Documentación de API

ÍNDICE

1. Documentación API

1.1. Autenticación

En esta página se describe cómo funciona la autenticación en el servicio Latch.

Todas las solicitudes a la API de Latch deben estar **firmadas**. El proceso de firma es una versión simplificada del protocolo OAuth de dos vías.

Cada solicitud HTTP a la API **debe ir acompañada de dos encabezados** de autenticación: Authorization y Date.

1.1.1. El encabezado Authorization

El encabezado Authorization debe tener el formato siguiente:

```
11PATHS requestId requestSignature
```

- **requestId** puede ser o bien un applicationId o un userId, dependiendo de la API a la que acceda la petición.
- **11PATHS** es una constante que determina el método de autenticación.
- **applicationId** es un identificador alfanumérico que se obtiene al registrar una aplicación nueva.
- **requestSignature** es una firma derivada de la URL, parámetros, encabezados personalizados y fecha de la solicitud actual, todos ellos con hash utilizando un **Secreto** que también se obtiene mediante el applicationId al registrar la aplicación.

La firma de solicitud es una cadena codificada en Base64 y con firma **HMAC-SHA1**.

La cadena se debe crear siguiendo este proceso:

1. Empezar por una cadena vacía.
2. Anexar el nombre de método en mayúsculas. Actualmente, solo se admiten los valores GET, POST, PUT y DELETE.
3. Anexar un separador de línea, " " (Punto Unicode U+000A).
4. Anexar una cadena con la fecha actual en este formato exacto aaaa-MM-dd HH:mm:ss. Todo en este formato debe explicarse por sí solo, ten

en cuenta que todo es numérico y se debe completar con ceros en caso necesario para que todos los números tengan dos dígitos, excepto el año, que tiene cuatro. Este valor se debe mantener para utilizarlo en el encabezado Date. El proceso de comprobación de firma fallará si no coinciden ambos.

5. Anexar un separador de línea, " " (Punto Unicode U+000A).
6. Serializar todos los encabezados específicos de esta aplicación (no cada encabezado HTTP de la solicitud). Todos los nombres de estos encabezados empiezan por X-11paths-.
    1. Convertir todos los nombres de encabezados a minúsculas.
    2. Ordenar los encabezados por nombre de encabezado en orden alfabético ascendente.
    3. Para cada valor de encabezado, convierte los encabezados de varias líneas en una única línea sustituyendo los caracteres de línea nueva " " por espacios sencillos " ".
    4. Crear una cadena vacía. A continuación, para cada encabezado después de la ordenación y transformaciones anteriores, añadir a la cadena recién creada el nombre de encabezado seguido de dos puntos ":" y del valor de encabezado. Cada nombre:valor debe estar separado del siguiente por un espacio sencillo " ".
    5. Recortar la cadena para asegurarse de que no contenga ningún carácter de espacio al inicio o al final.
7. Anexar un separador de línea, " " (Punto Unicode U+000A).
8. Anexa la cadena de consulta con codificación URL que consta de la ruta (empezando por la primera barra inclinada) y los parámetros de consulta. La cadena de consulta no debe contener el nombre de host o el puerto y no debe contener ningún carácter de espacio como prefijo o sufijo.
9. Solamente para peticiones POST o PUT, anexar un separador de línea, " " (Punto Unicode U+000A).
10. Solamente para peticiones POST o PUT, serializar los parámetros de la petición de la siguiente forma, siendo el nombre del parámetro y su valor, la representación en UTF-8 de su codificación URL.

1. Ordenar los parámetros por nombre de parámetro en orden alfabético ascendente y a continuación por valor de parámetro.

2. Crear una cadena vacía. A continuación, para cada parámetro después de la ordenación, añadir a la cadena recién creada el nombre de parámetro seguido de un signo igual "=" y del valor del parámetro. Cada nombre=valor debe estar separado del siguiente por un ampersand "&".

3. Recortar la cadena para asegurarse de que no contenga ningún carácter de espacio al inicio o al final.

Una vez que se haya creado la cadena siguiendo el proceso descrito más arriba, se debe firmar mediante el algoritmo **HMAC-SHA1** y el secreto obtenido al registrar la aplicación. Después de la firma, los datos binarios sin procesar se deben codificar en base64. La cadena resultante es la cadena **requestSignature** que añadir al encabezado **Authorization**.

1.1.2. El encabezado X-11Paths-Date

El encabezado **X-11Paths-Date** contiene el valor de la fecha UTC actual y debe tener el siguiente formato:

yyyy-MM-dd HH:mm:ss

donde yyyy es el año, MM es el número del mes, dd es el número del día, HH es la hora en formato de 24 horas, mm son los minutos y ss los segundos. Todos los valores se deben completar con ceros para que tengan valores de dos dígitos, excepto el año, que tiene cuatro.

Es muy importante que el valor y el formato de este encabezado sean exactamente los mismos que los utilizados en el proceso de creación de **requestSignature** para el encabezado de autorización como se explica más arriba.

Ten en cuenta que puedes seguir utilizando el encabezado HTTP **Date** estándar en el formato que desees, por ejemplo RFC 1123. Solo asegúrate de no confundir ambos y de utilizar siempre el valor que utilices en **X-11Paths-Date** en el proceso de firma. La API ignorará el encabezado **Date** estándar.

Errores de Autenticación

La siguiente es una lista de condiciones de error posibles que pueden ocurrir durante la autenticación:

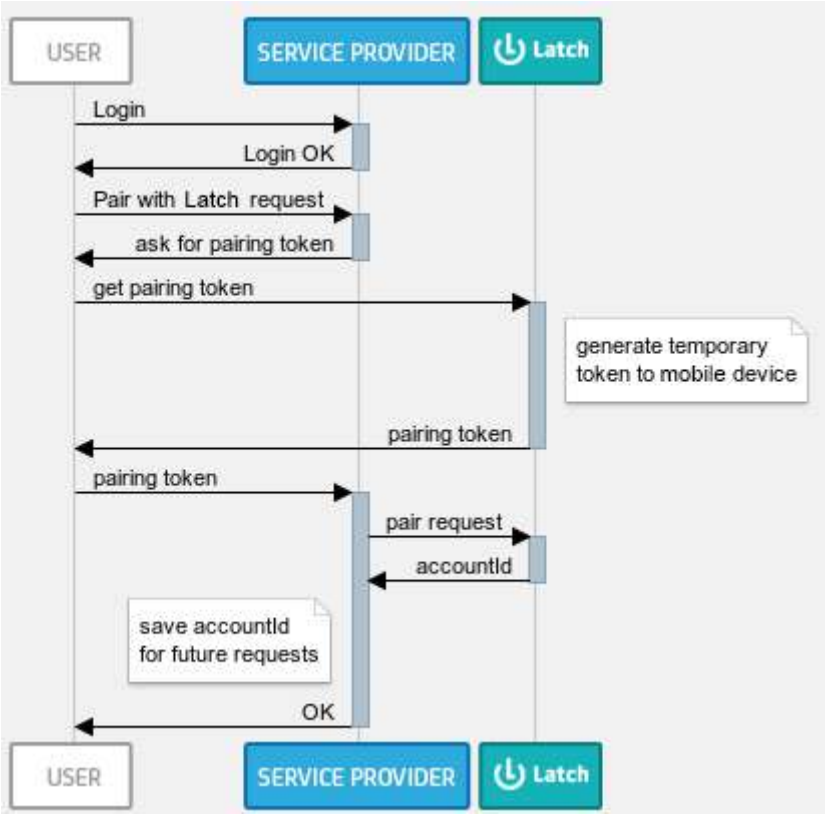
- **Error 101**: Invalid Authorization header format
- **Error 102**: Invalid application signature
- **Error 103**: Authorization header missing
- **Error 104**: Date header missing
- **Error 108**: Invalid date format
- **Error 109**: Request expired, date is too old
- **Error 111**: User not authorized
- **Error 112**: Invalid user signature
- **Error 113**: Secret signing this request is not authorized to perform this operation

1.2. API Aplicación

En esta sección se detalla la información relacionada con la interacción de la aplicación con la API de Latch.

1.2.1. Parear Cuenta

En esta página se describe el método preferido para parear cuentas mediante un **token de pareado** que los usuarios pueden obtener a través de las aplicaciones móviles.



Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de Autenticación.

Solicitud

Para parear con la API de token, lleva a cabo una solicitud HTTP GET en el siguiente extremo:

```
https://latch.tu.com/api/2.0/pair/{token}
```

donde **token** es un token alfanumérico de seis caracteres de longitud que se debe obtener del usuario que, a su vez, lo ha obtenido mediante la app móvil.

Adicionalmente (y de manera totalmente opcional) puedes incluir un parámetro commonName donde incluyas un identificador del usuario en tu sistema. Este identificador se mostrará en la herramienta de soporte (LST) y te ayudará a identificar al usuario de una manera más sencilla.

```
https://latch.tu.com/api/2.0/pair/{token}?commonName=YOUR_COMMON_NAME
```

Respuesta

Si todo va bien, la API devolverá una respuesta con este aspecto:

```
{"data":{"accountId":"..."}}
```

donde accountId es un token alfanumérico de 64 caracteres de longitud que se puede utilizar para identificar de forma única la cuenta que se está pareando con esta aplicación. **Este valor se debe almacenar** para utilizarlo posteriormente en llamadas siguientes tales como estado o anular pareado.

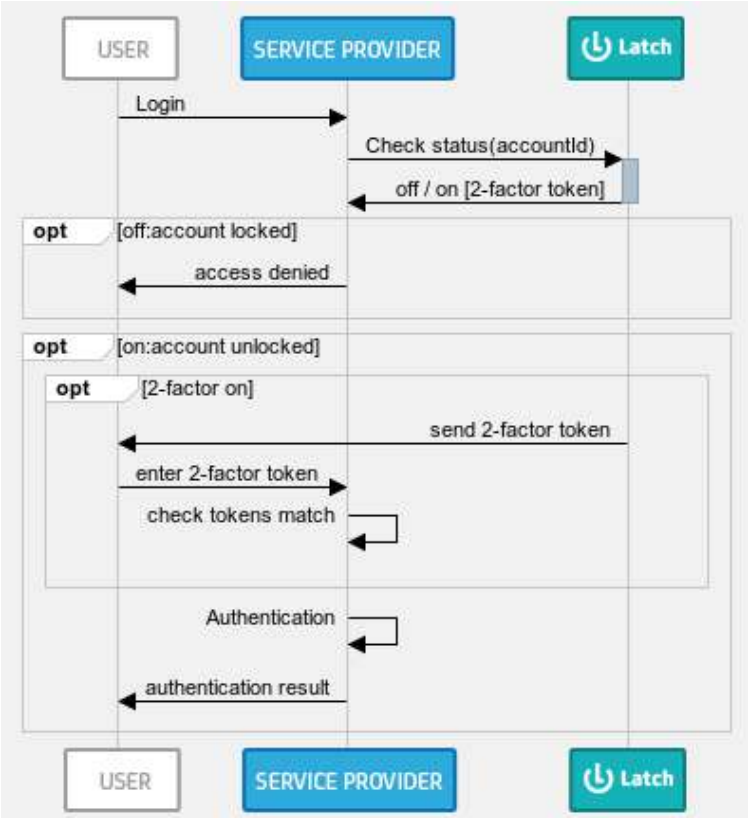
Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 205**: Account and application already paired
  Se devolverá este error cuando un usuario cuya cuenta ya esté pareada con esta aplicación haya generado el token especificado.
- **Error 206**: Pairing token not found or expired
  La causa más probable de este error es el uso de un token caducado (los tokens se deben utilizar hasta 60 segundos después de su creación). Otra razón obvia para este error es que la solicitud incluya un token inventado.
- **Error 401**: Missing parameter in API call
  La causa de este error es que falta o está vacío algún parámetro requerido.
- **Error 406**: Invalid parameter length
  Se devolverá este error cuando el parámetro commonName tenga una longitud mayor a 100 caracteres.

1.2.2. Estado de la Cuenta

En esta página se describe cómo comprobar el estado de una cuenta y las distintas respuestas posibles.



Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de Autenticación.

Solicitud

Para utilizar la API de estatus, lleva a cabo una solicitud HTTP GET en el siguiente extremo:

```
https://latch.tu.com/api/2.0/status/{accountId}
```

donde accountId es un token alfanumérico de 64 caracteres de longitud que se puede utilizar para identificar de forma única la cuenta cuyo estado se está consultando, obtenido mediante el método de pareado.

Si has creado una o más operaciones para ser controladas por **Latches** individuales, puedes consultar el estado de cada operación usando su identificador de operación:

```
https://latch.tu.com/api/2.0/status/{accountId}/op/{operationId}
```

IMPORTANTE: Si has creado operaciones para una aplicación, **NO** deberías necesitar consultar el estado usando el ID de aplicación. Igualmente ocurre si tienes operaciones anidadas. Una vez que una operación tiene operaciones hijas, el ID de aplicación de la operación padre no debería ser usado para llamadas de estado.
Nada te impide hacerlo, y funcionará, pero en las apps móviles, las aplicaciones con operaciones o las operaciones con hijos, no tienen las opciones de OTP, opción de Programar o Autobloqueo. En vez de eso, actúan como un latch maestro sobre todas las operaciones hijas, permitiendo bloquearlas o desbloquearlas todas con una sola acción.

Opcionalmente, si se desea consultar el estado:

- *Eliminando* cualquier contraseña de un solo uso en la respuesta, añadiendo la terminación "/nootp" a cualquiera de los dos extremos.
- *Sin enviar* las notificaciones push a las aplicaciones móviles, añadiendo la terminación "/silent" a cualquiera de los dos extremos.

```
https://latch.tu.com/api/2.0/status/{accountId}/nootp

https://latch.tu.com/api/2.0/status/{accountId}/silent

https://latch.tu.com/api/2.0/status/{accountId}/nootp/silent

https://latch.tu.com/api/2.0/status/{accountId}/op/{operationId}
/nootp/silent
```

Respuesta

Si todo va bien, la API devolverá una respuesta JSON, que tendrá este aspecto:

```
{"data":{"operations":{"applicationId":{"status":"...",
"two_factor":{...}, "operations":{...}}}}}
```

Hay dos valores posibles para el campo de **status**: on y off.

Este campo two_factor es opcional y cuando está presente, tiene el formato siguiente:

```
"two_factor":{"token":"...","generated":...}
```

donde **token** es un valor alfanumérico de seis caracteres de longitud que también se enviará al dispositivo móvil del usuario y que se debe comprobar para completar la operación. El campo **generated** es la marca horaria en la que se generó el token y se puede utilizar para controlar la ventana de tiempo durante la que se puede validar el token.

El objeto operations, si está presente, contiene las **operaciones secundarias** y está definido recursivamente por los mismos campos: status, two_factor y operations.

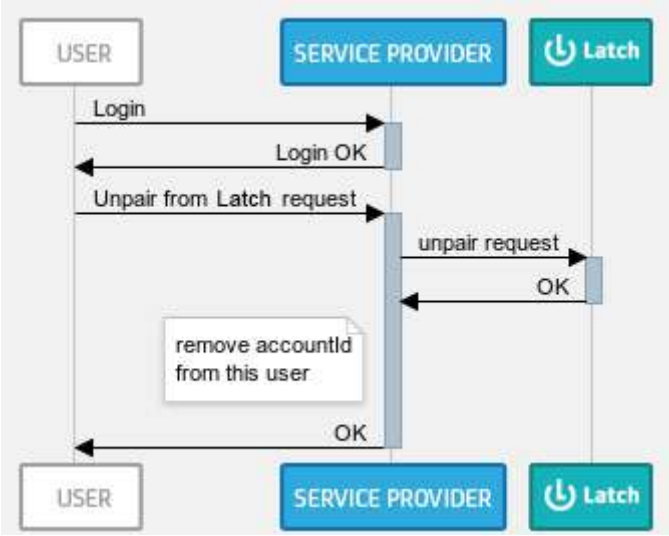
Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 201**: Account not paired
  Se produce cuando el identificador de cuenta es erróneo o no está pareado actualmente con la aplicación que solicita la comprobación de estado.
- **Error 401**: Missing parameter in API call
- **Error 704**: Silent notification is not applied due to subscription limits.

1.2.3. Desparear Cuenta

En esta página se describe cómo desparear una cuenta de Latch.



Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de Autenticación.

Solicitud

Para anular el pareado, lleva a cabo una solicitud HTTP GET en el siguiente extremo:

```
https://latch.tu.com/api/2.0/unpair/{accountId}
```

donde accountId es un token alfanumérico de 64 caracteres de longitud que se puede utilizar para identificar de forma única la cuenta cuyo pareado se está anulando, obtenido mediante el método de pareado.

Respuesta

Si todo va bien, la API devolverá una respuesta JSON vacía, que tendrá este aspecto:

```
{}
```

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 201**: Account not paired (API > 0.6)
- **Error 204**: Error unpairing account (API <= 0.6)
  Este error se produce cuando el identificador de cuenta es erróneo o no está pareado actualmente con la aplicación que solicita la anulación de pareado.
- **Error 401**: Missing parameter in API call

1.2.4. Modificar Estado

En esta página se describe cómo modificar el estado de una cuenta de Latch.

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de Autenticación.

Solicitud

Para utilizar la API de modificaciones externas, lleva a cabo una solicitud HTTP POST en los siguientes extremos en función de si se trata de bloquear (lock) o desbloquear (unlock) el estado de un Latch para una aplicación u operación:

```
https://latch.tu.com/api/2.0/lock/{accountId}
https://latch.tu.com/api/2.0/unlock/{accountId}
```

donde accountId es un token alfanumérico de 64 caracteres de longitud que se puede utilizar para identificar de forma única la cuenta cuyo estado se está modificando, obtenido mediante el método de pareado.

Si has creado una o más operaciones para ser controladas por Latches individuales, puedes modificar el estado de cada operación usando su identificador de operación:

```
https://latch.tu.com/api/2.0/lock/{accountId}/op/{operationId}
https://latch.tu.com/api/2.0/unlock/{accountId}/op/{operationId}
```

Respuesta

Si todo va bien, la API devolverá una respuesta JSON vacía, que tendrá este aspecto:

```
{}
```

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 111**: User not authorized
  Se produce cuando el plan de subscripción actual del usuario no incluye las herramientas de soporte.
- **Error 201**: Account not paired
  Se produce cuando el identificador de cuenta es erróneo o no está pareado actualmente con la aplicación que solicita la comprobación de estado.
- **Error 401**: Missing parameter in API call

1.2.5. Gestionar Operaciones

En esta página se describe cómo gestionar las **operaciones** de una aplicación, permitiendo añadir, actualizar o eliminar operaciones.

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de Autenticación.

Solicitudes

Añadir una operación

Para añadir una operación, lleva a cabo una solicitud HTTP PUT en el siguiente extremo:

`https://latch.tu.com/api/2.0/operation`

Parámetros de la petición:

- parentId: Identificador de la aplicación u operación a la que deseamos añadir la nueva operación.
- name: Nuevo nombre para la operación.
- two_factor (Opcional): Valor para el segundo factor de autenticación. Valores posibles: MANDATORY, OPT_IN, DISABLED (Valor por defecto: DISABLED).
- lock_on_request (Opcional): Valor para el bloqueo tras consultar. Valores posibles: MANDATORY, OPT_IN, DISABLED (Valor por defecto: DISABLED).

Respuesta

Si todo va bien, la API devolverá una respuesta JSON que tendrá este aspecto, siendo operationId el identificador de la operación recién creada:
{data:{"operationId":"..."}}

Modificar una operación

Para modificar una operación, lleva a cabo una solicitud HTTP POST en el siguiente extremo:

```
https://latch.tu.com/api/2.0/operation/{operationId}
```

Siendo 'operationId' el identificador de dicha operación.

Parámetros de la petición:

- name: Nuevo nombre para la operación.
- two_factor (Opcional): (Opcional): Valor para el segundo factor de autenticación.
  Valores posibles: MANDATORY, OPT_IN, DISABLED. Valor por defecto DISABLED.
- lock_on_request (Opcional): Valor para el bloqueo tras consultar.
  Valores posibles: MANDATORY, OPT_IN, DISABLED. Valor por defecto DISABLED.

Respuesta

Si todo va bien, la API devolverá una respuesta JSON vacía:

```
{}
```

Eliminar una operación

Para eliminar una operación, lleva a cabo una solicitud HTTP DELETE en el siguiente extremo:
https://latch.tu.com/api/2.0/operation/{operationId}

Siendo 'operationId' el identificador de dicha operación.

Respuesta

Si todo va bien, la API devolverá una respuesta json vacía, que tendrá este aspecto:

```
{}
```

Ver tus operaciones

Para ver tus operaciones, lleva a cabo una solicitud HTTP GET en el siguiente extremo:

```
https://latch.tu.com/api/2.0/operation
https://latch.tu.com/api/2.0/operation/{operationId}
```

Respuesta

Si todo va bien, la API devolverá una respuesta con este aspecto:

```
{"data":{"operations":{"operationId":{"name":"...","two_factor":
"...","lock_on_request":"...","operations":{...}}}}}
```

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 301**: Application or Operation not found
  Se produce cuando el identificador de operación utilizado no se corresponde con un identificador válido.
- **Error 401**: Missing parameter in API call
- **Error 703**: Application or Operation not created due to subscription limits
  Se produce cuando se intenta crear una operación fuera de los límites de la subscripción.

1.2.6. Historial de Usuario

En esta página se describe cómo consultar el historial de una cuenta.

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de Autenticación.

Solicitud

Para utilizar la API de historial, lleva a cabo una solicitud HTTP GET en el siguiente extremo:

```
https://latch.tu.com/api/2.0/history/{accountId}/{from}/{to}
```

donde **accountId** es un token alfanumérico de 64 caracteres de longitud que se puede utilizar para identificar de forma única la cuenta cuyo histórico se está consultando, que se ha obtenido mediante el método de pareado.

los párametros **from** y **to**, que son opcionales, permiten filtrar el historial entre dos fechas. Ambos parámetros son numéricos y se corresponden con la cantidad de milisegundos transcurridos desde el 1 de Enero de 1970 00:00:00 GMT. (Epoch Time).

Respuesta

Si todo va bien, la API devolverá una respuesta JSON, con este aspecto:

```
{ "data": { "applicationId": { ... }, "count": ...,
"clientVersion": { ... } }, "lastSeen": ..., "history": [ { "t":
..., "action": "...", "what": "...", "value": "...", "was":
"...", "name": "...", "userAgent": "...", "ip": "..." } ] } }
```

Donde la información contenida es:

- applicationId: información relativa a la aplicación y sus operaciones.
- count: cantidad de elementos contenidos en el array history.

- clientVersion: información relativa a los clientes móviles utilizados por el usuario.
- lastSeen: tiempo en que se registró la última actividad del usuario.
- history: array que contiene las acciones realizadas en la cuenta, bien por el usuario, bien por el desarrollador.
    - t: tiempo en que se realizó dicha acción.
    - action: si se trata de una acción realizada por el usuario (USER_UPDATE), por el desarrollador (DEVELOPER_UPDATE) o una consulta de estado (get).
    - what: el parámetro sobre el que realizó la acción (status, two_factor, from...).
    - was: valor anterior de la acción si se trata de una modificación.
    - value: valor actual de la acción.
    - name: nombre de la aplicación u operación sobre la que se realizó la acción.
    - ip / userAgent: IP y User-Agent de quien realizó la acción.

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 111**: User not authorized
  Cuando el plan de subscripción actual del usuario no incluye las herramientas de soporte.
- **Error 201**: Account not paired
  Este error se produce cuando el identificador de cuenta es erróneo o no está pareado actualmente con la aplicación que solicita la comprobación de estado.
- **Error 401**: Missing parameter in API call

- **Error 405**: History response is limited to 1000 entries for the selected date range
  Este error no fatal se produce cuando la respuesta debería contener más elementos del máximo que se devuelven (1000).

1.2.7. Gestionar Instancias

En esta página se describe cómo gestionar las **instancias** de una aplicación u operación asociadas a un accountId concreto, así como la forma en que se puede interactuar con ellas.

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de autenticación.

Solicitudes

Añadir una instancia

Para añadir una instancia a una aplicación, lleva a cabo una solicitud HTTP PUT en el siguiente extremo:

```
https://latch.tu.com/api/2.0/instance/{accountId}
```

Para añadir instancias a una operación, lleva a cabo una solicitud HTTP PUT en el siguiente extremo:

```
https://latch.tu.com/api/2.0/instance/{accountId}/op/{operationId}
```

donde accountId es un token alfanumérico de 64 caracteres de longitud que se puede utilizar para identificar de forma única la cuenta cuyas instancias se estan modificando. Se habrá obtenido previamente mediante el método de pareado.

Parámetros de la petición:

- instances: El nombre de las instancias a crear en formato instances=nombre, se pueden crear varias simultáneamente separando por &.

Ejemplo: instances=Name_1&instances=Name_2&instances=Name_N

Respuesta

Si todo va bien, la API devolverá una respuesta json que tendrá este aspecto, siendo los distintos 'instancesId' los identificadores de las instancias recién creadas que se deberán guardar para interactuar con ellas:

```
{"data":{"instances":{"instanceId_1":"Name_1","instanceId_2":"Name_2","instanceId_N":"Name_N", ...}}}
```

Eliminar una instancia

Para eliminar una instancia de una aplicación, lleva a cabo una solicitud HTTP DELETE en el siguiente extremo:

```
https://latch.tu.com/api/2.0/instance/{accountId}/i/{instanceId}
```

Para eliminar una instancia de una operación, lleva a cabo una solicitud HTTP DELETE en el siguiente extremo:

```
https://latch.tu.com/api/2.0/instance/{accountId}/op/{operationId}/i/{instanceId}
```

Siendo 'instanceId' el identificador de dicha instancia y 'operationId' el identificador de la operación a la que pertenece

Respuesta

Si todo va bien, la API devolverá una respuesta json vacía, que tendrá este aspecto:

```
{}
```

Ver instancias

Para obtener las instancias de una aplicación, lleva a cabo una solicitud HTTP GET en el siguiente extremo:

```
https://latch.tu.com/api/2.0/instance/{accountId}
```

Para obtener las instancias de una operación, lleva a cabo una solicitud HTTP GET en el siguiente extremo:

```
https://latch.tu.com/api/2.0/instance/{accountId}/op/{operationId}
```

Siendo 'instanceId' el identificador de dicha instancia y 'operationId' el identificador de la operación a la que pertenece

Respuesta

Si todo va bien, la API devolverá una respuesta con este aspecto:

```
{"data": {"instanceId_1": {"name": "MyInstance1", "two_factor": "MANDATORY|DISABLED|OPT_IN", "lock_on_request": "MANDATORY|DISABLED|OPT_IN"}, "instanceId_2": {"name": "...", "two_factor": "...", "lock_on_request": "..."}}}
```

Editar una instancia

Para editar una instancia de una aplicación, lleva a cabo una solicitud HTTP POST en el siguiente extremo:

```
https://latch.tu.com/api/2.0/instance/{accountId}/i/{instanceId}
```

Para editar una instancia de una operación, lleva a cabo una solicitud HTTP POST en el siguiente extremo:

```
https://latch.tu.com/api/2.0/instance/{accountId}/op/{operationId}/i/{instanceId}
```

Siendo 'instanceId' el identificador de dicha instancia y 'operationId' el identificador de la operación a la que pertenece

Parámetros de la petición:

- name: (Optional) Nuevo nombre para la instancia.
- two_factor (Opcional): Valor para el segundo factor de autenticación. Valores posibles: MANDATORY, OPT_IN, DISABLED. Valor por defecto DISABLED.
- lock_on_request (Opcional): Valor para el bloqueo tras consultar. Valores posibles: MANDATORY, OPT_IN, DISABLED. Valor por defecto DISABLED.

Respuesta

Si todo va bien, la API devolverá una respuesta json vacía, que tendrá este aspecto:

```
{}
```

Consultar el estado de una instancia

Para consultar el estado de una instancia, lleva a cabo una solicitud HTTP GET en uno de los siguientes extremos (para instancias de una aplicación o instancias de una operación):

```
https://latch.tu.com/api/2.0/status/{accountId}/i/{instanceId}
```

```
https://latch.tu.com/api/2.0/status/{accountId}/op/{operationId}/i/{instanceId}
```

Tanto la respuesta como el resto de opciones adicionales como 'silent' o 'nootp' están detalladas en la sección 'Estado de la cuenta'.

Modificar el estado de una instancia

Para modificar el estado de una instancia, lleva a cabo una solicitud HTTP POST en uno de los siguientes extremos (para instancias de una aplicación o instancias de una operación):

```
https://latch.tu.com/api/2.0/lock/{accountId}/i/{instanceId}
```

```
https://latch.tu.com/api/2.0/lock/{accountId}/op/{operationId}/i
/{instanceId}
```

```
https://latch.tu.com/api/2.0/unlock/{accountId}/i/{instanceId}
```

```
https://latch.tu.com/api/2.0/unlock/{accountId}/op/{operationId}
/i/{instanceId}
```

Tanto la respuesta como información adicional puede consultarse en la sección 'Modificar estado'

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 301:** Application or Operation not found. Este error se produce cuando el identificador de operación utilizado no se corresponde con un identificador de operación válido.
- **Error 302:** Instance not found. Este error se produce cuando el identificador de instancia utilizado no se corresponde con un identificador de instancia válido.
- **Error 401:** Missing parameter in API call
- Error 703: Application or Operation not created due to subscription limits. Este error se produce cuando se intenta crear una instancia fuera de los límites de la subscripción.

1.2.8. Editar CommonName

En esta página se describe cómo editar un nombre común para un usuario pareado.

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de Autenticación.

Solicitud

Definir un nombre común

Para establecer un nombre común, realice una solicitud HTTP POST en el siguiente endpoint:

```
https://latch.tu.com/api/3.0/commonName/{accountId}/
```

Parámetros de la petición:

- commonName: Identificador del proveedor de servicio para el usuario emparejado

Respuesta

Si todo va bien, la API devolverá una respuesta JSON vacía, que tendrá este aspecto:

```
{}
```

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- Error 201: Account not paired
  Este error se produce cuando el identificador de cuenta es erróneo o no está pareado actualmente con la aplicación que solicita la comprobación de estado.
- Error 401: Missing parameter in API call
  La causa de este error es que falta o está vacío algún parámetro requerido.
- Error 406: Invalid parameter length
  Se devolverá este error cuando el parámetro commonName tenga una longitud mayor a 100 carácteres.

1.3. API Usuario

1.3.1. Gestionar Aplicaciones

En esta página se describe cómo gestionar las **aplicaciones** de un usuario, permitiendo añadir, actualizar o eliminar aplicaciones.

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu userId y firmarlo como se explica en la sección de Autenticación.

Solicitudes

Añadir una aplicación

Para añadir una aplicación, lleva a cabo una solicitud HTTP PUT en el siguiente extremo:

```
https://latch.tu.com/api/2.0/application
```

Parámetros de la petición:

- name: El nuevo nombre para la aplicación.
- contactEmail: Email de contacto para la aplicación.
- contactPhone: Télefono de contacto para la aplicación.
- two_factor (Opcional): Valor para el segundo factor de autenticación.
- Valores posibles: MANDATORY, OPT_IN, DISABLED. Valor por defecto DISABLED.
- lock_on_request (Opcional): Valor para el bloqueo tras consultar. Valores posibles: MANDATORY, OPT_IN, DISABLED. Valor por defecto DISABLED.

Respuesta

Si todo va bien, la API devolverá una respuesta json que tendrá este aspecto, siendo 'applicationId' el identificador de la aplicación recién creada:

```
{data:{"applicationId":"...", "secret":"..."}}
```

Modificar una aplicación

Para modificar una aplicación, lleva a cabo una solicitud HTTP POST en el siguiente extremo:

```
https://latch.tu.com/api/2.0/application/{applicationId}
```

Siendo 'applicationId' el identificador de dicha aplicación.

Parámetros de la petición:

- name: El nuevo nombre para la aplicación.
- contactEmail: Email de contacto para la aplicación.
- contactPhone: Télefono de contacto para la aplicación.
- two_factor (Opcional): Valor para el segundo factor de autenticación.
- Valores posibles: MANDATORY, OPT_IN, DISABLED. Valor por defecto DISABLED.
- lock_on_request (Opcional): Valor para el bloqueo tras consultar. Valores posibles: MANDATORY, OPT_IN, DISABLED. Valor por defecto DISABLED.

Respuesta

Si todo va bien, la API devolverá una respuesta json vacía, que tendrá este aspecto:

```
{}
```

Eliminar una aplicación

Para eliminar una aplicación, lleva a cabo una solicitud HTTP DELETE en el siguiente extremo:

```
https://latch.tu.com/api/2.0/application/{applicationId}
```

Siendo 'applicationId' el identificador de dicha aplicación.

Respuesta

Si todo va bien, la API devolverá una respuesta json vacía, que tendrá este aspecto:

```
{}
```

Ver tus aplicaciones

Para ver tus aplicaciones, lleva a cabo una solicitud HTTP GET en el siguiente extremo:

```
https://latch.tu.com/api/2.0/application
```

Respuesta

Si todo va bien, la API devolverá una respuesta con este aspecto:

```
{"data":{"operations":{"applicationId":{"name":"...","two_factor
":"...","lock_on_request":"...","operations":{...}}}}}
```

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 111:** User not authorized
  Este error se produce cuando el plan de subscripción actual del usuario no incluye la API de usuario
- **Error 301:** Application or Operation not found
  Este error se produce cuando el identificador de aplicación utilizado no se corresponde con un identificador de aplicación válido.
- **Error 401:** Missing parameter in API call
- **Error 402:** Invalid parameter value
  Este error se produce cuando algunos parámetros tienen un formato incorrecto, como direcciones de correo, números de teléfono o nombres.
- **Error 703:** Application or Operation not created due to subscription limits

Este error se produce cuando se intenta crear una aplicación fuera de los límites de la subscripción.

1.3.2. Subscripción Desarrollador

En esta página se describe como obtener la información de subscripción de un usuario.

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu userId y firmarlo como se explica en la sección de autenticación.

Solicitudes

Ver subscripción

Para obtener la información de subscripción, lleva a cabo una solicitud HTTP GET en el siguiente extremo:

```
https://latch.tu.com/api/2.0/subscription
```

Respuesta

Si todo va bien, la API devolverá una respuesta json, que tendrá este aspecto:

```
{"data":{"subscription":{"id":"subcriptionName","applications":{
"inUse":X,"limit":X},"operations":{"appName1":{"inUse":X,"limit"
:X},"users":{"inUse":X,"limit":X}}}}
```

donde la información contenida es:

subscriptionName: El nombre de la subscripción actual.

inUse: La cantidad de aplicaciones, usuarios u operaciones usandose actualmente.

limit: La máxima cantidad permitida para el plan de subscripción actual. -1 para ilimitado.

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

Error 111: User not authorized
Este error se produce cuando el plan de subscripción actual del usuario no incluye la API de usuario

1.4. Servidor de TOTP

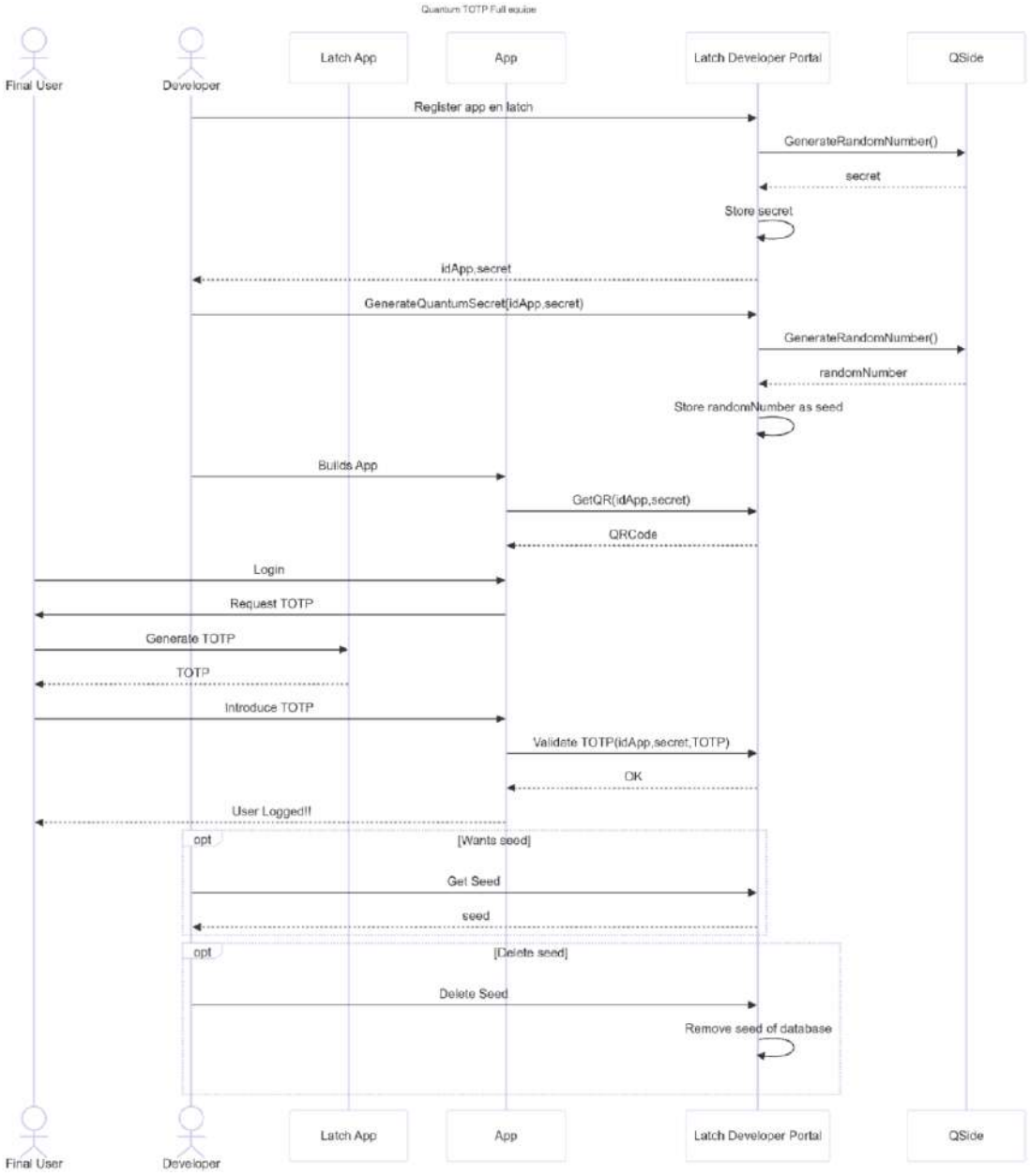
1.4.1. Crear TOTP

Este servicio actúa como un servidor TOTP (Time-based One-Time Password) que permite crear, gestionar y validar contraseñas de un solo uso basadas en el tiempo.

Proporciona una API para crear y gestionar estos TOTPs y secretos asociados a distintos usuarios. Las principales funcionalidades incluyen:

- **Crear un TOTP:** Permite a los usuarios crear un TOTP para su autenticación.
- **Eliminar un TOTP:** Los TOTPs creados pueden ser eliminados cuando ya no se necesiten.
- **Obtener información de un TOTP:** Puedes consultar la información de un TOTP específico.
- **Validar un TOTP:** Permite validar un código TOTP proporcionado por un usuario.

La secuencia de las operaciones y su utilización se representa en el siguiente diagrama:



Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de Autenticación.

Solicitud

Crea un nuevo TOTP para un usuario dado. Se requiere userId (identificador del usuario) y commonName (nombre común del TOTP)

`POST https://latch.tu.com/api/3.0/totps`

Parámetros

- userId: Identificador que se le asociará al usuario (dependiente del proveedor)
- commonName: Nombre a asociar al usuario (dependiente del proveedor)

Respuesta

Si todo va bien, el API responderá con la información relativa al TOTP:

```
{
    "data": {
        "totpId": "identificador del TOTP",
        "secret": "secreto del TOTP",
        "appId": "identificador de la app de Latch",
        "identity": {
            "id": "identificador del usuario",
            "name": "nombre del usuario"
        },
        "issuer": "nombre de la app",
        "algorithm": "algoritmo del TOTP",
        "digits": "dígitos del TOTP",
        "period": "período del TOTP",
        "createdAt": "fecha de creación",
        "qr": "imagen del código QR en base64",
        "uri": "URI del TOTP, formato otpauth"
    }
}
```

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 301:** Application or Operation not found
  Este error se produce cuando el identificador de aplicación utilizado no se corresponde con un identificador de aplicación válido.
- **Error 304:** No totp server configured
  No existe un TOTP Server configurado para esta app.
- **Error 401:** Missing parameter in API call
  Parámetros inválidos.
- **Error 307:** Unable to generate totp
  No se ha podido generar el secreto para el TOTP. Error interno.

1.4.2. Eliminar TOTP

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de autenticación.

Solicitud

Elimina un TOTP específico utilizando su identificador totpId.

```
DELETE https://latch.tu.com/api/3.0/totps/{totpId}
```

Parámetros

- totpId: Identificador del TOTP que se desea eliminar

Respuesta

Si todo va bien, se devolverá un **HTTP RESPONSE 204**.

Errores

- **Error 304**: No totp server configured
  No existe un TOTP Server configurado para esta app.
- **Error 305**: App totp not found
  TOTP no encontrado.

1.4.3. Obtener TOTP

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de autenticación.

Solicitud

Devuelve la información de un TOTP específico basado en su totpId.

```
GET https://latch.tu.com/api/3.0/totps/{totpId}
```

Parámetros

- totpId: Identificador del TOTP que se desea obtener

Respuesta

Si todo va bien, el API responderá con la información relativa al TOTP:

```
{
  "data": {
    "totpId": "identificador del TOTP",
    "secret": "secreto del TOTP",
    "appId": "identificador de la app de Latch",
```

```
    "identity": {
      "id": "identificador del usuario",
      "name": "nombre del usuario"
    },
    "issuer": "nombre de la app",
    "algorithm": "algoritmo del TOTP",
    "digits": "dígitos del TOTP",
    "period": "período del TOTP",
    "createdAt": "fecha de creación",
    "qr": "imagen del código QR en base64",
    "uri": "URI del TOTP, formato otpauth"
  }
}
```

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 304:** No totp server configured
  No existe un TOTP Server configurado para esta app.
- **Error 305:** App totp not found
  TOTP no encontrado.

1.4.4. Validar TOTP

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de autenticación.

Solicitud

Valida un código TOTP específico proporcionado por el usuario.
El totpId identifica el TOTP y code es el código generado que se quiere validar.

```
POST https://latch.tu.com/api/3.0/totps/{totpId}/validate
```

Parámetros

- totpId: Identificador del TOTP que se utilizará para validar el algoritmo, sobre el código enviado
- code: Código enviado por el usuario final, para validar mediante el algoritmo especificado por el totpId, y que se desea validar

Respuesta

En ambos casos, se responderá con un HTTP Code de 200, de acuerdo a:

Código correcto:

```
{}
```

Código incorrecto:

```
{"error": {
 "code": 306,
 "message": "Invalid totp code"
   }
}
```

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 304:** No totp server configured
  No existe un TOTP Server configurado para esta app.

- **Error 305:** App totp not found
  TOTP no encontrado.
- **Error 306:** Invalid totp code
  Código no válido según los parámetros almacenados y el timestamp actual.
- **Error 307:** Unable to generate totp
  Error al conseguir los parámetros de generación de TOTPs.
- **Error 401:** Missing parameter in API call
  Alguno de los parámetros de entrada no existen.
- **Error 402:** Invalid parameter value
  Parámetros inválidos.
- **Error 708:** TOTP not validated due to subscription limits
  TOTP no válido por límites en la suscripción del developer

1.5. Controles de Autorización

Introducción

La funcionalidad de **Controles de Autorización** en el producto **Latch** permite establecer un mecanismo para gestionar la aprobación de acciones dentro de un sistema, basado en la intervención de uno o más usuarios. Esta característica es útil en escenarios donde es necesario contar con la aprobación explícita de ciertos usuarios antes de permitir el avance de un proceso o flujo.

Creación de un Control de Autorización

La creación de un control de autorización se realiza a través de la herramienta llamada **LST**. Desde LST, los administradores o usuarios con permisos pueden definir una nueva regla o control que gestione las autorizaciones sobre un grupo específico de usuarios.

Pasos para crear un control de autorización

1. Acceder a la herramienta **LST**.
2. Seleccionar la opción para crear un nuevo **Control de Autorización**.
3. Especificar los usuarios involucrados en el control.
4. Definir si el control debe basarse en:

    - **Aprobación Unánime (TODOS)**: El flujo o acción solo se permitirá si todos los usuarios seleccionados aprueban la solicitud.
    - **Aprobación Parcial (ALGUNO)**: El flujo o acción se permitirá si al menos uno de los usuarios seleccionados aprueba la solicitud.

5. Guardar el control de autorización.

Identificación del Control de Autorización

Cada control de autorización creado posee un identificador único denominado controlID. Este controlID es fundamental para interactuar con la funcionalidad a través de la API del sistema, permitiendo su consulta o modificación posterior.

Uso de la API para los Controles de Autorización

Una vez que se ha creado un control de autorización, este puede ser gestionado o consultado a través de las **APIs** de Latch. El controlID será el identificador principal que se utilice para llevar a cabo cualquier operación sobre ese control.

Operaciones disponibles vía API:

- **Consultar un control de autorización**: Permite recuperar los detalles del control, incluyendo los usuarios asociados y las reglas de aprobación (TODOS o ALGUNO).
- **Actualizar un control de autorización**: Modificar los usuarios o las condiciones de aprobación.
- **Eliminar un control de autorización**: Remover un control de autorización si ya no es necesario.

Ejemplo de Escenario de Uso

Supongamos que en un flujo de negocio se requiere la aprobación de un grupo de tres usuarios antes de completar una transacción. Se puede configurar un control de autorización en **Latch** donde:

- Los tres usuarios involucrados son añadidos al control.
- Se define que la aprobación debe ser unánime, es decir, **todos** los usuarios deben aprobar antes de que la transacción sea ejecutada.

En otro caso, si se requiere que solo uno de los tres usuarios pueda aprobar para continuar, se configuraría el control con la opción **alguno**.

Conclusión

La funcionalidad de **Controles de Autorización** en **Latch** ofrece una manera flexible y potente de gestionar aprobaciones de usuarios, permitiendo configuraciones basadas tanto en la participación de todos los usuarios involucrados como en la aprobación por parte de un subconjunto. La integración con **LST** y la exposición a través de API facilita su uso tanto a nivel de interfaz gráfica como de automatización en aplicaciones externas.

Autenticación

Para utilizar esta solicitud tienes que autenticar tu app utilizando tu applicationId y firmarlo como se explica en la sección de autenticación.

Solicitud

Consulta el estado de un control de autorización determinado.

```
GET https://latch.tu.com/api/3.0/control-status/{controlId}
```

Parámetros

- controlId: Identificador del control de autorización a consultar

Respuesta

Si todo va bien, el API responderá con la información relativa al estado del control de autorización `controlId`:

```
{
    "data": {
        "status": "on" | "off"
    }
}
```

Errores

La siguiente es una lista de condiciones de error posibles que pueden ocurrir al intentar utilizar esta API:

- **Error 113:** Secret signing this request is not authorized to perform this operation
  La firma appId/secret no está autorizada para realizar esta operación.
- **Error 301:** Application or Operation not found
  Este error se produce cuando el identificador de operación utilizado no se corresponde con un identificador de operación válido.
- **Error 302:** Instance not found
  Este error se produce cuando el identificador de instancia utilizado no se corresponde con un identificador de instancia válido.
- **Error 1100:** Authorization control not found
  Identificador del control de autorización no encontrado.
- **Error 1104:** Error checking authorization control status
  Error al consultar el control de autorización.

2. Webhooks

Los WebHooks permiten a los desarrolladores recibir notificaciones a tiempo real cuando sus usuarios pareados realicen acciones a través de la app móvil.

Una vez registrada una URI para recibir WebHooks, Latch empezará a enviar peticiones HTTP a dicha URI cada vez que haya cambios en algún usuario

2.1. Añadir un Webhook

Para añadir un nuevo WebHook, ve a la página de tu aplicación en 'Mis Aplicaciones'

A continuación, añade la URI completa de tu WebHook sin incluir query (e.j: https://www.example.com/hook) en la sección "WebHooks". Ten en cuenta que dicha URI necesita ser accesible públicamente en Internet; es decir URIs como '127.0.0.1' o 'localhost' no serán válidas puesto que Latch no será capaz de comunicarse con tu red local.

Una vez que hayas introducido una URI para un WebHook, se enviará una petición de verificación a dicha URI. Esta verificación es una petición HTTP GET que incluye un parámetro 'challenge'. Tu WebHook debe responder un eco con dicho parámetro para poder ser verificado. Esta verificación se realiza para comprobar que realmente tu aplicación quiere recibir notificaciones en dicha URI. Si accidentalmente introduces una URI errónea (o si alguien de forma maliciosa introduce tu servidor como WebHook), tu aplicación no responderá correctamente al desafío y Latch no enviará ninguna notificación

Ejemplo:

```
GET https://www.example.com/hook?challenge=ABC123
```

Si tu WebHook responde correctamente al desafío anterior; Latch empezará a enviar notificaciones a la URI de tu WebHook con los cambios de tus usuarios, (véase la pestaña 'Notificaciones de WebHooks' para más información). Por otro lado si tu aplicación no responde correctamente, verás un mensaje de error indicando el problema encontrado.

NOTA: Tu URI sólo dispone de diez segundos para responder a la notificación antes de que se produzca un 'timeout'.

2.2. Notificaciones Webhook

Cuando una URI de WebHook es añadida, empezarás a recibir "peticiones de notificación" cada vez que uno de tus usuarios haga algún cambio en tu aplicación.

Una petición de notificación es una petición HTTP POST que incluye un JSON en su cuerpo. **Ten en cuenta que múltiples cambios simultáneos para múltiples usuarios pueden incluirse en la misma notificación.** Puedes encontrar el formato JSON específico para cada tipo de notificación debajo.

Firma de notificaciones

Cada petición de notificación incluirá una cabecera HTTP X-11paths-Authorization formada por la firma HMAC-SHA1 del cuerpo de la petición, (se utiliza el secreto de tu aplicación como la clave de firmado). Esto permitirá verificar a tu aplicación que la notificación proviene de Latch. Es aconsejable que compruebes la validez de la firma para cada notificación recibida.

Bloqueo / desbloqueo de un latch

Esta notificación se envía cada vez que un usuario modifica el latch de tu aplicación o alguna de sus operaciones interiores.

```
{
  "t":1457913600,
  "accounts":{
    "ACCOUNT_ID1":[
                {
                 "type":"UPDATE",
                 "id":"APP_OR_OPERATION_ID | GLOBAL_LATCH",
                 "source":"USER_UPDATE | DEVELOPER_UPDATE",
                 "new_status":"on | off | interval"
                },
```

```
                { ... }
    ],
    "ACCOUNT_ID2":[
      ...
    ]
  }
}
```

Información:

- **t**: El 'timestamp' (UTC) cuando Latch envía esta notificación.
- **ACCOUNT_ID:** El accountId de tus usuarios obtenido durante la fase de pareado.
- **id:** El identificador de aplicación u operación del latch que el usuario acaba de modificar. Si el usuario modifica el 'latch maestro' de todas las aplicaciones el valor será 'GLOBAL_LATCH'.
- **source:** Si el cambio se ha realizado por el usuario (USER_UPDATE) o por el desarrollador utilizando la API de soporte o la herramienta LST (DEVELOPER_UPDATE)
- **new_status:** El nuevo estado del latch modificado: 'on' para desbloqueos; 'off' para bloqueos; 'interval' si el usuario ha activado la opción 'bloqueo programado' para dicho latch.

3. Secretos Limitados

Cada aplicación de Latch proporciona un identificador de aplicación (appId) y un secreto; este par de claves permiten firmar las peticiones a la API para garantizar que han sido realizadas por el dueño legítimo de dicha aplicación. Sin embargo existen múltiples escenarios donde dicho secreto puede verse comprometido (aplicaciones de escritorio, apps móviles, etcétera).

En estos escenarios, un secreto comprometido podría permitir a un posible atacante realizar operaciones de la API no deseadas. Para evitarlo se pueden crear para cada aplicación hasta un máximo de 3 secretos con un alcance limitado. Estos secretos se utilizan de la misma forma que el secreto maestro, pero solo pueden usarse para firmarse aquellas peticiones para las que se han configurado.

Un secreto limitado puede contener uno o varios de los siguientes permisos:

- **Estado:** Permite realizar llamadas de estado a la API, para conocer el estado de un Latch.
- **Pareados:** Permite utilizar las llamadas de pareado y despareado de un usuario.
- **Soporte:** Permite utilizar las llamadas de soporte de la API para bloquear y desbloquear “latches”.
- **Histórico:** Permite consultar el histórico de acciones de un servicio.
- **Operaciones:** Otorga accesos para realizar las llamadas de gestión de operaciones. Esta gestión posibilita la creación, edición, consulta y eliminación de operaciones.
- **Instancias:** Otorga accesos para realizar las llamadas de gestión de instancias. Esta gestión posibilita la creación, edición, consulta y eliminación de instancias. Para consultar el estado de una instancia se utiliza el permiso 'ESTADO' y para modificar su estado el de 'SOPORTE'.
- **Servidor de TOTPs:** Otorga accesos para realizar las llamadas de gestión de los TOTPs.
- **Nº secretos cuánticos:** Otorga accesos para realizar las llamadas de gestión de secretos cuánticos.
- **Controles de autorización:** Otorga accesos para realizar las llamadas de gestión de control de autorización.

4. Deep Linking

4.1. ¿Qué es el deep linking?

'Deep linking' o enlace profundo es una técnica mediante la cual se permite lanzar aplicaciones móviles nativas mediante un enlace.

El Deep linking permite conectar una url única con una acción concreta de una aplicación móvil; dependiendo de la plataformas móvil, las URIs necesarias para lanzar la aplicación pueden ser diferentes.

Estas son las funcionalidades de nuestras aplicaciones que se pueden hacer actualmente con 'deep linking' o enlace profundo. De momento sólo están disponibles para Android e iOS.

4.2. Deep linking en las aplicaciones Latch

Se puede utilizar cualquiera de estos recursos URI tanto desde un navegador web como desde una aplicación móvil nativa:

Lanzar la aplicación Latch

Android

```
latch://launch
```

iOS

```
latch://launch
```