Kochira wo goran kudasai) - "Please look at this"

### デモ中:

- 今から～します (Ima kara ~ shimasu) - "I will now ~"
- ここで～します (Koko de ~ shimasu) - "Here, I will ~"
- 注目してください (Chūmoku shite kudasai) - "Please pay attention"

### 質問対応:

- よい質問ですね (Yoi shitsumon desu ne) - "That's a good question"
- 説明します (Setsumei shimasu) - "I'll explain"
- 英語でもよろしいですか (Eigo demo yoroshii desu ka) - "May I answer in English?"

## ⌚ 最終チェックリスト (Final Checklist)

### プレゼン前日:

- 原稿を3回通して読む
- ローマ字版で発音確認
- デモの流れを確認
- テストデータを準備

### プレゼン当日朝:

- もう一度原稿を読む
- 声を出して練習
- システムが動くか確認
- リラックスする

### プレゼン直前:

- 深呼吸する
  - 水を飲む
  - 笑顔で始める
  - 自信を持つ
- 

頑張ってください！あなたならできます！🚀💪

(Ganbatte kudasai! Anata nara dekimasu!) Good luck! You can do it! 🚀💪