



VISMA Integraciones

Visma People - HR Core

Servicio Colaboradores

Alta / Modificación / Consulta

Documento Técnico

REST API

ÍNDICE

Servicios	4
Autenticación	4
Servicio Administration	5
Alta de Token de Acceso	5
Consulta de Tenants	6
Alta y Modificación de Colaboradores	7
Servicio Mediator	8
Alta de Colaborador	8
Modificación de Colaborador	10
Servicio Contacts	13
Alta de Domicilios	13
Modificación de Domicilios	15
Edición de Emails del Colaborador	17
Servicio Person	19
Actualización del número de legajo	19
Actualización del Reporta A	20
Alta de Estudios del Colaborador	21
Servicio Organization	22
Asociación de Modelo Organizacional a Colaborador	22
Consulta de Estructuras del Colaborador	23
Alta de Estructuras del Colaborador	24
Modificación de Estructuras del Colaborador	25
Servicio Mediator	27
Alta de Familiares (NO TIENE ACCIÓN PUT)	27
Alta de Estudios del Colaborador	31
Consulta de Títulos Educativos	33
Alta de Títulos Educativos	33
Consulta de Niveles de Educación	35
Consulta de Instituciones Educativas	35
Alta de Instituciones Educativas	36
Consulta de Estados Civiles	37
Servicio GeoLocation	37
Consulta de Nacionalidades	37
Consulta de Países	38
Consulta de Estados/Provincias	39
Consulta de Ciudades	39

Consulta de Distritos/Partidos	40
Consulta de Tipos de domicilio	41
Servicio Document	42
Consulta de Tipos de Documento	42
Servicio Person	42
Consulta de Parentesco	42
Servicio Organization	43
Consulta de Modelos Organizacionales	43
Consulta de Formas de Pago (Datos Bancarios)	44
Consulta de Tipos de Pago (Datos Bancarios)	44
Consulta de Bancos (Datos Bancarios)	45
Consulta de Tipos de Estructura	46
Consulta de Estructuras por tipo	46
Servicio Sync para sincronizar deltas de datos no sincronizados previamente	48
Introducción	48
¿Cómo funciona?	48
Flujo de interacción	48
Consulta de Aplicaciones	49
Alta y Actualización de Aplicaciones	50
Consulta de cambios en la aplicación	51
Estructura de datos de la sincronización	52

Servicios

A continuación se listan las URLs de cada servicio mencionado anteriormente, según entorno:

Producción

Servicio	Entorno	URL
Administration	Producción	https://apim.vismalatam.com/webapiadmin
Mediator	Producción	https://apim.vismalatam.com/mediator
Person	Producción	https://apim.vismalatam.com/person
GeoLocation	Producción	https://apim.vismalatam.com/geolocation
Documents	Producción	https://apim.vismalatam.com/document
Contacts	Producción	https://apim.vismalatam.com/contact
Organizations	Producción	https://apim.vismalatam.com/organization

Autenticación

En función de la seguridad, todas las solicitudes (a excepción del Servicio de Alta de Token de Acceso) deben contener los siguientes valores en el encabezado a fin de ser debidamente autorizadas:

Nombre	Valor
X-Tenant-Id	{idTenant}
Authorization	<i>Bearer</i> {token}
Ocp-Apim-Subscription-Key	9c1427b7ee214ab8b824ff7f6f5f6072

Siendo estos:

- {idTenant} el Identificador del *Tenant* (Base de Datos/Cliente), previamente obtenido a través del servicio de [Consulta de Tenants](#).

- {token} el token de acceso, previamente obtenido a través del servicio de [Alta de Token de Acceso](#).

Servicio Administration

Alta de Token de Acceso

Acción	POST
URL	{url_servicio_admin}/authentication/login
Solicitud	Url: QueryString: Body: { "username": <<user>>, "password": <<pass>> "grant_type": "password" }
Respuesta	{ "access_token": "string", "token_type": "string", "expires_in": "string" }

Observaciones:

- access_token en la respuesta es el token de acceso.
- a este servicio no se le envían los encabezados "X-Tenant-Id" ni "Authorization"

Ejemplo:

ejemplo de la base de training

```
{  
  "username": "wapiptraining",  
  "password": "1#UJs9pPrezEXaE8",  
  "grant_type": "password"
```

}

Nota: el usuario "wapitraining" tiene que tener permiso de conexión al entorno de visma people (solicitar acceso a Gustavo Tripodi y además dicho usuario tiene que tener el email validado, mas no solicitar cambio de contraseña cuando se loguee en visma people)

Consulta de Tenants

Acción	GET
URL	{url_servicio_admin}/account/tenants
Solicitud	Url: QueryString: Body:
Respuesta	<pre>[{ "Id": "string", "ClientName": "string", "DBName": "string", "TenantName": "string", "WebApiEnabled": true, "EmployeeSecurityStrategy": "string", "Version": "string", "Country": "string" }]</pre>

Observaciones:

- `ClientName` en la respuesta indica el nombre del Cliente.
- `Id` en la respuesta indica el identificador del Tenant/Base de Datos.
- a este servicio no se le envía el encabezado "X-Tenant-Id"

Alta y Modificación de Colaboradores

Para dar de alta a un Colaborador utilizando las integraciones APIs REST se debe hacer uso de los siguientes servicios:

- **Mediator**
- **Person**
- **GeoLocation**
- **Documents**
- **Contacts**
- **Organizations**

Y el orden de la secuencia debe ser la siguiente:

1. Crear el Colaborador (**Mediator**)
 - a. Datos básicos
 - b. Documentos
2. Crear lo(s) Domicilio(s) del Colaborador (**Contacts**)
3. Asociar el Modelo de Organización (**Organization**)
4. Crear Estudios del Colaborador (**Person**)
5. Crear lo(s) Familiar(es) del Colaborador (**Mediator**)

Siendo el **punto 1), 2) y 3) de carácter obligatorio**, y 4) y 5) opcional.

En realidad, el orden de 2), 3) y 4) se puede intercambiar, sólo que sí ó sí, se debe cumplir con el punto 1) primero, pues de allí se obtiene el **Id del Colaborador** generado, que es un dato solicitado en todos los demás servicios.

Servicio Mediator

Alta de Colaborador

Acción	POST
URL	{url_servicio_mediator}/api/employee-mediator
Solicitud	<p>Url:</p> <p>QueryString:</p> <p>Body:</p> <pre>{ "person": { "id": 0, "name": "string", "secondName": "string", "lastName": "string", "secondLastName": "string", "dateOfBirth": "2021-05-19T22:22:40.810Z", "gender": -1, "idMaritalStatus": 0, "idNationality": 0, "idEntityType": 0, "idCountryOfBirth": 0, "residesAbroad": true }, "documents": [{ "id": 0, "idCountry": 0, "creationDate": "2021-05-19T22:22:40.810Z", "expirationDate": "2021-05-19T22:22:40.810Z", "idDocumentType": 0, "documentIdentifier": "string", "mainDocument": true, "idEntity": 0, "idEntityType": 0 }], "hireDate" : "2021-05-19T22:22:40.810Z" }</pre>

Respuesta	<pre>{ "idPerson": 0, "fileNumber": "string" }</pre>
-----------	--

Observaciones:

- **id** lo resuelve el servicio (enviar cero u omitir).
- `idEntityType` lo resuelve el servicio (enviar cero u omitir).
- `idMaritalStatus` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Estados Civiles](#).
- `idNationality` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Nacionalidades](#).
- `idCountryOfBirth` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Países](#).
- `gender` acepta dos valores posibles: 1 (masculino) y 2 (femenino).
- `documents.id` lo resuelve el servicio (enviar cero u omitir).
- `documents.idEntity` lo resuelve el servicio (enviar cero u omitir).
- `documents.idEntityType` lo resuelve el servicio (enviar cero u omitir).
- `documents.idCountry` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Países](#).
- `documents.idDocumentType` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Tipos de Documento](#).
- `documents.creationDate` es opcional.
- `documents.expirationDate` es opcional.
- `documents.mainDocument` debe ser `=True` en uno y sólo un ítem de la lista.
- `"allowDuplicateIfDocumentExists"` debe ser `=False`
- `hireDate` se debe informar la fecha de contratación.

Ejemplo:

```
{
  "person": {
    "id": 0,
    "name": "Juan",
    "secondName": "Carlos",
    "lastName": "Medina",
    "secondLastName": "Sánchez",
    "dateOfBirth": "1983-11-17T00:00:00.000Z",
    "gender": 1,
    "idMaritalStatus": 0,
    "idNationality": 3,
```

```

    "idEntityType": 5,
    "idCountryOfBirth": 3,
    "residesAbroad": true
  },
  "documents": [
    {
      "id": 1,
      "idCountry": 3,
      "creationDate": "2022-10-17T00:00:00.000Z",
      "idDocumentType": 10,
      "documentIdentifier": "20190648248",
      "mainDocument": true,
      "idEntity": 0,
      "idEntityType": 0,
      "idEmployee": 0
    }
  ],
  "hireDate" : "2022-06-17T00:00:00.000Z"
}

```

Modificación de Colaborador

Acción	PUT
URL	{url_servicio_mediator}/api/employee-mediator
Solicitud	<p>Url:</p> <p>QueryString:</p> <p>Body:</p> <pre> { "person": { "id": 0, "name": "string", "secondName": "string", "lastName": "string", "secondLastName": "string", "dateOfBirth": "2021-05-19T22:22:40.810Z", "gender": -1, "idMaritalStatus": 0, "idNationality": 0, "idEntityType": 0, "idCountryOfBirth": 0, </pre>

	<pre> "residesAbroad": true }, "documents": [{ "id": 0, "idCountry": 0, "creationDate": "2021-05-19T22:22:40.810Z", "expirationDate": "2021-05-19T22:22:40.810Z", "idDocumentType": 0, "documentIdentifier": "string", "mainDocument": true, "idEntity": 0, "idEntityType": 0 }], "hireDate" : "2021-05-19T22:22:40.810Z" } </pre>
Respuesta	<pre> { "idPerson": 0, "fileNumber": "string" } </pre>

Observaciones:

- `id` se debe indicar el ID del Colaborador a modificar
- `idEntityType` lo resuelve el servicio (enviar cero u omitir).
- `idMaritalStatus` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Estados Civiles](#).
- `idNationality` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Nacionalidades](#).
- `idCountryOfBirth` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Países](#).
- `gender` acepta dos valores posibles: 1 (masculino) y 2 (femenino).
- `documents.id` lo resuelve el servicio (enviar cero u omitir).
- `documents.idEntity` lo resuelve el servicio (enviar cero u omitir).
- `documents.idEntityType` lo resuelve el servicio (enviar cero u omitir).
- `documents.idCountry` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Países](#).
- `documents.idDocumentType` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Tipos de Documento](#).
- `documents.creationDate` es opcional.
- `documents.expirationDate` es opcional.
- `documents.mainDocument` debe ser `=True` en uno y sólo un ítem de la lista.

- "allowDuplicateIfDocumentExists" debe ser =False
- hireDate se debe informar la fecha de contratación.

Ejemplo:

```
{
  "person": {
    "id": 635,
    "name": "Juan",
    "secondName": "Carlos",
    "lastName": "Medina",
    "secondLastName": "Sánchez",
    "dateOfBirth": "1983-11-17T00:00:00.000Z",
    "gender": 1,
    "idMaritalStatus": 12, //CASADO
    "idNationality": 1, //ARGENTINA
    "idEntityType": 0,
    "idCountryOfBirth": 3, //ARGENTINA
    "residesAbroad": false
  },
  "documents": [
    {
      "id": 0,
      "idCountry": 3, //ARGENTINA
      "creationDate": null,
      "expirationDate": null,
      "idDocumentType": 1, //DNI
      "documentIdentifier": "30652541",
      "mainDocument": true,
      "idEntity": 0,
      "idEntityType": 0
    }
  ],
  "allowDuplicateIfDocumentExists": False,
  "hireDate": "2021-05-19T22:22:40.810Z"
}
```

Servicio Contacts

Alta de Domicilios

Acción	POST
URL	{url_servicio_contacts}/api/addresses
Solicitud	<p>Url:</p> <p>QueryString:</p> <p>Body:</p> <pre>{ "id": 0, "idEntity": 0, "idEntityType": 0, "street": "string", "streetNumber": "string", "floor": "string", "department": "string", "zipCode": "string", "betweenStreets": "string", "neighborhood": "string", "idCountry": 0, "idState": 0, "idCity": 0, "default": true, "idAddressType": 0, "latitude": 0, "longitude": 0, "gmapsAddress": "string", "additionalAttributes": { "additionalProp1": {}, "additionalProp2": {}, "additionalProp3": {} } }</pre>
Respuesta	Integer

Observaciones:

- id lo resuelve el servicio (enviar cero u omitir).

- `idEntity` debe ser el valor de `idPerson` devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- `idEntityType` debe ser `=1` (Colaborador).
- `street` es requerido.
- `streetNumber` es requerido.
- `floor` es requerido.
- `department` es requerido.
- `zipCode` es requerido.
- `betweenStreets` es requerido.
- `neighborhood` es requerido.
- `idCountry` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Países](#).
- `idState` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Estados/Provincias](#).
- `idCity` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Ciudades](#).
- `idAddressType` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Tipos de Domicilio](#).
- `gmapsAddress` es requerido.
- `default` debe ser `=True` o `=False`, pudiendo tener sólo un Domicilio como principal.
- `aditionalAttributes` varía su estructura en función del país al cual pertenezca el Domicilio. Para el caso de Argentina, ver el ejemplo aquí debajo.

Ejemplo:

```
{
  "idEntity": 647,           // ID DEL COLABORADOR
  "idEntityType": 1,       // COLABORADOR
  "street": "Mario Bravo",
  "streetNumber": "780",
  "floor": "4",
  "department": "B",
  "zipCode": "1175",
  "betweenStreets": "Entre Lavalle y Tucumán",
  "neighborhood": "Almagro",
  "idCountry": 3,          // ARGENTINA
  "idState": 97,          // BUENOS AIRES
  "idCity": 25149,        // TORTUGUITAS
  "default": true,
  "idAddressType": 2,     // PARTICULAR
  "latitude": 50.4,
  "longitude": 10,
```

```

"gmmapsAddress": "Maronna 1652 Bahía Blanca",
"additionalAttributes": {
  "IdZona": 479,
  "IdPartido": 5706,
  "Sector": "sector",
  "Torre": "torre",
  "Manzana": "manzana",
  "Kilometro": "km",
  "Circunscripcion": "circu",
  "Cuerpo": "cuerpo",
  "Lote": "lote",
  "Parcela": "parcela",
  "Bloque": "bloque",
  "Seccion": "seccion",
  "Casa": "casa"
}
}

```

Modificación de Domicilios

Acción	PUT
URL	{url_servicio_contacts}/api/addresses
Solicitud	Url: QueryString: Body: <pre> { "id": 0, "idEntity": 0, "idEntityType": 0, "street": "string", "streetNumber": "string", "floor": "string", "department": "string", "zipCode": "string", "betweenStreets": "string", "neighborhood": "string", "idCountry": 0, "idState": 0, "idCity": 0, "default": true, "idAddressType": 0, </pre>

	<pre> "latitude": 0, "longitude": 0, "gmapsAddress": "string", "additionalAttributes": { "additionalProp1": {}, "additionalProp2": {}, "additionalProp3": {} } </pre>
Respuesta	integer

Observaciones:

- `id` se debe indicar el ID del domicilio a modificar.
- `idEntity` debe ser el valor de `idPerson` devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- `idEntityType` debe ser =1 (Colaborador).
- `street` es requerido.
- `streetNumber` es requerido.
- `floor` es requerido.
- `department` es requerido.
- `zipCode` es requerido.
- `betweenStreets` es requerido.
- `neighborhood` es requerido.
- `idCountry` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Países](#).
- `idState` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Estados/Provincias](#).
- `idCity` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Ciudades](#).
- `idAddressType` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Tipos de Domicilio](#).
- `gmapsAddress` es requerido.
- `default` debe ser =*True* o =*False*, pudiendo tener sólo 1 Domicilio como principal.
- `aditionalAttributes` varía su estructura en función del país al cual pertenezca el Domicilio. Para el caso de Argentina, ver el ejemplo aquí debajo.

Ejemplo:

```
{
```



```
"id": 23776 // ID DEL DOMICILIO
"idEntity": 647, // ID DEL COLABORADOR
"idEntityType": 1, // COLABORADOR
"street": "Mario Bravo",
"streetNumber": "780",
"floor": "4",
"department": "B",
"zipCode": "1175",
"betweenStreets": "Entre Lavalle y Tucumán",
"neighborhood": "Almagro",
"idCountry": 3, // ARGENTINA
"idState": 97, // BUENOS AIRES
"idCity": 25149, // TORTUGUITAS
"default": true,
"idAddressType": 2, // PARTICULAR
"latitude": 50.4,
"longitude": 10,
"gmapsAddress": "Maronna 1652 Bahía Blanca",
"additionalAttributes": {
  "IdZona": 479,
  "IdPartido": 5706,
  "Sector": "sector",
  "Torre": "torre",
  "Manzana": "manzana",
  "Kilometro": "km",
  "Circunscripcion": "circu",
  "Cuerpo": "cuerpo",
  "Lote": "lote",
  "Parcela": "parcela",
  "Bloque": "bloque",
  "Seccion": "seccion",
  "Casa": "casa"
}
}
```

Edición de Emails del Colaborador

Acción	PUT
URL	{url_servicio_contacts}/api/emails
Solicitud	Url: QueryString: Body: [{ "idEntity": 49228, "idEntityType": 1, "description": "mailLaboral@gmail.com", "type": 1 }, { "idEntity": 49228, "idEntityType": 1, "description": "mailPersonal@gmail.com", "type": 2 }]
Respuesta	

Observaciones:

- `idEntity`: Campo `idEntity` en la consulta del servicio [Alta de Colaborador](#).
- `idEntityType`: Debe ir el valor fijo "1" que corresponde a un Colaborador
- `description`: Email válido

`type`: para email laboral el valor es "1" / si es personal el valor es "2"

Servicio Person

Actualización del número de legajo

Acción	PUT
URL	{url_servicio_person}/api/employees
Solicitud	Url: QueryString: Body: { "idPerson": 0, "fileNumber": 0, "startDate": "2021-05-20T14:02:22.668Z" "endDate": "2021-05-20T14:02:22.668Z" }
Respuesta	

Observaciones:

- `idPerson` debe ser el valor de `idPerson` devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- `fileNumber` debe ser el valor informado del número de legajo en el proceso de On Boarding.
- `startDate` debe ser la fecha de contratación del Colaborador.
- `endDate` debe enviarse el valor nulo

Ejemplo:

```
{  
  "idPerson": 76149 ,  
  "fileNumber": "numerolegajo", //1234570990 por ejemplo  
  "startDate": "2021-07-01T00:30:06.084Z", //es la fecha de contratación del  
  Colaborador  
  "endDate": "2021-07-01T00:30:06.084Z" //se debe enviar el valor NULL  
}
```

Actualización del Reporta A

Acción	PATCH
URL	{url_servicio_person}/api/employees/{idEmployee}
Solicitud	<p>Url: integer idEmployee</p> <p>QueryString:</p> <p>Body:</p> <pre>[{ "value": {}, "path": "string", "op": "string", "from": "string" }]</pre>
Respuesta	

Observaciones:

- value debe ser el valor de idPerson devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- path debe ser "/idDirectReport"
- op debe ser "replace"
- from debe ser ""

Ejemplo:

```
[
  {
    "value": "2241", // ID DEL MANAGER
    "path": "/idDirectReport",
    "op": "replace",
    "from": ""
  }
]
```

Alta de Estudios del Colaborador

Acción	POST
URL	{url_servicio_person}/api/educations
Solicitud	<p>Url:</p> <p>QueryString:</p> <p>Body:</p> <pre>[{ "id": 0, "idEntity": 0, "idEntityType": 0, "idInstitution": 0, "idDegree": 0, "dateFrom": "2021-05-21T00:38:07.976Z", "dateTo": "2021-05-21T00:38:07.976Z" }]</pre>
Respuesta	[integer]

Observaciones:

- `id` lo resuelve el servicio (enviar cero u omitir).
- `idEntity` debe ser el valor de `idPerson` devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- `idEntityType` debe ser =1 (Colaborador).
- `idInstitution` debe ser el valor de `id` devuelto en la respuesta del servicio de [Consulta de Instituciones Educativas](#).
- `idDegree` debe ser el valor de `id` devuelto en la respuesta del servicio de [Consulta de Títulos Educativos](#).
- `dateFrom` debe ser una fecha válida o bien *NULL* u omitirse.
- `dateTo` debe ser una fecha válida o bien *NULL* u omitirse.

Ejemplo:

```
[
  {
    "id": 0,
```

```
"idEntity": 647,           // ID DEL COLABORADOR
"idEntityType": 1,        // COLABORADOR
"idInstitution": 42, // UNIV. DE LA MATANZA
"idDegree": 1,           // UNIVERSITARIO
"dateFrom": "2021-05-21T00:38:07.976Z",
"dateTo": "2021-05-21T00:38:07.976Z"
}
]
```

Servicio Organization

Asociación de Modelo Organizacional a Colaborador

Acción	PUT
URL	{url_servicio_organizations}/api/employee-organization-models
Solicitud	Url: QueryString: Body: { "organizationModel": 0, "employee": 0, "startDate": "2021-05-20T14:02:22.668Z" }
Respuesta	

Observaciones:

- `idOrganizationModel` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Modelos Organizacionales](#).
- `idEmployee` debe ser el valor de `idPerson` devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- `startDate` Debe informarse la fecha de contratación, misma fecha utilizada en la alta de la fase.

Ejemplo:

```
{
  "idOrganizationModel": 3,          // EMPLEO MENSUALIZADO
  "idEmployee": 647,                // ID DEL COLABORADOR
  "startDate": "2021-05-20T14:02:22.668Z"
}
```

Consulta de Estructuras del Colaborador

Acción	GET
URL	{url_servicio_organizations}/api/employee-attributes/GetEmployeeAttributesHistory/{idEmployee}
Solicitud	Url: <code>integer idEmployee</code> QueryString: <code>string(\$date-time) date(opcional)</code> Body:
Respuesta	<pre>{ "id": 0, "idTypeStructure": 0, "descriptionTypeStructure": "string", "idStructure": 0, "descriptionStructure": "string", "startDate": "2022-11-02T19:06:12.305Z", "endDate": "2022-11-02T19:06:12.305Z" }</pre>

Observaciones:

- `idEmployee` debe ser el valor de `idPerson` devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- `date` Fecha de vigencia asociada a las estructuras del colaborador

Ejemplo:

```
[
  {
    "id": 45483,
    "idTypeStructure": 29,
    "descriptionTypeStructure": "Actividad SICOSS",
    "idStructure": 521,
    "descriptionStructure": "PERS. PERM. DISC. EMPRESAS SERV. EVENT.",
    "startDate": "2003-01-01T00:00:00",
    "endDate": null
  }
]
```


]

Alta de Estructuras del Colaborador

Acción	POST
URL	{url_servicio_organizations}/api/employee-attributes
Solicitud	Url: QueryString: Body: <pre>{ "id": 0, "idEmployee": 0, "idReasonType": 0, "idStructure": 0, "startDate": "2022-11-02T19:41:43.209Z", "endDate": "2022-11-02T19:41:43.209Z", (Opcional) "reason": "string" }</pre>
Respuesta	integer

Observaciones:

- idEmployee debe ser el valor de idPerson devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- idReasonType
- idStructure
- startDate
- endDate
- reason

Ejemplo:

```

{
  "id": 45483,
  "idTypeStructure": 29,
  "descriptionTypeStructure": "Actividad SICOSS",
  "idStructure": 521,
  "descriptionStructure": "PERS. PERM. DISC. EMPRESAS SERV. EVENT.",
  "startDate": "2003-01-01T00:00:00",
  "endDate": null
}

```

Modificación de Estructuras del Colaborador

Acción	PUT
URL	{url_servicio_organizations}/api/employee-attributes
Solicitud	<p>Url:</p> <p>QueryString:</p> <p>Body:</p> <pre> { "id": 0, "idEmployee": 0, "idReasonType": 0, "idStructure": 0, "startDate": "2022-11-03T16:47:18.199Z", "endDate": "2022-11-03T16:47:18.199Z", "reason": "string" } </pre>
Respuesta	integer

Observaciones:

- idEmployee debe ser el valor de idPerson devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- idReasonType
- idStructure

- startDate
- endDate
- reason

Ejemplo:

```
{  
  "id": 0,  
  "idEmployee": 2241,  
  "idReasonType": 2,  
  "idStructure": 2502,  
  "startDate": "2022-01-01T00:00:00.000Z",  
  "endDate": "2022-12-31T16:47:18.199Z",  
  "reason": "Cambio de Empresa"  
}
```

Servicios de Consulta

Servicio Mediator

Alta de Familiares (NO TIENE ACCIÓN PUT)

Acción	POST
URL	{url_servicio_mediator}/api/relative-mediator/employee/{idEmployee}/relative
Solicitud	<p>Url: integer idEmployee</p> <p>QueryString:</p> <p>Body:</p> <pre>{ "person": { "id": 0, "name": "string", "secondName": "string", "lastName": "string", "secondLastName": "string", "dateOfBirth": "2021-05-20T01:36:08.646Z", "gender": 1, "idMaritalStatus": 0, "idNationality": 0, "idEntityType": 0, "idCountryOfBirth": 0, "residesAbroad": true }, "documents": [{ "id": 0, "idCountry": 0, "creationDate": "2021-05-20T01:36:08.646Z", "expirationDate": "2021-05-20T01:36:08.646Z", "idDocumentType": 0, "documentIdentifier": "string", "mainDocument": true, "idEntity": 0, "idEntityType": 0 }] }</pre>

	<pre>], "relative": { "idPerson": 0, "idEmployee": 0, "startDate": "2021-05-20T01:36:08.646Z", "idRelationshipType": 0, "active": true, "idHealthInsurance": 0, "idPlanHealthInsurance": 0, "familyDependent": true, "lifeInsuranceBeneficiary": true, "adopted": true, "nurserySchool": true, "nurserySchoolEndDate": "2021-05-20T01:36:08.646Z", "disability": true, "disabilityEndDate": "2021-05-20T01:36:08.646Z", "familySalaryPay": true, "familySalaryPayEndDate": "2021-05-20T01:36:08.646Z", "familySalaryPayCertificate": true, "emergencyContact": true, "studies": true, "additionalAttributes": { "additionalProp1": "string", "additionalProp2": "string", "additionalProp3": "string" } } } </pre>
Respuesta	Integer

Observaciones:

- `person` debe cumplir con los mismos requisitos definidos en el servicio de [Alta de Colaborador](#).
- `documents` debe cumplir con los mismos requisitos definidos en el servicio de [Alta de Colaborador](#)
- `relative.idPerson` lo resuelve el servicio (enviar cero u omitir).
- `relative.idEmployee` lo resuelve el servicio (enviar cero u omitir). En realidad se considera el valor `{idEmployee}` parametrizado en la URL. Este valor debe ser el valor de `idPerson` devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- `relative.idRelationshipType` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Tipos de Parentesco](#).
- `relative.active` debe ser =*True*
- `relative.startDate` debe ser una fecha válida inferior a la actual.

- `relative.idHealthInsurance` debe ser *NULL*, o bien directamente omitirse.
- `relative.idPlanHealthInsurance` debe ser *NULL*, o bien directamente omitirse.
- `relative.familyDependent` debe ser *=True o =False*.
- `relative.lifeInsuranceBeneficiary` debe ser *=True o =False*.
- `relative.adopted` debe ser *=True o =False*.
- `relative.nurserySchool` debe ser *=True o =False*.
- `relative.nurserySchoolEndDate` debe ser *NULL* si `relative.nurserySchool=False`, caso contrario debe ser una fecha cuya diferencia con la fecha de `person.dateOfBirth` sea menor a 5 años.
- `relative.disability` debe ser *=True o =False*.
- `relative.disabilityEndDate` debe ser *NULL* si `relative.disability=False`, caso contrario debe ser una fecha válida.
- `relative.familySalaryPay` debe ser *=True o =False*.
- `relative.familySalaryPayEndDate` debe ser *NULL* si `relative.familySalaryPay=False`, caso contrario debe ser una fecha válida.
- `relative.familySalaryPayCertificate` debe ser *=True o =False*.
- `relative.emergencyContact` debe ser *=True o =False*.
- `relative.studies` debe ser *=True o =False*.
- `relative.additionalAttributes` debe ser *NULL*, o bien directamente omitirse.

Ejemplo:

```
{
  "person": {
    "id": 0,
    "name": "María",
    "secondName": "Cristina",
    "lastName": "Salvador",
    "secondLastName": "Torres",
    "dateOfBirth": "1960-06-21T00:00:00.000Z",
    "gender": 2,
    "idMaritalStatus": 12,      //CASADO
    "idNationality": 1,       //ARGENTINA
    "idEntityType": 0,
    "idCountryOfBirth": 3,    //ARGENTINA
    "residesAbroad": false
  },
  "documents": [
    {
      "id": 0,
      "idCountry": 3,          //ARGENTINA
      "creationDate": null,
    }
  ]
}
```

```
    "expirationDate": null,
    "idDocumentType": 1,    //DNI
    "      ": "6290130",
    "mainDocument": true,
    "idEntity": 0,
    "idEntityType": 0
  }
],
"relative": {
  "idPerson": 0,
  "idEmployee": 0,
  "startDate": "2021-05-20T01:36:08.646Z",
  "idRelationshipType": 8,    //CONCUBINO
  "active": true,
  "idHealthInsurance": 0,
  "idPlanHealthInsurance": 0,
  "familyDependent": true,
  "lifeInsuranceBeneficiary": true,
  "adopted": true,
  "nurserySchool": false,
  "nurserySchoolEndDate": null,
  "disability": false,
  "disabilityEndDate": null,
  "familySalaryPay": false,
  "familySalaryPayEndDate": null,
  "familySalaryPayCertificate": true,
  "emergencyContact": true,
  "studies": true,
  "additionalAttributes": null
}
}
```

Alta de Estudios del Colaborador

Acción	POST
URL	{url_servicio_person}/api/educations
Solicitud	<p>Url:</p> <p>QueryString:</p> <p>Body:</p> <pre>[{ "id": 0, "idEntity": 0, "idEntityType": 0, "idInstitution": 0, "idDegree": 0, "dateFrom": "2021-05-21T00:38:07.976Z", "dateTo": "2021-05-21T00:38:07.976Z" }]</pre>
Respuesta	[integer]

Observaciones:

- `id` lo resuelve el servicio (enviar cero u omitir).
- `idEntity` debe ser el valor de `idPerson` devuelto en la respuesta del servicio de [Alta de Colaborador](#).
- `idEntityType` debe ser =1 (Colaborador).
- `idInstitution` debe ser el valor de `id` devuelto en la respuesta del servicio de [Consulta de Instituciones Educativas](#).
- `idDegree` debe ser el valor de `id` devuelto en la respuesta del servicio de [Consulta de Títulos Educativos](#).
- `dateFrom` debe ser una fecha válida o bien *NULL* u omitirse.
- `dateTo` debe ser una fecha válida o bien *NULL* u omitirse.

Ejemplo:

```
[
  {
    "id": 0,
```



```
"idEntity": 647,           // ID DEL COLABORADOR
"idEntityType": 1,        // COLABORADOR
"idInstitution": 42, // UNIV. DE LA MATANZA
"idDegree": 1,           // UNIVERSITARIO
"dateFrom": "2021-05-21T00:38:07.976Z",
"dateTo": "2021-05-21T00:38:07.976Z"
}
]
```

Consulta de Títulos Educativos

Acción	GET
URL	{url_servicio_person}/api/degrees
Solicitud	Url: QueryString: Body:
Respuesta	<pre>[{ "id": 0, "description": "string", "educationLevel": { "id": 0, "description": "string" }, "duration": 0, "percentage": 0 }]</pre>

Observaciones:

- ninguna

Alta de Títulos Educativos

Acción	POST
URL	{url_servicio_person}/api/degrees
Solicitud	Url: QueryString: Body: { "id": 0, "description": "string",

	<pre>"idEducationLevel": 0, "duration": 0, "percentage": 0 }</pre>
Respuesta	integer

Observaciones:

- `idEducationLevel` debe ser el valor de `id` devuelto en la respuesta del servicio de Consulta de Niveles de Educación.

Ejemplo:

```
{  
  "id": 0,  
  "description": "Analista programador",  
  "idEducationLevel": 6, // Terciario  
  "duration": 3,  
  "percentage": 100  
}
```

Consulta de Niveles de Educación

Acción	GET
URL	{url_servicio_person}/api/education-levels
Solicitud	Url: QueryString: Body:
Respuesta	<pre>[{ "id": 0, "description": "string" }]</pre>

Observaciones:

- ninguna

Consulta de Instituciones Educativas

Acción	GET
URL	{url_servicio_person}/api/institutions
Solicitud	Url: QueryString: Body:
Respuesta	<pre>[{ "id": 0, "description": "string", "abbreviated": "string", "externalCode": "string", }]</pre>

	<pre> "idCountry": 0, "idRegime": 0, "idInstitutionType": 0 }] </pre>
--	--

Observaciones:

- ninguna

Alta de Instituciones Educativas

Acción	POST
URL	{url_servicio_person}/api/institutions
Solicitud	<p>Url:</p> <p>QueryString:</p> <p>Body:</p> <pre> { "id": 0, "description": "string", "abbreviated": "string", "externalCode": "string", "idCountry": 0, "idRegime": 0, "idInstitutionType": 0 } </pre>
Respuesta	integer

Observaciones:

- `idEducationLevel` debe ser el valor de `id` devuelto en la respuesta del servicio de [Consulta de Niveles de Educación](#).
- `externalCode` puede ser cero, *NULL* u omitirse.
- `idRegime` debe ser cero, *NULL* u omitirse.
- `idInstitutionType` debe ser cero, *NULL* u omitirse.

Ejemplo:

```
{
  "id": 0,
  "description": "Universidad Argentina de la Empresa",
  "abbreviated": "UADE",
  "externalCode": null,
  "idCountry": 3,           // ARGENTINA
  "idRegime": 0,
  "idInstitutionType": 0
}
```

Consulta de Estados Civiles

Acción	POST
URL	{url_servicio_person}/api/marital-status
Solicitud	Url: QueryString: Body:
Respuesta	[<pre>{ "id": 0, "description": "string" }</pre>]

Servicio GeoLocation

Consulta de Nacionalidades

Acción	GET
URL	{url_servicio_geolocation}/api/nationalities
Solicitud	Url:

	QueryString: Body:
Respuesta	<pre>[{ "id": 0, "description": "string" }]</pre>

Observaciones:

- ninguna

Consulta de Países

Acción	GET
URL	{url_servicio_geolocation}/api/countries
Solicitud	Url: QueryString: Body:
Respuesta	<pre>[{ "id": 0, "description": "string", "iso": "string" }]</pre>

Observaciones:

- ninguna

Consulta de Estados/Provincias

Acción	GET
URL	{url_servicio_geolocation}/api/states/{idCountry}
Solicitud	<p>Url: integer idCountry</p> <p>QueryString:</p> <p>Body:</p>
Respuesta	<pre>[{ "id": 0, "description": "string" }]</pre>

Observaciones:

- `idCountry` debe ser uno de los valores `id` devuelto en la respuesta del servicio de Consulta de Países.

Consulta de Ciudades

Acción	GET
URL	{url_servicio_geolocation}/api/cities/{idCountry}
Solicitud	<p>Url: integer idCountry</p> <p>QueryString: integer idState (opcional) integer idCounty (opcional)</p> <p>Body:</p>
Respuesta	[

	<pre> { "id": 0, "description": "string" }] </pre>
--	---

Observaciones:

- `idCountry` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Países](#).
- `idState` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Estados/Provincias](#).
- `idCounty` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Distritos/Partidos](#).

Consulta de Distritos/Partidos

Acción	GET
URL	{url_servicio_geolocation}/api/counties/{idState}
Solicitud	<p>Url:</p> <p>integer idState</p> <p>QueryString:</p> <p>Body:</p>
Respuesta	<pre> [{ "id": 0, "description": "string" }] </pre>

Observaciones:

- `idState` debe ser uno de los valores `id` devuelto en la respuesta del servicio de [Consulta de Estados/Provincias](#).

Consulta de Tipos de domicilio

Acción	GET
URL	{url_servicio_contacts}/api/address-types
Solicitud	Url: QueryString: Body:
Respuesta	<pre>[{ "id": 0, "description": "string" }]</pre>

Observaciones:

- Ninguna.

Servicio Document

Consulta de Tipos de Documento

Acción	GET
URL	{url_servicio_documents}/api/document-types
Solicitud	Url: integer idCountry integer idEntityType QueryString: Body:
Respuesta	[{ "id": 0, "description": "string", "required": true }]

Observaciones:

- idEntityType debe llevar el valor =1 (Colaborador)
- idCountry debe ser uno de los valores id devuelto en la respuesta del servicio de Consulta de Países.

Servicio Person

Consulta de Parentesco

Acción	GET
URL	{url_servicio_organizations}/api/api/relationship-types

Solicitud	Url: QueryString: Body:
Respuesta	<pre>[{ "id": 0, "description": "string" }]</pre>

Observaciones:

- ninguna

Servicio Organization

Consulta de Modelos Organizacionales

Acción	GET
URL	{url_servicio_organizations}/api/organization-models
Solicitud	Url: QueryString: Body:
Respuesta	<pre>[{ "id": 0, "description": "string", "defaultValue": true }]</pre>

Observaciones:

- ninguna

Consulta de Formas de Pago (Datos Bancarios)

Acción	GET
URL	{url_servicio_organizations}/api/payment-methods
Solicitud	Url: QueryString: Body:
Respuesta	<pre>[{ "id": 0, "description": "string" }]</pre>

Observaciones:

- ninguna

Consulta de Tipos de Pago (Datos Bancarios)

Acción	GET
URL	{url_servicio_organizations}/api/payment-types
Solicitud	Url: QueryString: Body:

Respuesta	<pre>[{ "id": integer, "description": "string" }]</pre>
-----------	---

Observaciones:

- ninguna

Consulta de Bancos (Datos Bancarios)

Acción	GET
URL	{url_servicio_organizations}/api/structures/{idStructureType}/Structures
Solicitud	<p>Url:</p> <p>integer idStructureType</p> <p>QueryString:</p> <p>Body:</p>
Respuesta	<pre>[{ "idTercero": 52939, "idCountry": 3, "abbreviated": null, "branchName": "Centro", "idAccreditationModel": 0, "sftpCopy": false, "idAccountCreationModel": 0, "idSeizureModel": 0, "id": 48596, "description": "Banco de la República Argentina", "idStructureType": 41, "externalCode": "26533", "active": true, "structureType": null }]</pre>

Observaciones:

- `idStructureType` en la URL debe ser =41 (Banco).

Consulta de Tipos de Estructura

Acción	GET
URL	{url_servicio_organizations}/api/structure-types
Solicitud	Url: QueryString: Body:
Respuesta	<pre>[{ "id": 0, "description": "string", "system": true, "idStructureClass": 0, "order": 0, "loadPGM": "string", "active": true }]</pre>

Observaciones:

- ninguna

Consulta de Estructuras por tipo

Acción	GET
--------	-----

URL	{url_servicio_organizations}/api/{IdStructureType}/Structures
Solicitud	<p>Url:</p> <p>integer IdStructureType</p> <p>QueryString:</p> <p>Body:</p>
Respuesta	<pre>[{ "id": 0, "description": "string", "idStructureType": 0, "externalCode": "string", "active": true, "structureType": { "id": 0, "description": "string", "system": true, "idStructureClass": 0, "order": 0, "loadPGM": "string", "active": true } }]</pre>

Observaciones:

- IdStructureType = Código del tipo de estructura

Servicio Sync para sincronizar deltas de datos no sincronizados previamente

Introducción

La API de Visma cuenta con una serie de endpoints para gestionar la sincronización de datos para poder mantener actualizado repositorios de datos externos.

¿Cómo funciona?

Para poder sincronizar, se debe crear al menos una Aplicación. Esta permite llevar seguimiento de los cambios sucedidos a partir de una fecha de última actualización.

Es en función de esta última que se identifican los cambios a enviar como respuesta del servicio de sincronización.

Por ejemplo, el cliente puede crear una Aplicación "App #1", y allí informar que la fecha de última sincronización fue hace 1 mes.

Entonces, cuando consuma el servicio de sincronización indicando que desea obtener los datos a sincronizar para "App #1", la API le devolverá todos los ítems que se hayan creado/modificado/eliminado desde 1 mes hasta la fecha actual.

Luego, es responsabilidad del consumidor indicar a la API la nueva fecha de última sincronización, así la próxima vez que consulte el servicio, éste le devolverá los cambios sucedidos a partir de la última vez que se sincronizó, y de esta manera no acarrear constantemente los datos de sincronizaciones previas, lógicamente creciendo cada vez más en volumen a medida que transcurre el tiempo.

Flujo de interacción

La interacción con los endpoints debería ser la siguiente:

- Única vez..
 - POST `"/sync/applications"` para crear una aplicación.
- Frecuentemente, con la frecuencia deseada...
 - GET `"/sync"` para obtener las novedades de sincronización.
 - POST `"/sync/applications"` para actualizar la aplicación indicando la nueva fecha de última sincronización.

Ejemplo:

1. Se crea la Aplicación "App #1", indicando que la última fecha de sincronización fue hace 30 días.
2. Luego se solicitan los datos a sincronizar para "App #1". A la fecha actual, la API devuelve varios cambios al consumidor, que han sucedido entre la fecha actual y 30 días en el pasado.
3. El consumidor informa a la API la nueva fecha de última sincronización (típicamente, con la fecha actual). Este paso es sumamente importante, de lo contrario en cada nuevo consumo del servicio se estarán acarreado ítems que ya se habían informado anteriormente, agrandando el volumen de datos innecesariamente.
4. Al día siguiente, o cuando se desee volver a obtener los cambios sucedidos, la API tomará la última fecha de sincronización informada, y a partir de allí obtener qué cambios han sucedido entre esa fecha y la fecha actual.

Consulta de Aplicaciones

Acción	GET
URL	{url_servicio_organizations}/sync/applications
Solicitud	<p>Url:</p> <p>QueryString:</p> <pre>string applicationName date-time lastUpdate string orderBy integer page integer pageSize</pre> <p>Body:</p>
Respuesta	<pre>{ "totalCount": integer, "values": [{ "name": "string", "description": "string", "lastUpdate": date-time, "id": integer }] }</pre>

	<pre> }] }</pre>
--	--------------------------------

Observaciones:

- ninguna

Alta y Actualización de Aplicaciones

Acción	POST
URL	{url_servicio_organizations}/sync/applications
Solicitud	Url: QueryString: Body: { "name": "string", "description": "string", "lastUpdate": "2022-11-16T21:03:17.799Z" }
Respuesta	{ "name": "string", "description": "string", "lastUpdate": date-time, "id": integer }

Observaciones:

- lastUpdate: Es la fecha desde la cual se realizará la próxima sincronización.

Consulta de cambios en la aplicación

Acción	GET
URL	{url_servicio_organizations}/sync
Solicitud	<p>Url:</p> <p>QueryString:</p> <pre>string applicationName string parentEntity string entity integer page integer pageSize</pre> <p>Body:</p>
Respuesta	<pre>{ "parentId": 0, "typeId": 0, "type": "string", "entity": "string", "parenEntity": 0, "changeUpdate": "2022-11-01T11:05:23.218Z", "changeUser": "string", "data": {} }</pre>

Observaciones:

- `type / typeId` : Son el identificador y descripción del tipo de operación: 1=Insert | 2=Update | 3=Delete
- `parentId` : El identificador del registro del cual "depende".
Cada ítem tiene una "agrupación" que se utiliza para asociar/agrupar algunos datos. Por ejemplo, tanto `Employee`, `Family_Member`, `Address`, `Phone`, `Phase`, etc. dependen de la Parent Entity "Employee".
- `entity/parentEntity` : Son la entidad sobre la cual se está informando (por ejemplo, "Master_Structure" y la entidad "padre" a la que está vinculada (por ejemplo, "Organization").
- `changeDate`: La fecha y hora en que sucedió el cambio.
- `changeUser`: El usuario de sistema que realizó el cambio.

- `data` : El registro en concreto que ha sido cambiado. [ver estructura de datos](#).
Aquí es importante notar que la estructura de datos informada en cada caso coincide exactamente con la estructura devuelta en el endpoint a través del cual se obtendrían sus datos si se consume.

Estructura de datos de la sincronización

Al consumir el endpoint `GET "/sync"`, se obtiene una lista de N ítems donde N es la cantidad de cambios sucedidos desde la última fecha de actualización informada. Estos ítems tienen la siguiente estructura:

Address	<pre>{ "id": integer, "parentId": integer, "typeId": integer, "type": "string", "entity": "Address", "parentEntity": "Employee", "changeDate": "date-time", "changeUser": "string", "data": { "employeeId": integer, "isMain": true, "type": "string", "tower": "string", "floor": "string", "officeDepartment": "string", "square": "string", "body": "string", "batch": "string", "plot": "string", "block": "string", "circumscription": "string", "section": "string", "house": "string", "houseNumber": "string", "betweenStreets": "string", "zipCode": "string", "neighborhood": "string", "street": "string", "email": "string", "cpa": "string", "longitude": "string", "latitude": "string", "country": "string", "city": "string", } }</pre>
---------	---

	<pre> "locality": "string", "zone": "string", "kilometer": "string", "state": "string", "id": 0 } } </pre>
Bank_account	<pre> { "id": integer, "parentId": integer, "typeId": integer, "type": "Update", "entity": "Bank_Account", "parentEntity": "Employee", "changeDate": "date-time", "changeUser": "string", "data": { "active": boolean, "paymentMethod": "string", "bankCode": "string", "accountNumber": "string", "cbuNumber": "string", "paymentType": "string", "debitAccountCompany": "string", "branchCode": "string", "branch": "string", "percentage": integer, "id": string } } </pre>
Documents	<pre> { "id": integer, "parentId": integer, "typeId": integer, "type": "string", "entity": "Document", "parentEntity": "Employee", "changeDate": "date-time", "changeUser": "string", "data": { "number": "string", "type": { "initials": "string", "description": "string", "id": integer }, "country": { "description": "string", </pre>

	<pre> "isDefault": boolean, "codeBank": "string", "codeISO": "string", "id": integer }, "emissionDate": "date-time", "dueDate": "date-time", "legalPersonality": "string", "certifiedInstitution": "string", "id": integer } </pre>
Employee	<pre> { "id": integer, "parentId": integer, "typeId": integer, "type": "string", "entity": "Employee", "parentEntity": "Employee", "changeDate": "date-time", "changeUser": "string", "data": { "countryOfBirth": "string", "dateOfCountryEntry": "date", "email": "string", "fiscalNumbers": [{ "number": "string", "type": { "initials": "string", "description": "string", "id": integer }, }, { "country": { "description": "string", "isDefault": boolean, "codeBank": string, "codeISO": string, "id": integer }, }, "emissionDate": date, "dueDate": date, "legalPersonality": "string", "certifiedInstitution": "string", "id": integer }], "genre": "string", "maritalStatus": { "date": "date", "description": "string", } } </pre>

	<pre> "id": integer }, "nationalIdentificationNumbers": [{ "number": "string", "type": { "initials": "string", "description": "string", "id": integer } }], "nationalities": [{ "description": "string ", "isDefault": boolean, "externalDescription": string, "externalCode": "string", "id": integer }], "nickname": "string", "studyLevel": { "description": "string", "id": integer }, "handicapped": boolean, "handicapType": string, "placeOfBirth": "string", "phones": [{ "areaCode": string, "number": "string", "type": "string", "isDefault": boolean, "id": integer }], "reportsToExternalId": "string", "isActive": boolean, "organizationModelId": integer, "organizationModel": "string", "hiringDate": "date", "dateOfBirth": "date", "externalId": "string", "familyName": "string", "firstName": "string", "imageUrl": string, "lastName": "string", "middleName": "string", "id": inreger } }</pre>
--	---

Family_Member	<pre>{ "id": integer, "parentId": integer, "typeId": integer, "type": "Update", "entity": "Document", "parentEntity": "Family_Member", "changeDate": "date-time", "changeUser": "String", "data": { "number": "string", "type": { "initials": "string", "description": "string", "id": integer }, "country": { "description": "string", "isDefault": bool, "codeBank": string, "codeISO": string, "id": integer }, "emissionDate": date-time, "dueDate": date-time, "legalPersonality": "string", "certifiedInstitution": "string", "id": integer } }</pre>
Holiday	<pre>{ "id": integer, "parentId": integer, "typeId": integer, "type": "string", "entity": "Holiday", "parentEntity": "Leave", "changeDate": "date-time", "changeUser": "string", "data": { "id": integer, "accountId": integer, "account": "string", "total": integer, "balance": integer } },</pre>

<p>Leave</p>	<pre>{ "id": integer, "parentId": integer, "typeId": integer, "type": "string", "entity": "Leave", "parentEntity": "Leave", "changeDate": "date-time", "changeUser": "string", "data": { "employeeId": integer, "typeLeaveId": integer, "observations": "string", "typeLeaveDescription": "string", "leaveState": integer, "leaveStateDescription": "string", "dateFrom": "date-time", "dateTo": "date-time", "workingDays": integer, "holidays": integer, "days": integer, "hours": integer, "dayType": integer, "order": string, "startTime": date-time, "endTime": date-time, "id": integer } }</pre>
<p>master_structure</p>	<pre>{ "id": integer, "parentId": integer, "typeId": integer, "type": "string", "entity": "Master_Structure", "parentEntity": "integer", "changeDate": "date-type", "changeUser": "string", "data": { "description": "string", "extendedDescription": "string", "externalId": "string", "active": boolean, "type": { "description": "string", "id": integer }, "id": integer } }</pre>

<p>master_structure_type</p>	<pre>{ "id": integer, "parentId": integer, "typeId": integer, "type": "string", "entity": "Master_Structure_Type", "parentEntity": "String", "changeDate": "date-type", "changeUser": "string", "data": { "description": "string", "id": integer } }</pre>
<p>pay_element</p>	<pre>{ "id": integer, "parentId": integer, "typeId": integer, "type": "string", "entity": "Pay_Element", "parentEntity": "Pay_Element", "changeDate": "date-time", "changeUser": "string", "data": { "employeeId": integer, "employeeExternalId": "string", "conceptId": "string", "conceptExternalId": "string", "parameterId": integer, "dateFrom": date-time, "dateTo": date-time, "value": decimal, "retroactivityStartYear": date-time, "retroactivityStartMonth": date-time, "retroactivityEndYear": date-time, "retroactivityEndMonth": date-time, "reasonType": { "id": integer, "description": string, "reason": string }, "eventDate": date-time, "id": integer } }</pre>
<p>phase</p>	<pre>{ "id": integer, "parentId": integer, "typeId": integer,</pre>

	<pre> "type": "string", "entity": "Phase", "parentEntity": "Employee", "changeDate": "date-time", "changeUser": "String", "data": { "advanceNotice": string, "compensation": boolean, "endDate": date-time, "holidays": boolean, "real": boolean, "recognizedStartDate": boolean, "salary": boolean, "startDate": "date-time", "status": boolean, "employeeId": integer, "decouplingCause": string, "id": integer } } </pre>
phone	<pre> { "id": integer, "parentId": integer, "typeId": integer, "type": "Update", "entity": "Phone", "parentEntity": "Employee", "changeDate": "date-time", "changeUser": "string", "data": { "areaCode": string, "number": "string", "type": "string", "isDefault": boolean, "id": integer } } </pre>
structure	<pre> { "id": integer, "parentId": integer, "typeId": integer, "type": "string", "entity": "Structure", "parentEntity": "Employee", "changeDate": "date-time", "changeUser": "string", "data": { "structureId": integer, "dateFrom": "date-time", </pre>

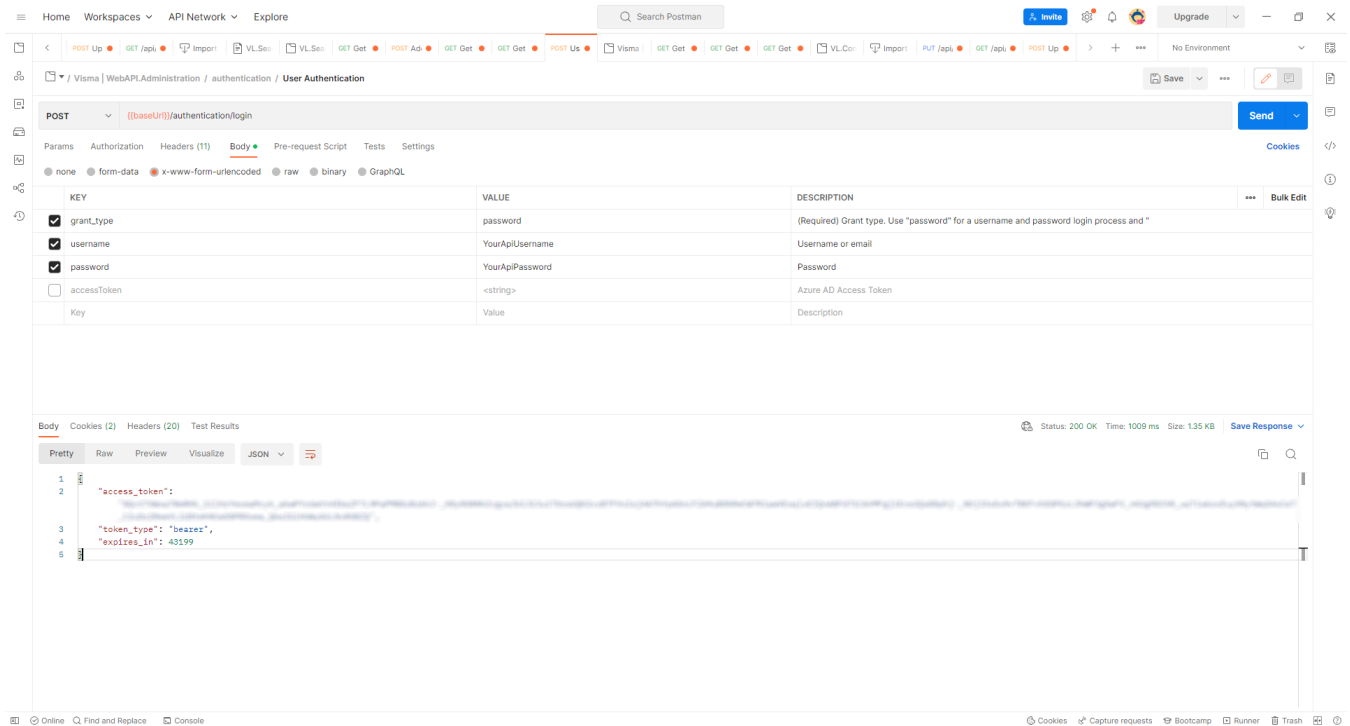
	<pre>"dateTo": date-time, "description": "string", "externalId": "string", "type": { "description": "string", "id": integer }, "reason": { "id": integer, "description": "string", "reason": "string" }, "model": { "description": "string", "id": integer }, "id": integer } }</pre>
--	---

Ejemplo de utilización de sync

Login - generación del token

POST <https://webapiadmin.vismalata.com/authentication/login>

Con el usuario y la contraseña, llamamos al servicio de login para obtener el token.



The screenshot shows a REST client interface for a POST request to `[[BaseURI]]/authentication/login`. The request body is form-data and includes the following fields:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> grant_type	password	(Required) Grant type. Use "password" for a username and password login process and "
<input checked="" type="checkbox"/> username	YourApiUsername	Username or email
<input checked="" type="checkbox"/> password	YourApiPassword	Password
<input type="checkbox"/> accessToken	<string>	Azure AD Access Token
Key	Value	Description

The response body (JSON) is:

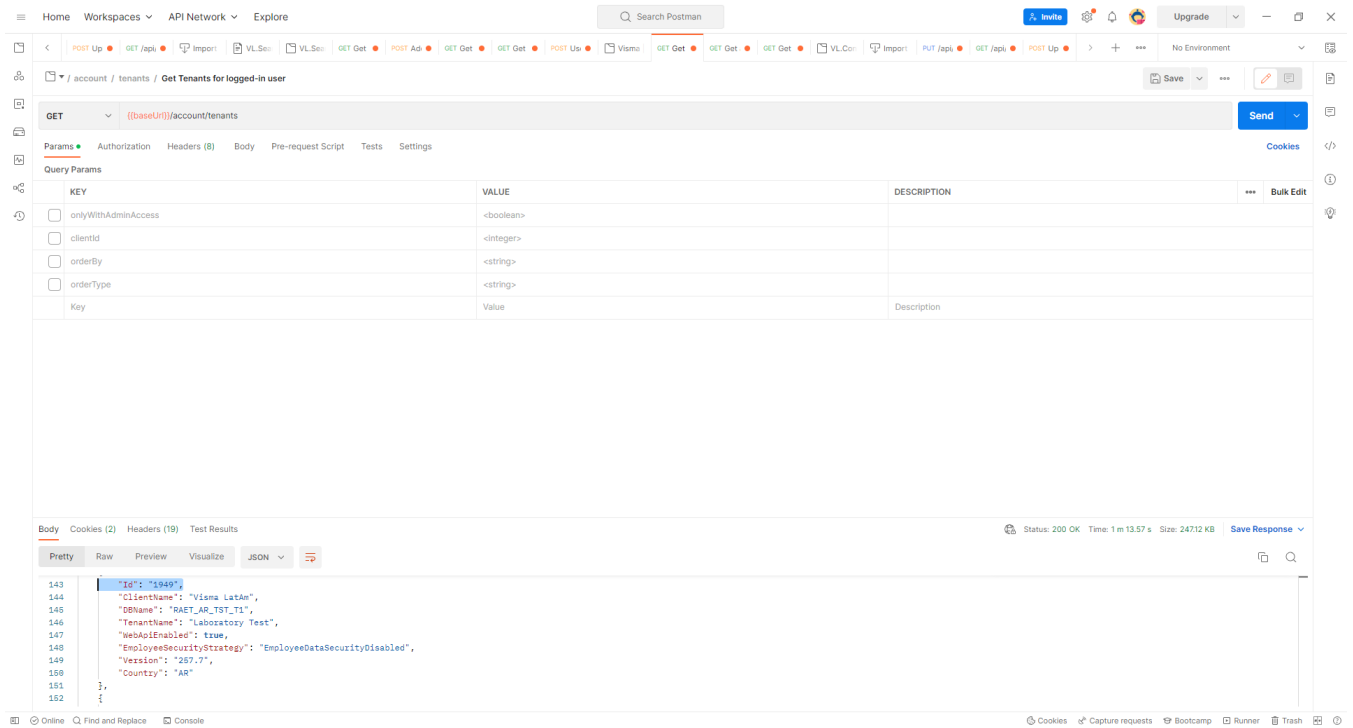
```

1 {
2   "access_token": "...",
3   "token_type": "bearer",
4   "expires_in": 43199
5 }
  
```

Obtener los Id's de los tenant

GET <https://webapiadmin.vismalatam.com/account/tenants>

Desde este servicio obtenemos los id's de los tenants que luego utilizaremos en los headers como "X-RAET-Tenant-Id"



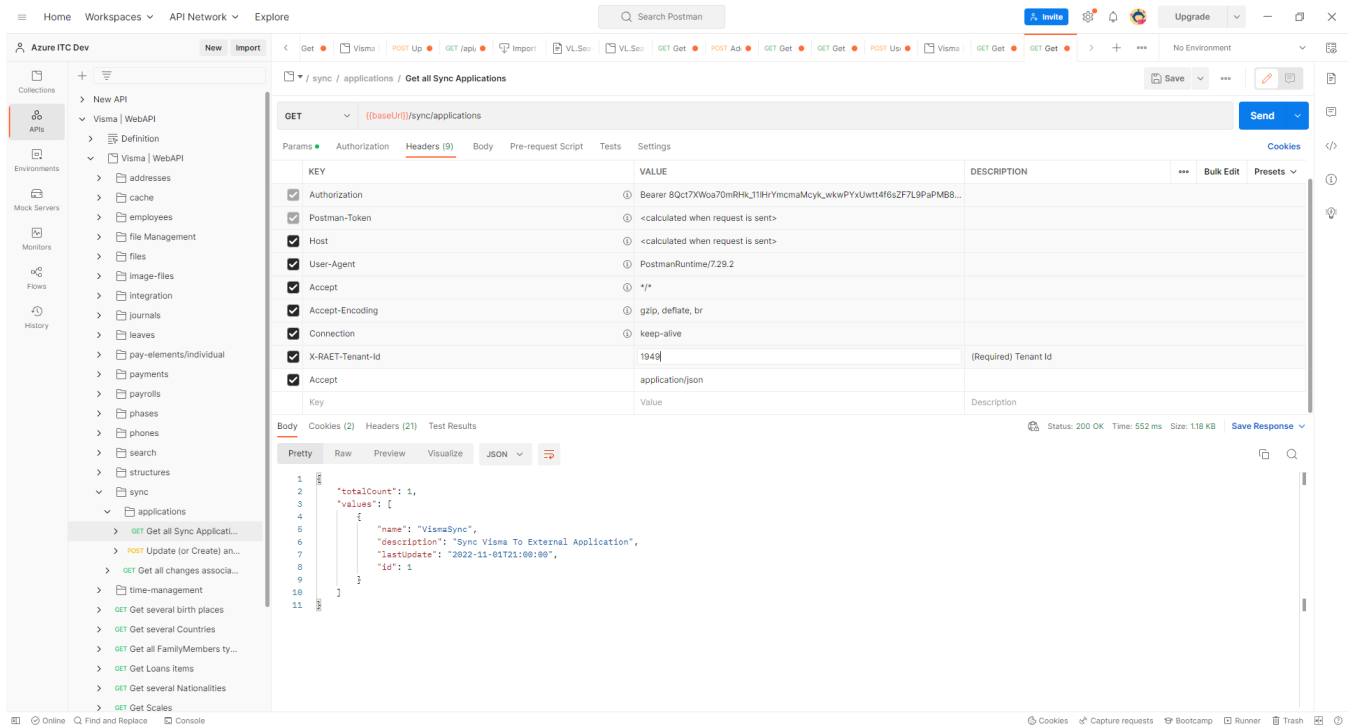
The screenshot shows a Postman interface with a GET request to `[[BaseURL]]/account/tenants`. The response body is a JSON object with the following structure:

```
143 {
144   "id": "1949",
145   "clientName": "Visma LatAm",
146   "dbName": "BAET_AR_TST_T2",
147   "tenantName": "Laboratory Test",
148   "webApiEnabled": true,
149   "employeeSecurityStrategy": "EmployeeDataSecurityDisabled",
150   "version": "267.7",
151   "country": "AR"
152 }
```

Obtener las aplicaciones disponibles en el cliente

GET <https://webapi.vismalatam.com/sync/applications>

Consultamos las aplicaciones disponibles para sincronizar. Si no tenemos ninguna, creamos una desde [Alta y Actualización de Aplicaciones](#)



The screenshot shows a Postman interface for a GET request to `[[baseUri]]/sync/applications`. The request headers are as follows:

KEY	VALUE	DESCRIPTION
Authorization	Bearer 8Oct7XWoa70mRHk_11HrYmcmMcYk_wwPYxUwt4f6s2F7L9PaPMB8...	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.29.2	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
X-RAET-Tenant-Id	1946	(Required) Tenant Id
Accept	application/json	
Key	Value	Description

The response body is a JSON object:

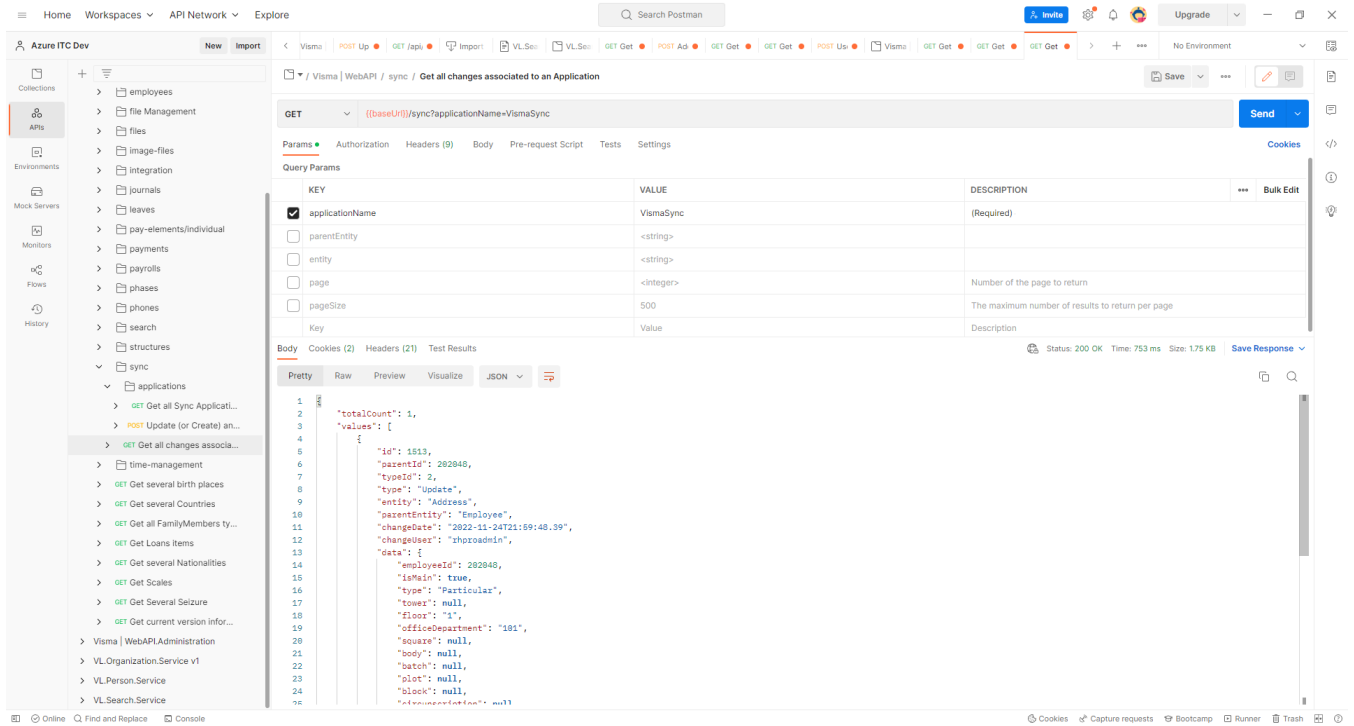
```

1 {
2   "totalCount": 1,
3   "values": [
4     {
5       "name": "VismaSync",
6       "description": "Sync Visma To External Application",
7       "lastupdate": "2022-11-01T21:08:08",
8       "id": 1
9     }
10  ]
11 }
  
```

Consulta de las diferencias a sincronizar

GET <https://webapi.vismalatam.com//sync?applicationName=VismaSync>

Consultamos los cambios que se produjeron desde la última sincronización, en base a la fecha de la aplicación. Podemos ver que hay una sincronización.



GET / Visma | WebAPI / sync / Get all changes associated to an Application

Params • Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> applicationName	VismaSync	(Required)
<input type="checkbox"/> parentEntity		
<input type="checkbox"/> entity	<string>	
<input type="checkbox"/> page	<integer>	Number of the page to return
<input type="checkbox"/> pageSize	500	The maximum number of results to return per page
Key	Value	Description

Body Cookies (2) Headers (21) Test Results

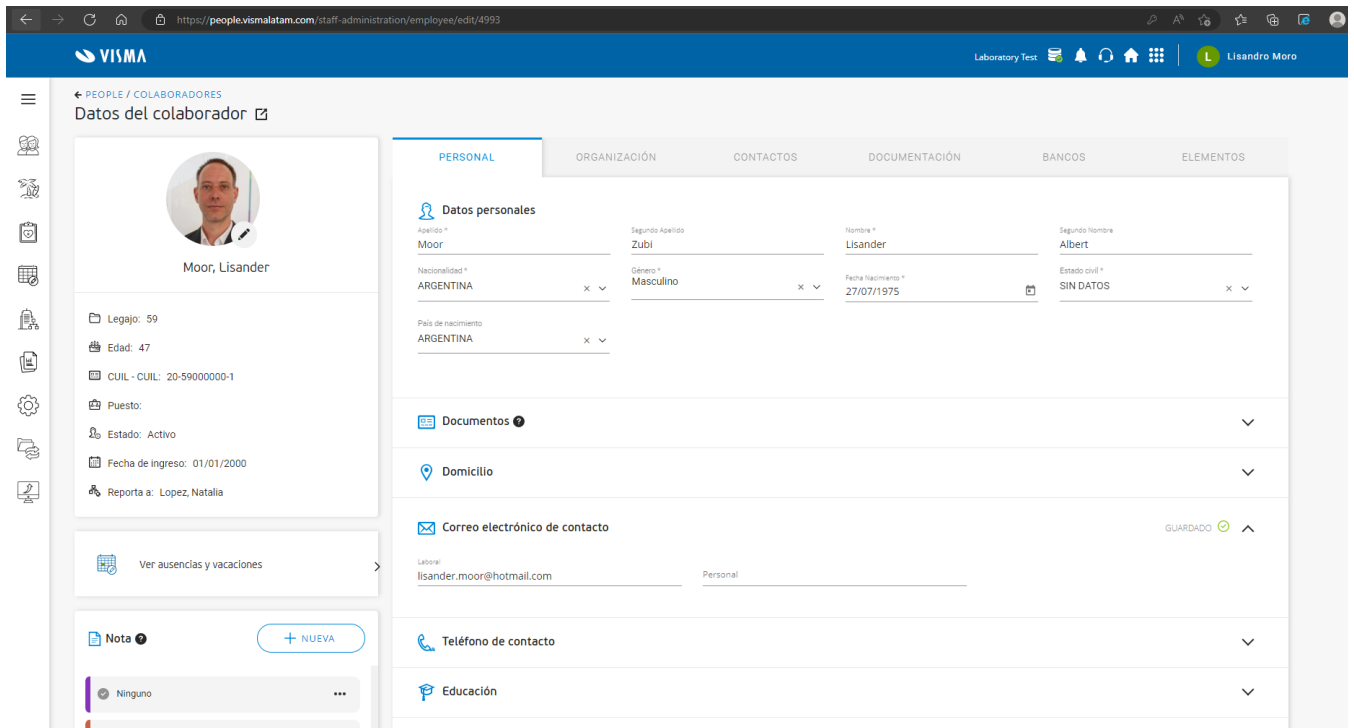
Status: 200 OK Time: 753 ms Size: 1.75 KB Save Response

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
{
  "totalCount": 1,
  "values": [
    {
      "id": 1513,
      "parentID": 202045,
      "typeID": 2,
      "type": "Update",
      "entity": "Address",
      "parentEntity": "Employee",
      "changeDate": "2022-11-24T21:59:48.39",
      "changeUser": "shpoadmIn",
      "data": {
        "employeeId": 202045,
        "login": true,
        "type": "Particular",
        "tower": null,
        "floor": "1",
        "officeDepartment": "891",
        "square": null,
        "body": null,
        "batch": null,
        "plot": null,
        "block": null,
        "address": null
      }
    }
  ]
}

```

Modificación de datos de un empleado desde la aplicación, ej: el correo electrónico.



PEOPLE / COLABORADORES

Datos del colaborador

PERSONAL ORGANIZACIÓN CONTACTOS DOCUMENTACIÓN BANCOS ELEMENTOS

Datos personales

Apellido * Moor Segundo Apellido Zubi Nombre * Lisander Segundo Nombre Albert

Nacionalidad * ARGENTINA x v Género * Masculino x v Fecha Nacimiento * 27/07/1975 Estado civil * SIN DATOS x v

País de nacimiento ARGENTINA x v

Documentos v

Domicilio v

Correo electrónico de contacto GUARDADO v

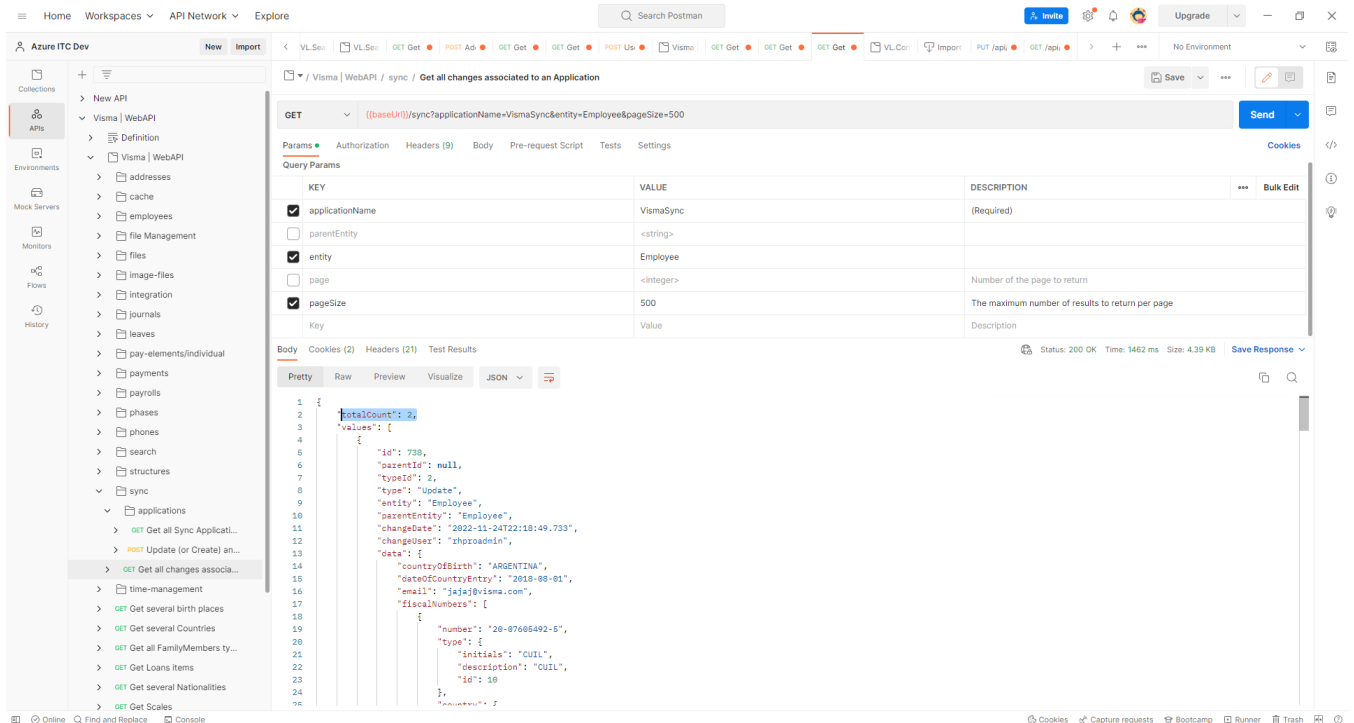
Laboral: lisander.moor@hotmail.com Personal

Teléfono de contacto v

Educación v

GET <https://webapi.vismalatam.com/sync?applicationName=VismaSync>

Volvemos a consultar el servicio de sync y vemos los cambios reflejados. En este caso podemos ver 2 cambios posteriores a la fecha de la aplicación.



The screenshot shows a REST client interface with the following details:

- Request:** GET `(((baseUri)))/sync?applicationName=VismaSync&entity=Employee&pageSize=500`
- Query Params:**

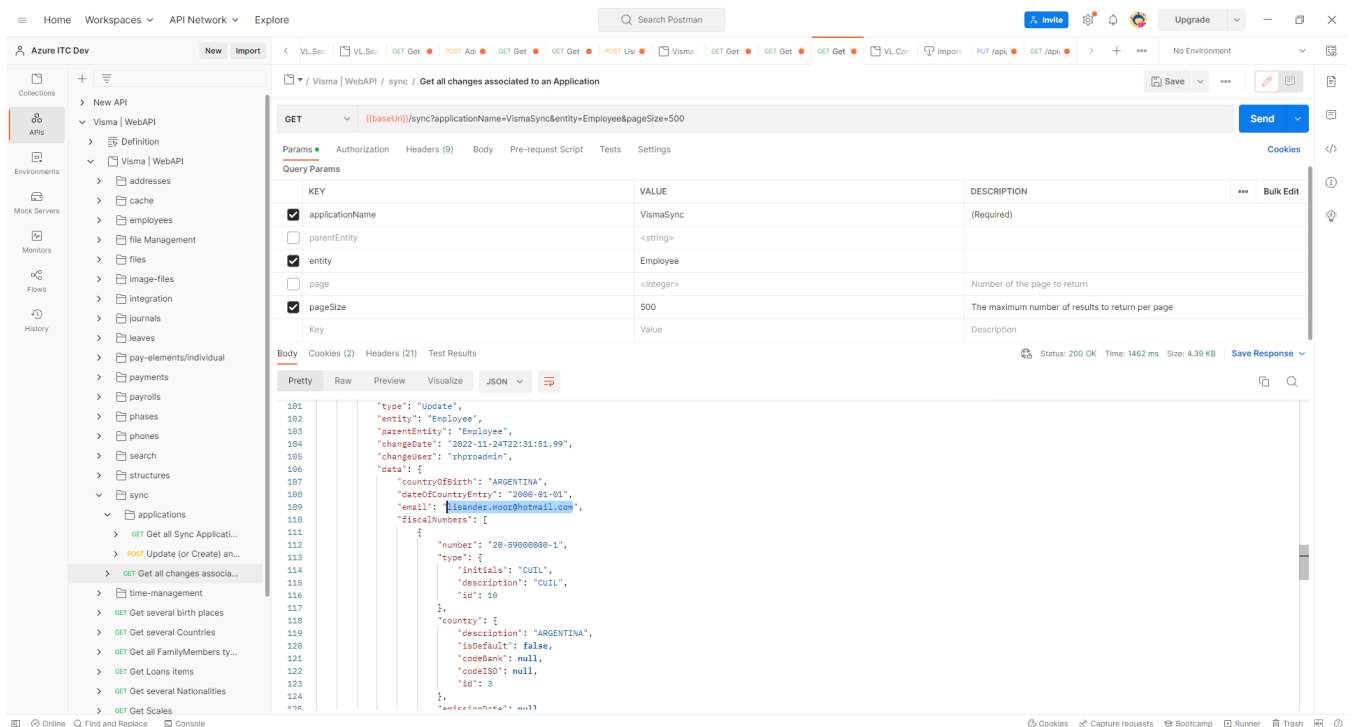
KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> applicationName	VismaSync	(Required)
<input type="checkbox"/> parentEntity	<string>	
<input checked="" type="checkbox"/> entity	Employee	
<input type="checkbox"/> page	<integer>	Number of the page to return
<input checked="" type="checkbox"/> pageSize	500	The maximum number of results to return per page
Key	Value	Description
- Response Body (Pretty):**

```

1  {
2    "totalCount": 2,
3    "values": [
4      {
5        "id": 738,
6        "parentid": null,
7        "typeid": 2,
8        "type": "Update",
9        "entity": "Employee",
10       "parentEntity": "Employee",
11       "changeDate": "2022-11-24T22:18:49.733",
12       "changeUser": "shproadIn",
13       "data": {
14         "countryOfBirth": "ARGENTINA",
15         "dateOfCountryEntry": "2010-00-01",
16         "email": "sajaj@visma.com",
17         "fiscalNumbers": [
18           {
19             "number": "20-87665492-6",
20             "type": {
21               "initials": "CUIL",
22               "description": "CUIL",
23               "id": 10
24             }
25           }
26         ]
27       }
28     }
29   ]
30 }

```

Dentro del detalle de la respuesta, se observan los datos del empleado modificado.



The screenshot shows a REST client interface with the following details:

- Request:** GET `(((baseUri)))/sync?applicationName=VismaSync&entity=Employee&pageSize=500`
- Query Params:**

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> applicationName	VismaSync	(Required)
<input type="checkbox"/> parentEntity	<string>	
<input checked="" type="checkbox"/> entity	Employee	
<input type="checkbox"/> page	<integer>	Number of the page to return
<input checked="" type="checkbox"/> pageSize	500	The maximum number of results to return per page
Key	Value	Description
- Response Body (Pretty):**

```

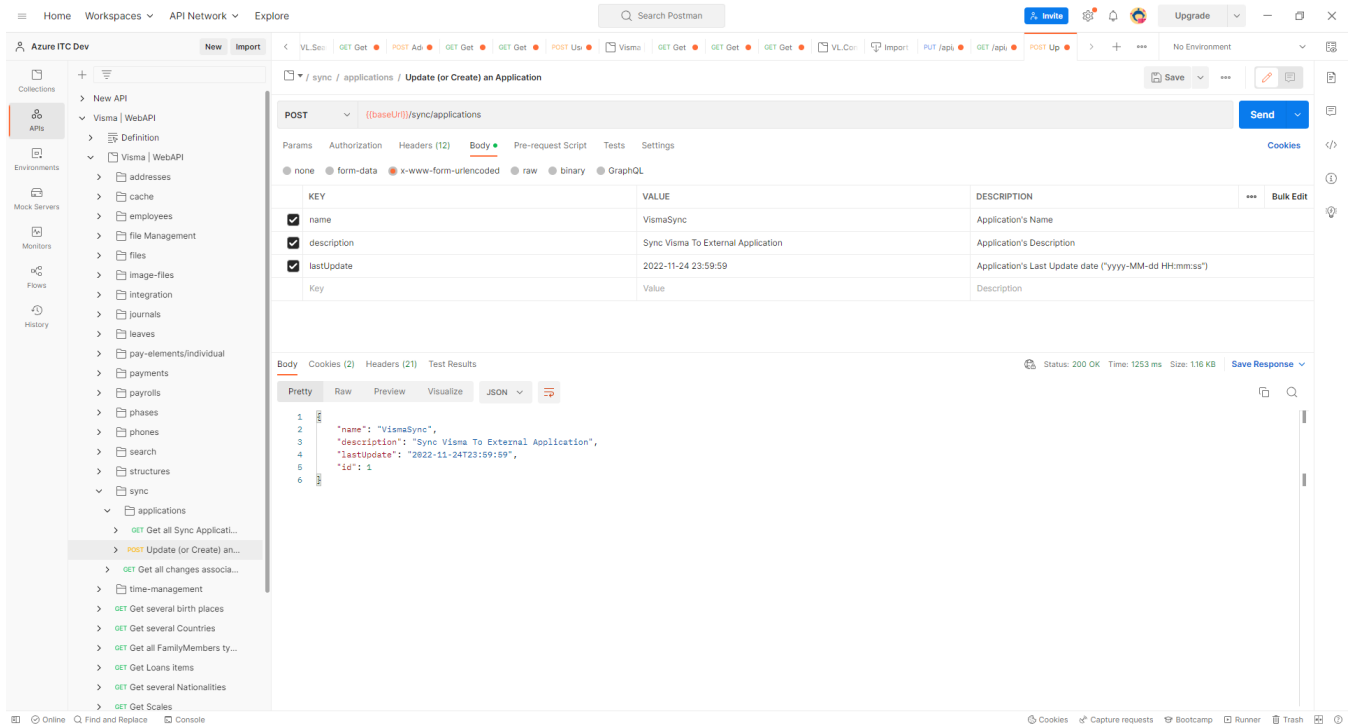
101  "type": "Update",
102  "entity": "Employee",
103  "parentEntity": "Employee",
104  "changeDate": "2022-11-24T22:31:51.99",
105  "changeUser": "shproadIn",
106  "data": {
107    "countryOfBirth": "ARGENTINA",
108    "dateOfCountryEntry": "2000-01-01",
109    "email": "lisander.moo@hotmail.com",
110    "fiscalNumbers": [
111      {
112        "number": "20-5900000-1",
113        "type": {
114          "initials": "CUIL",
115          "description": "CUIL",
116          "id": 10
117        }
118      }
119    ],
120    "country": {
121      "description": "ARGENTINA",
122      "isDefault": false,
123      "codeBank": null,
124      "codeISO": null,
125      "id": 3
126    },
127    "active": true
128  }
129 }

```

Actualizar la fecha de la última sincronización

POST <https://webapi.vismalatam.com/sync/applications>

Modificamos la fecha de sincronización de la aplicación, para que solo se muestren los cambios posteriores.



The screenshot shows the Postman interface for a POST request to the endpoint `sync/applications`. The request body is a JSON object with the following fields:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	VismaSync	Application's Name
<input checked="" type="checkbox"/> description	Sync Visma To External Application	Application's Description
<input checked="" type="checkbox"/> lastUpdate	2022-11-24 23:59:59	Application's Last Update date ('yyyy-MM-dd HH:mm:ss')
Key	Value	Description

The response body is shown in the Pretty view, displaying the following JSON:

```
1 {
2   "name": "VismaSync",
3   "description": "Sync Visma To External Application",
4   "lastupdate": "2022-11-24T23:59:59",
5   "id": 1
6 }
```

The status bar at the bottom indicates a successful response: Status: 200 OK, Time: 1253 ms, Size: 116 KB.

Volvemos a consultar las sincronizaciones y vemos que ya no hay modificaciones pendientes.

Home Workspaces API Network Explore Search Postman

Azure ITC Dev New Import

Visma | WebAPI / sync / Get all changes associated to an Application

GET {{baseUri}}/sync/applicationName=VismaSync

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION	***	Bulk Edit
<input checked="" type="checkbox"/> applicationName	VismaSync	(Required)		
<input type="checkbox"/> parentEntity	<string>			
<input type="checkbox"/> entity	Employee			
<input type="checkbox"/> page	<integer>	Number of the page to return		
<input type="checkbox"/> pageSize	500	The maximum number of results to return per page		
Key	Value	Description		

Body Cookies (2) Headers (21) Test Results Status: 200 OK Time: 567 ms Size: 1.07 KB Save Response

Pretty Raw Preview Visualize JSON

```

1
2
3
4
{"totalCount": 0,
 "values": []}

```

Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash