

パーソナルデータ連携モジュール アプリケーション開発ガイド別紙

1.0版

2022年10月1日

日本電気株式会社

ドキュメント名	版	日付	
パーソナルデータ連携モジュール アプリケーション開発ガイド別紙	1	2022/10/1	

--	--

目次
<ul style="list-style-type: none">・ はじめに・ 4.1 ログインおよび各サービス使用方法・ 5.20 利用者作成バッチ・ 5.21 CToken連携バッチ・ 5.22 Region改定再同意依頼バッチ・ 5.23 Book自動閉鎖バッチ・ 5.24 Region利用自動終了対象追加バッチ・ 5.25 Region利用自動終了バッチ・ 5.26 Book削除バッチ・ 5.27 出力データファイル管理作成バッチ・ 5.28 利用者データ削除バッチ・ 5.29 Region終了対象追加バッチ・ 5.30 Region利用終了バッチ・ 5.31 Region利用者連携バッチ

ドキュメント名	版	日付	
パーソナルデータ連携モジュール アプリケーション開発ガイド別紙	1	2022/10/1	

はじめに

アプリケーション開発ガイド別紙は、アプリケーション開発ガイド内で説明が必要なシーケンス図を記載するためのドキュメントである。
対応するシーケンスに関しては元資料 アプリケーション開発ガイド を参照すること。

ドキュメント名	版	日付	
パーソナルデータ連携モジュール アプリケーション開発ガイド別紙	1	2022/10/1	

--	--

機能ID		関連要件ID
	<div>4.1 ログインおよび各PxR-Service使用方法</div> <div><pre>sequenceDiagram actor Operator participant OS as オペレーターサービス participant PBP as PxR-Block-Proxyサービス participant PS as PxR-Service Operator->>OS: 1 ログイン OS-->>Operator: 2 ログイン成功 Operator->>OS: 3 リクエスト OS->>PBP: 4 セッション確認 PBP-->>OS: 5 確認成功 OS->>PS: 6 リクエスト PS-->>OS: 7 レスポンス OS-->>Operator: 8 レスポンス</pre></div> <div><ol style="list-style-type: none">オペレーターサービス ログインAPIを呼び出して、ログインを実施します。ログインに成功してレスポンスを返却します。 発行されるセッションの有効期限は3時間です。APIにアクセスするたびにアクセス時刻から3時間後まで有効期限が延長されます。セッションの有効期限は、カタログで変更可能です。PxR-Block-Proxyサービス 各種メソッドAPIを呼び出して、2で取得したセッションID、呼び出し先のPxR-Service、APIを指定してリクエストを行います。オペレーターサービスを呼び出して、セッション確認を実施します。セッション確認に成功した結果を返却します。セッションの有効期限が延長されます。指定されたPxR-Service、APIを呼び出します。8. 呼び出したPxR-Service、APIのレスポンスを返却します。</div>	

ドキュメント名	版	日付	
パーソナルデータ連携モジュール アプリケーション開発ガイド別紙	1	2022/10/1	

--	--

機能ID		関連要件ID
	<div>5.20 利用者作成バッチ</div> <div><pre>sequenceDiagram actor B1 as バッチ participant S1 as APP/WF Book運用サービス B1->>S1: 1 利用者作成 (バッチ) S1-->>B1: 2 処理完了 actor B2 as バッチ participant S2 as APP/WF Book運用サービス B2->>S2: 1 利用者作成 (バッチ) S2-->>B2: 2 処理完了</pre></div> <div>処理内容</div> <div>1. Book運用サービス 利用者作成(バッチ)APIを呼び出す</div>	

ドキュメント名	版	日付	
パーソナルデータ連携モジュール アプリケーション開発ガイド別紙	1	2022/10/1	

--	--

ItemID		関連要件ID
	<div>5.21 CToken連携/バッチ</div> <div><pre>sequenceDiagram actor Batch participant Service as APP/WF Local-CTokenサービス Note over Batch, Service: 1 CToken台帳連携対象件数取得 Service-->>Batch: 2 CToken台帳連携対象件数 loop [CToken台帳連携対象件数/1000回分] Batch->>Service: 3 CToken台帳連携 Service-->>Batch: 4 処理完了 end actor Batch participant Service as APP/WF Local-CTokenサービス</pre></div> <div>処理内容</div> <div>1. Local-CTokenサービス CToken台帳連携対象件数APIを呼び出す</div> <div>2. 取得した対象件数を使用し、件数/1000 回分3～4をループさせる</div> <div>3. Local-CTokenサービス CToken台帳連携APIを呼び出す</div> <div>4. PxR-Block-Proxyサービス POST APIをRoot-BlockのCToken台帳サービス Local-CToken差分登録APIを指定して呼び出す</div>	

要件ID	関連要件ID
5.22 Region改定再同意依頼パッチ	
<div></div> <p>処理内容</p> <ol style="list-style-type: none">グローバル設定を取得して、Regionサービス運用の設定がONになっているかを確認するRegionサービス運用の設定がOFFの場合、以降の処理を行わないBook管理サービス.Region利用規約更新通知個人登録 API を呼び出し、Region利用規約更新通知個人管理に通知対象の個人を登録するBook管理サービス.Region利用規約更新通知個人取得 API にて通知対象の個人を10件ずつ取得する取得したレコードを元に通知サービス.登録APIで再同意通知メッセージを登録するBook管理サービス.Region利用規約更新通知最終送信日時登録 API を呼び出し、Region利用規約更新通知個人管理の最終送付日時を更新する	

ドキュメント名		版	日付	
パーソナルデータ連携モジュール アプリケーション開発ガイド別紙		1	2022/10/1	
要件ID		関連要件ID		
5.23 Book自動閉鎖パッチ				
<pre>sequenceDiagram actor Patch as パッチ participant BookService as Book管理サービス loop "【対象が取得できなくなるまで】" Patch->>BookService: 1PF利用規約未同意個人取得 BookService-->>Patch: 2自動終了対象個人(10件) end loop "【件数分】" Patch->>BookService: 3強制停止 BookService-->>Patch: 4処理完了 Patch->>BookService: 5My-Condition-Data出力コード取得 BookService-->>Patch: 6My-Condition-Data出力コード end alt "【My-Condition-Data出力コードのレコードが取得できない場合】" Patch->>BookService: 7データ出力準備 BookService-->>Patch: 8処理完了 end</pre>				
処理内容				
1. Book管理サービス.PF利用規約未同意個人取得 API を呼び出し、Bookの自動終了を行う対象の個人を10件ずつ取得する				
2. 取得したレコードを元にBook管理サービス.強制停止 API を呼び出す				
3. Book管理サービス.My-Condition-Data出力コード取得 API を呼び出し、以下の条件で My-Condition-Data出力コード のレコードを取得する bookId = 1.で取得したレコードの bookId type = 5: Book強制閉鎖				
4. 上記 3. にてレコードが取得できなかった場合、Book管理サービス.データ出力準備 API を呼び出す				

要件ID	関連要件ID
5.24 Region利用自動終了対象追加バッチ	
<div><pre>sequenceDiagram actor Batch as バッチ participant PXR as PxR-Root カタログサービス participant Book as Book管理サービス Batch->>PXR: 1公開カタログ取得 PXR-->>Batch: 2公開カタログ alt [Regionサービス運用の設定がOFF] Batch->>End: 処理終了 else [通知対象が取得できなくなるまで] Batch->>Book: 3Region利用規約未同意個人取得 Book-->>Batch: 4未同意個人データ(10件) loop [件数分] Batch->>Book: 5My-Condition-Data出力コード取得 Book-->>Batch: 6My-Condition-Data出力コード Batch->>Book: 7データ出力準備 Book-->>Batch: 8処理完了 end end</pre></div> <p>処理内容</p> <ol style="list-style-type: none">グローバル設定を取得して、Regionサービス運用の設定がONになっているかを確認するRegionサービス運用の設定がOFFの場合、以降の処理を行わないBook管理サービス.Region利用規約未同意個人取得 API を呼び出し、Region利用の自動終了を行う対象の個人を10件ずつ取得するBook管理サービス.My-Condition-Data出力コード取得 API を呼び出し、以下の条件で My-Condition-Data出力コード のレコードを取得する bookId = 1.で取得したレコードの bookId type = 3: Region利用自動終了 region = Region のカタログコード、カタログバージョン上記 3. にてレコードが取得できなかった場合、Book管理サービス.データ出力準備 API を呼び出す	

機能ID		関連要件ID
	<div>5.25 Region利用自動終了パッチ</div> <div><pre>sequenceDiagram actor Batch participant PxR as PxR-Root カタログサービス participant Book as Book管理サービス participant Ver as 本人性確認サービス Batch->>PxR: 1公開カタログ取得 PxR-->>Batch: 2公開カタログ alt [Regionサービス運用の設定がOFF] Batch->>Ver: 処理終了 else Batch->>Book: 3My-Condition-Data出力コード取得 Book-->>Batch: 4My-Condition-Data出力コード loop [件数分] Batch->>Book: 5My-Condition-Data出力コード更新 Book-->>Batch: 6処理完了 Batch->>Ver: 7My-Condition-Book一覧取得 API Ver-->>Batch: 8My-Condition-Book一覧 Batch->>Ver: 9確認済本人性確認コード生成 Ver-->>Batch: 10確認済本人性確認コード生成 Batch->>Book: 11利用者ID連携解除 Book-->>Batch: 12処理完了 Batch->>Book: 13My-Condition-Data出力コード更新 Book-->>Batch: 14処理完了 end end Batch->>PxR: 1公開カタログ取得 PxR-->>Batch: 2公開カタログ</pre></div> <div>処理内容</div> <div><ol style="list-style-type: none">グローバル設定を取得して、Regionサービス運用の設定がONになっているかを確認するRegionサービス運用の設定がOFFの場合、以降の処理を行わないBook管理サービス.My-Condition-Data出力コード取得 API を呼び出し、以下の条件で My-Condition-Data出力コード のレコードを取得する Region利用者連携解除承認ステータス = 承認不要:0 または 承認:2 Region利用者連携解除ステータス = 0: 未処理 処理中フラグ = 処理待ち:0上記 3. にてレコードが取得できた場合、<ol style="list-style-type: none">Book管理サービス.My-Condition-Data出力コード更新 API を呼び出し、処理中フラグを 処理中:1 に更新する 排他制御にて更新に失敗した場合、当該レコードの処理をスキップして次のレコードを処理する本人性確認サービス.確認済本人性確認コード生成 API を呼び出し、本人性確認コードを生成する POST /identity-verifycate/code/verifiedBook管理サービス.利用者ID連携解除 API を呼び出す上記処理がすべて正常に終了した場合、Book管理サービス.My-Condition-Data出力コード更新 API を呼び出し、以下の通り更新する Region利用者連携解除ステータス = 1: 処理済 処理中フラグ = 処理待ち:0上記処理でエラーが発生した場合、Book管理サービス.My-Condition-Data出力コード更新 API を呼び出し、以下の通り更新する 処理中フラグ = 処理待ち:0</div>	

機能ID		関連要件ID
	<div>5.26 Book削除バッチ</div> <div></div> <div>バッチ</div> <div>PyR-Root カタログサービス Book管理サービス 本人性確認サービス Book運用サービス</div> <div>処理内容</div> <div>1. PF利用規約を取得し、Book閉鎖時削除フラグ (deleting-data-flag) の値を確認する</div> <div>2. Book管理サービス.削除可能Book取得 API を呼び出し、Bookの自動削除を行う対象の個人を10件ずつ取得する</div> <div>3. 取得したレコードを元にBook管理サービス.強制削除 API を呼び出す。</div> <div>呼び出す際に、上記で取得した Region離脱時データ削除フラグ または Book閉鎖時削除フラグ が ture の場合、物理削除フラグに ture を設定する</div> <div>4. 本人性確認サービス.確認済本人性確認コード生成 API を呼び出し、本人性確認コードを生成する</div> <div>5. 取得したレコードを元に Book運用サービス.利用者削除 API を呼び出す。</div> <div>呼び出す際に、上記で取得した Region離脱時データ削除フラグ または Book閉鎖時削除フラグ が ture の場合、物理削除フラグに ture を設定する</div>	

ドキュメント名	版	日付	
パーソナルデータ連携モジュール アプリケーション開発ガイド別紙	1	2022/10/1	

		関連要件ID
5.27 出力データファイル管理作成バッチ		
処理内容		
<div>1. グローバル設定を取得して、Regionサービス運用の設定がONになっているかを確認する。Regionサービス運用の設定がOFFの場合、以降の処理を行わない</div> <div>2. Book管理サービス.出力データ収集アクター取得 API を呼び出し、出力データ収集アクターの承認済レコードを10件ずつ取得する</div> <div>3. 取得したレコードの condition.id をパラメータに指定して、Book管理サービス.出力データ管理取得 API を呼び出す</div> <div>4. Book管理サービスのレコードが取得できた場合、作成対象外として次のレコードを処理する</div> <div>5. 上記 2. で取得したレコードの データ返却指定 を確認する</div> <div>6. Book管理サービス.出力データ管理作成 API を呼び出し、condition 毎に出力データ管理のレコードを登録する。利用者データの出力用と個別データの出力用で最大2件のレコードが登録される</div> <div>7. Book管理サービス.出力データ管理作成 API を呼び出し、ダウンロードファイル用の出力データ管理のレコードを登録する</div> <div>8. condition の データ返却指定がすべて データの返却なし の場合 かつ、My-Condition-Dataデータ出力コード.出力タイプ が 4: Book閉鎖 または 5: Book強制閉鎖 の場合</div> <div><div>1. My-Condition-Data出力コード を使用して Book管理サービス.My-Condition-Data出力コード取得 API を呼び出す</div><div>2. 取得したレコードの BookId を使用して、Book管理サービス.Book削除予約 API を呼び出し、My-Condition-Bookを更新する</div></div>		

要件ID		関連要件ID
	<div>5.28 利用者データ削除バッチ</div> <div><pre>sequenceDiagram actor Batch as バッチ participant PxR as PxR-Root カタログサービス participant BookM as Book管理サービス participant BookU as Book運用サービス Batch->>PxR: 1公開カタログ取得 PxR-->>Batch: 2公開カタログ alt [Regionサービス運用の設定がOFF] Batch->>End: 処理終了 end loop [対象が取得できなくなるまで] loop [件数分] Batch->>BookM: 3利用者データ削除対象取得 BookM-->>Batch: 4利用者データ削除対象 Batch->>BookU: 5処理中フラグ更新 BookU-->>Batch: 6処理完了 Batch->>BookM: 7My-Condition-Book一覧取得 BookM-->>Batch: 8My-Condition-Book一覧 alt [出力タイプがRegion利用終了: 2 または Region利用自動終了: 3] Batch->>PxR: 9Regionカタログ取得 PxR-->>Batch: 10Regionカタログ Batch->>PxR: 11Region利用規約カタログ取得 PxR-->>Batch: 12Region利用規約カタログ else [出力タイプがBook閉鎖: 4 または Book強制閉鎖: 5] Batch->>PxR: 13Platform利用規約カタログ取得 PxR-->>Batch: 14Platform利用規約カタログ end end Batch->>BookU: 15利用者データ削除 BookU-->>Batch: 16処理完了 Batch->>BookM: 17出力ファイル管理ステータス更新 BookM-->>Batch: 18処理完了 end</pre></div> <div>処理内容</div> <div><ol style="list-style-type: none">グローバル設定を取得して、Regionサービス運用の設定がONになっているかを確認する。Regionサービス運用の設定がOFFの場合、以降の処理を行わないBook管理サービス.出力データ管理取得 API を呼び出し、以下の条件で出力データ管理取得のレコードを10件ずつ取得する 出力タイプ = Region利用終了: 2, Region利用自動終了: 3, Book閉鎖: 4, Book強制閉鎖: 5 承認ステータス = 承認: 1 出力ファイル種別 = アップロード: 2 出力ステータス = 作成済: 1 削除ステータス = 削除待ち: 2 処理中フラグ = 処理待ち: 0Book管理サービス.出力データ管理更新 API を呼び出し、処理中フラグを 処理中: 1 に更新する。排他制御にて更新に失敗した場合、当該レコードの処理をスキップして次のレコードを処理するBook管理サービス.My-Condition-Book一覧取得 API を呼び出し、利用者ID連携情報を取得する取得した利用者ID連携情報の actor, app, wf を使用して、自ブロックの利用者IDであることを確認する取得した 出力データ管理取得 レコードの 出力タイプ が Region利用終了: 2 または Region利用自動終了: 3 の場合、対象 Region の Region規約を取得し Region離脱時データ削除フラグ (deleting-data-flag) の値を確認する取得した 出力データ管理取得 レコードの 出力タイプ が Book閉鎖: 4 または Book強制閉鎖: 5 の場合、PF利用規約を取得し、Book閉鎖時削除フラグ (deleting-data-flag) の値を確認する取得したレコードをもとに、Book運用サービス.利用者データ削除 API を呼び出し、利用者データを削除する。 呼び出す際に、上記で取得した Region離脱時データ削除フラグ または Book閉鎖時削除フラグ が ture の場合、物理削除フラグに ture を設定する上記処理がすべて正常に終了した場合、Book管理サービス.出力データ管理更新 API を呼び出し、以下の通り更新する 処理中フラグ = 処理待ち: 0 削除ステータス = 削除済: 3上記処理でエラーが発生した場合、Book管理サービス.出力データ管理更新 API を呼び出し、以下の通り更新する 処理中フラグ = 処理待ち: 0</div>	

ドキュメント名			版	日付	
パーソナルデータ連携モジュール アプリケーション開発ガイド別紙			1	2022/10/1	
要件ID					関連要件ID
	<div>5.29 Region終了対象追加バッチ</div> <div></div> <div>処理内容</div> <div>1. グローバル設定を取得して、Regionサービス運用の設定がONになっているかを確認する。Regionサービス運用の設定がOFFの場合、以降の処理を行わない</div> <div>2. Book管理サービス.Region終了時利用者ID連携解除対象個人取得 API を呼び出し、Region利用の終了を行う対象の個人を10件ずつ取得する</div> <div>3. Book管理サービス.My-Condition-Data出力コード取得 API を呼び出し、以下の条件で My-Condition-Data出力コード のレコードを取得する bookId = 2.で取得したレコードの bookId type = 2: Region利用終了 region = Region のカタログコード、カタログバージョン</div> <div>4. 上記 3. にてレコードが取得できなかった場合、 以下のとおり Book管理サービス.データ出力準備 API を呼び出す</div> <div><pre>{ "type": 2, "pxrId": [2. で取得したレコードの pxrId], "actor": { "_value": [Region に紐づく Actor のカタログコード], "_ver": [Region に紐づく Actor のカタログバージョン] }, "region": { "_value": [Region のカタログコード], "_ver": [Region のカタログバージョン] }, "cooperation_release": true, "returnable": true }</pre></div>				

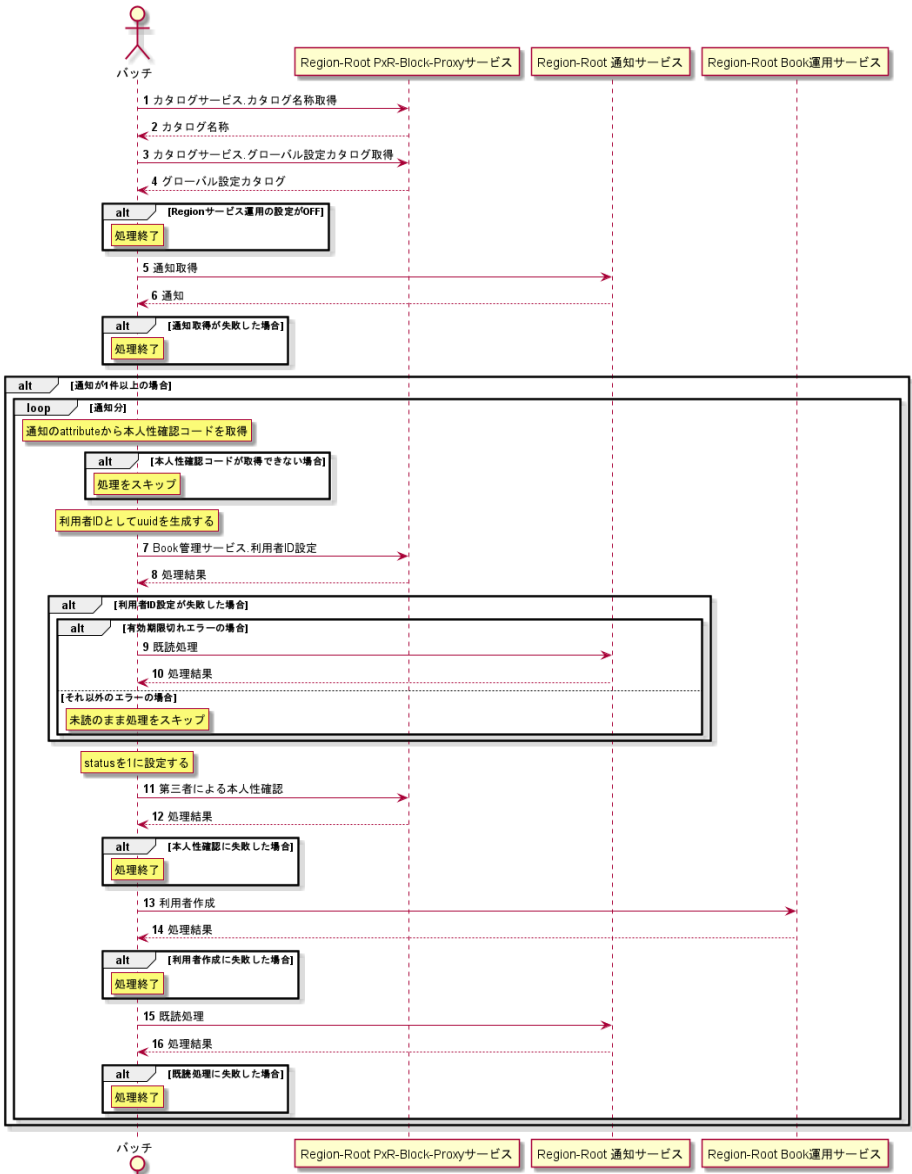
ドキュメント名	版	日付
パーソナルデータ連携モジュール アプリケーション開発ガイド別紙	1	2022/10/1

開発ID	関連要件ID
<div data-bbox="252 341 457 365">5.30 Region利用終了パッチ</div> <div data-bbox="325 371 890 1077"> </div> <div data-bbox="252 1083 1444 1751"> <p>処理内:</p> <ol style="list-style-type: none"> グローバル設定を取得して、Regionサービス運用の設定がONIになっているかを確認する。Regionサービス運用の設定がOFFの場合、以降の処理を行わない Book管理サービス.Region終了対象取得 API を呼び出し、Region利用の終了を行う対象のRegionを10件ずつ取得する Book管理サービス.My-Condition-Data出力コード取得 API を呼び出し、以下の条件で My-Condition-Data出力コード のレコードを取得する type = 2: Region自動終了 アクターカタログコード = 2: 取得したアクターカタログコード アクターカタログコード = 2: 取得したアクターカタログバージョン Regionカタログコード = 2: 取得したRegionカタログコード Regionカタログコード = 2: 取得したRegionカタログバージョン Region利用者連携解除承認ステータス = 承認不要:0 または 承認:2 Region利用者連携解除ステータス = 0: 未処理 処理中フラグ = 処理待ち:0 上記 3. にてレコードが取得できた場合、 <ol style="list-style-type: none"> Book管理サービス.My-Condition-Data出力コード更新 API を呼び出し、処理中フラグを 処理中:1 に更新する。 排他制御にて更新に失敗した場合、当該レコードの処理をスキップして次のレコードを処理する 本人性確認サービス.確認済本人性確認コード生成 API を呼び出し、本人性確認コードを生成する <pre> { "pxrid": "~", "actor": { "value": 1000002, "ver": 1 }, "region": { "value": 1000003, "ver": 1 } } </pre> Book管理サービス.利用者ID連携解除 API を呼び出す 上記処理がすべて正常に終了した場合、Book管理サービス.My-Condition-Data出力コード更新 API を呼び出し、以下の通り更新する Region利用者連携解除ステータス = 1: 処理済 処理中フラグ = 処理待ち:0 上記処理でエラーが発生した場合、Book管理サービス.My-Condition-Data出力コード更新 API を呼び出し、以下の通り更新する 処理中フラグ = 処理待ち:0 2. で取得したRegionカタログコードからRegionカタログを取得する。 取得したRegionカタログのstatusがcloseでない場合、カタログサービス.カタログ更新 API を呼び出しRegionカタログのstatusをcloseに変更する </div>	

ドキュメント名	版	日付
パーソナルデータ連携モジュール アプリケーション開発ガイド別紙	1	2022/10/1

要件ID	関連要件ID
------	--------

5.31 Region利用者連携パッチ



処理内容

- グローバル設定を取得して、Regionサービス運用の設定がONになっているかを確認する。Regionサービス運用の設定がOFFの場合、以降の処理を行わない
- 通知サービス取得 API を呼び出し、Region利用者連携申請通知を取得する
is_send(送信履歴) = false
is_unread(未読) = true
is_approval(承認済) = false
type(通知種別) = 通知.0
num(件数) = 0:全件検索
- 取得した通知のattributesから本人性確認コードを取得する。取得できなかった場合、処理をスキップする。
- 利用者IDの生成を行う。
- プロキシ経由で本人性確認サービス.利用者設定 API を呼び出し、利用者IDの設定を行う。
block = \$pxr-root-block
path = /identity-verifycate/user/settings
{
 "identifyCode": [通知で取得した本人性確認コード],
 "userId": [生成した利用者ID]
}
- プロキシ経由で本人性確認サービス.第三者による本人性確認 API を呼び出し、本人性確認を行う。
block = \$pxr-root-block
path = /identity-verifycate/verify/others/取得した本人性確認コード
{
 "status": 1
}
- Book運用サービス.利用者作成 API を呼び出し、利用者連携を行う。
{
 "identifyCode": [通知で取得した本人性確認コード],
 "attributes": {},
 "userInformation": null
}
- 通知サービス.既読処理 API を呼び出す。
{
 "id": [取得した通知.id]
}