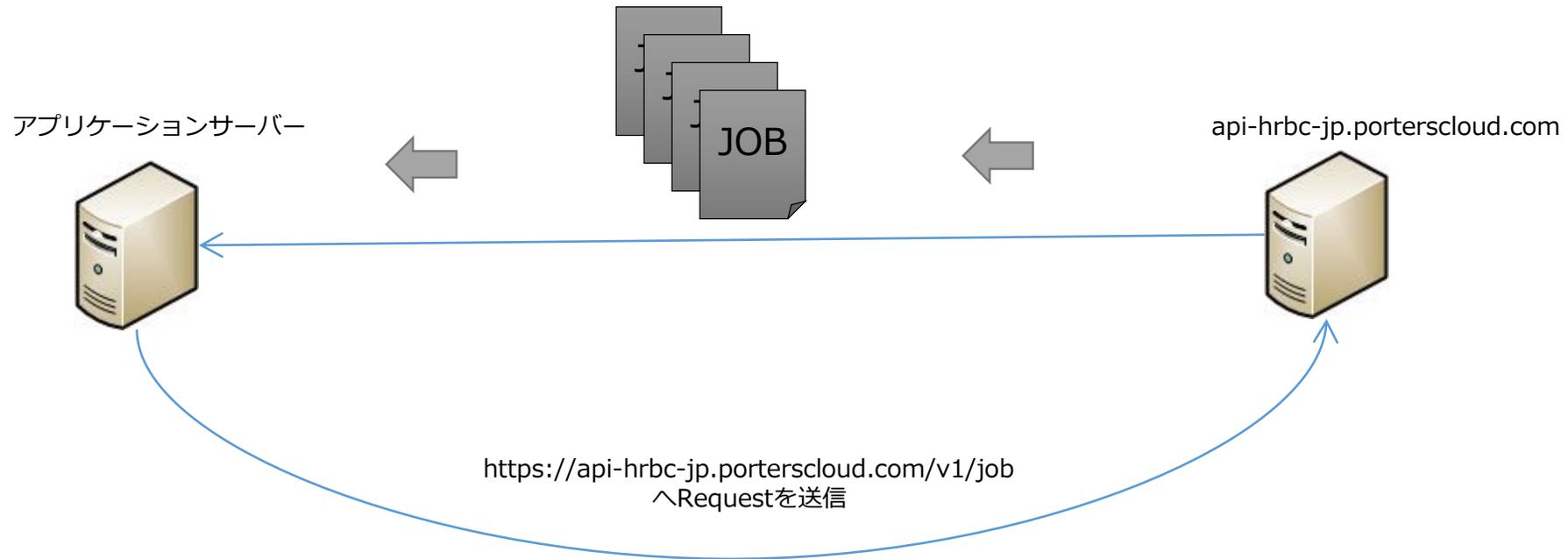


HRBC API 開発概要

HRBC APIとは

- HRBCに登録されている情報を利用したアプリケーションを開発するためのWeb APIです。
- アクセス許可を取得したリソースに対して、データを取得・登録・更新をすることができます。
 - アクセスできる情報は次のようなものがあります
 - ユーザー (User)
 - 項目情報 (Field)
 - 選択肢情報 (Option)
 - パーティション情報 (Partition)
 - 企業 (Client)
 - 企業担当者 (Recruiter)
 - JOB (Job)
 - 個人連絡先 (Candidate)
 - レジューメ (Resume)
 - 選考プロセス (Process)
 - 添付ファイル (Attachment)

例1: JOBをアプリケーションに取り込み



アプリケーションサーバーからJob APIを呼び出し、対象のJOBデータを抽出できます。

例2: キャンディデイトをHRBCに登録

アプリケーションサーバー



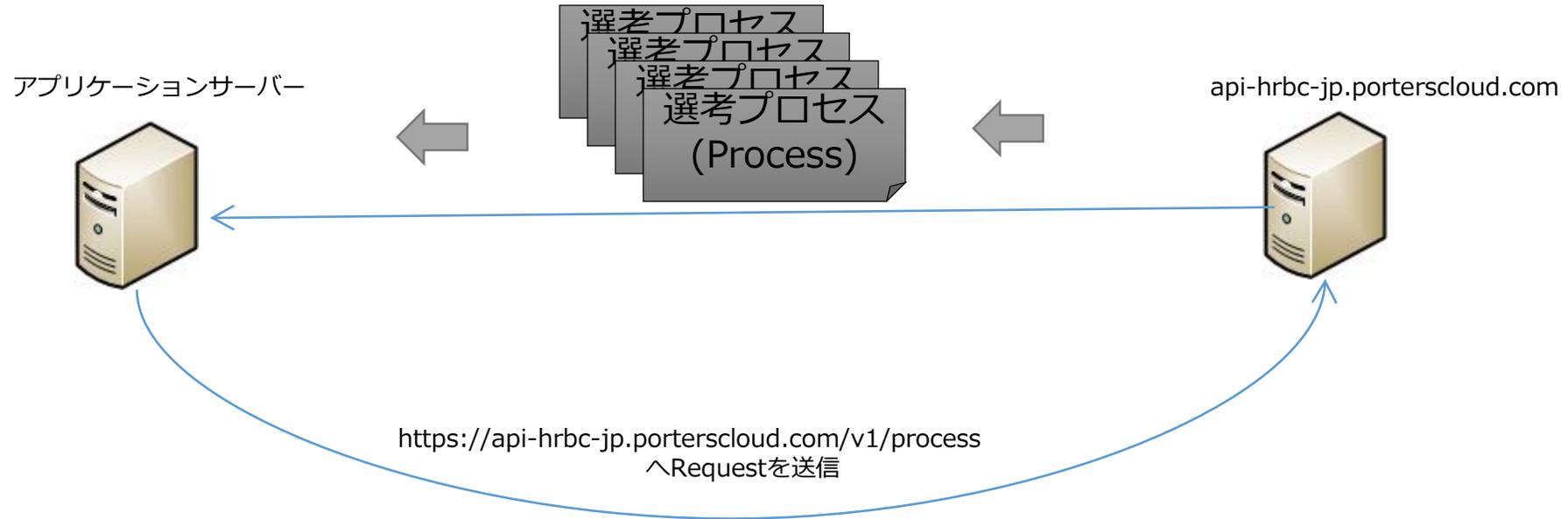
api-hrbc-jp.porterscloud.com



<https://api-hrbc-jp.porterscloud.com/v1/candidate>
へRequestを送信してキャンディデイトを登録

アプリケーションサーバー側からCandidate APIを呼び出し、キャンディデイトをHRBCへ登録できます。

例3: 選考状況を取得する



アプリケーションサーバーからProcess APIを呼び出し、対象の選考プロセスを抽出できます。

RESTful API

- APIのInterfaceはRESTをベースに設計されています。
- HTTP RequestのGETとPOSTを使用して各リソースにアクセスし、読み書きを行うことができます。
- 通信のデータフォーマットはXMLを使用します。

開発環境・実行環境

- HTTP RESTでのアクセスなので、HTTP Requestを制御できる開発言語であれば自由に選択できます。実績としてはJavaとPHPが多いですが.NET Frameworkでも問題ありません。
- アプリケーションを実行するWeb Serverを事前に登録しておく必要があります。
(事前に登録されたWeb Server以外からの認証は拒否されます)

AuthenticationとAuthorization

- 認証と権限付与はOAuth2.0ベースのアーキテクチャで構成されています
- 認証には二つのタイプがあります
 - Code認証
 - HRBCにログインしているユーザーとしてアプリケーションを動作させる方法
 - Code Direct認証
 - アプリケーションユーザーを使用してアプリケーションを動作させる方法。ユーザー操作を伴わない、バッチ処理などで使用します。

アクセスできるリソース (マスター関連)

- ユーザー (User)
- 項目情報 (Field)
- 選択肢情報 (Option)
- パーティション情報 (Partition)

API Request 例1-1

- 企業の項目を全て取得する

Method: GET

<https://api-hrbc-jp.porterscloud.com/v1/field?partition=1&resource=3&active=-1>

URLの最後にある fieldは、Fieldのデータにアクセスすることを示します。

partition=1というパラメータでは、Partition Idが1の領域にあるFieldにアクセスすることを示しています。

resource=3の3はJOBの項目を取得することを表しています。

active=-1は、使用項目と未使用項目の全てを取得することを示します。

API Request 例1-2

- 前頁のRequestのResponse

Response Body:

```
<Field Count="2" Start="0" Total="1234">  
  <Item>  
    <Field.P_Id>100</Field.P_Id>  
    <Field.P_Name>ポジション名</Field.P_Name>  
    <Field.P_Type>1</Field.P_Type>  
  </Item>  
  ...  
</Field>
```

項目の属性情報を含めたリストをXMLで取得することができます。

アクセスできるリソース (業務データ)

- 企業 (Client)
- 企業担当者 (Recruiter)
- JOB (Job)
- 個人連絡先 (Candidate)
- レジюме (Resume)
- 選考プロセス (Process)
- 添付ファイル (Attachment)

API Request 例2-1

- 日本時間の2015年3月25日から31日までに更新された求人情報 (JOB) を取得する場合のRequest

Method: GET

https://api-hrbc-jp.porterscloud.com/v1/job?

partition=1&field=Job.P_Id,Job.P_Position,Job.P_UpdateDate&

condition=Job.P_UpdateDate:ge:2015/03/24%2015:00,Job.P_UpdateDate:lt:2015/03/31%2015:00

URLの最後にある job は、JOBのデータにアクセスすることを示します。

次のpartition=1というパラメータでは、Partition Idが1の領域にあるJOBにアクセスすることを示しています。

condition=の部分で抽出条件を指定しています。ここでは、JOBの更新日(Job.P_UpdateDate)を指定しています。一つ目の条件で指定されている ge はGreater Than or Equalの略で >= を表します。二つ目の条件で指定されている lt はLess Thanの略で < を示します。これらの二つの条件を条件式として表すと次のようになります。

Job.P_UpdateDate >= 2015/03/24 15:00 AND Job.P_UpdateDate < 2015/03/31 15:00

時刻の指定で15:00が指定されていますが、これはHRBC APIでの日付と時刻の指定は全てUTCベースで行う必要があるため、日本時間への時差を考慮した指定となっているためです。

API Request 例2-2

- 前頁のRequestのResponse

Response Body:

```
<Job Start="0" Count="10" Total="2234">
  <Item>
    <Job.P_Id>10001</Job.P_Id>
    <Job.P_Position>Software Development Engineer</Job.P_Position>
    <Job.P_UpdateDate>2015/03/25</Job.P_UpdateDate>
  </Item>
  ...
</Job>
```

このようなXMLを取得することができます

API Request 例3

- 個人連絡先のIdが10001の電話番号を更新する

Method: POST

https://api-hrbc-jp.porterscloud.com/v1/candidate?partition=1

Request Body:

```
<Candidate>
  <Item>
    <Person.P_Id>10001</Person.P_Id>
    <Person.P_Telephone>0123-45-6789<Person.P_Telephone>
  </Item>
</Candidate>
```

URLの最後にある candidate は、個人連絡先のデータにアクセスすることを示します。

次のpartition=1というパラメータでは、Partition Idが1の領域にある個人連絡先にアクセスすることを示しています。

Request bodyの<Candidate>タグは、このデータが個人連絡先であることを示します。次の<Item>は一レコードを示すタグになります。<Item>タグは複数個並べて記述することができるため、一度のRequestで複数のレコードを更新することもできます。<Person.P_Id>は更新対象の個人連絡先のIdです。ここでは10001を指定しています。<Person.P_Telephone>は電話番号を示しており、0123-45-6789を指定しています。 15

アプリケーション開発時の注意事項

- 1回のRequestで処理できるレコード数はRead, Writeともに200件までです。大量のデータを処理したい場合は、200件ずつ区切って処理していく必要があります。
- APIは1アクセス単位で課金されます。できるだけ効率よくレコードを取得する方法を検討してください。
- 認証には有効期限があります。30分で認証の有効期限が切れるので、認証が切れたら再認証する仕組みを入れておく必要があります。認証用のAPIも1アクセス単位で課金されます。できるだけ認証回数を最少にできる方法を検討してください。
- 項目を決め打ちで開発した場合、HRBC側のマスター変更時にアプリケーションが動作しなくなる恐れがあります。できるだけ項目や選択肢の定義は設定ファイル等に定義して、後で変更可能な設計をしてください。