

Manual de instalación Widget Hipotecario

Encuentra toda la información que necesitas
para implementar el flujo de simulación.

```
<bci-widget  
  api-key="00898656-c6b9-43c5-84f7-fe197ea9709c"  
  proyecto="Iniciativa Dharma"  
  vendedor="Inmobiliaria"  
  rut="111111111"  
  tipo-propiedad="casa"  
  antiguedad-propiedad="nueva"  
  precio-propiedad="3000"  
  user-id="portal-id"  
></bci-widget>
```

 **Bci** API Market



Introducción

Te presentamos este manual de instalación del Widget Hipotecario, que está dirigido a desarrolladores. Podrás encontrar toda la información en lenguaje técnico (HTML y CSS) para implementar el flujo de simulación. Revísalo con tu equipo para que puedan conocer juntos el proceso de integración.



¡TE AYUDAMOS!

Si quieres hacer una consulta o necesitas asistencia, contáctanos a apiprogram@bci.cl y te responderemos a la brevedad posible.



Índice

Manual de instalación

1.	Regístrate y crea una aplicación -----	5
	Regístrate en API Market	
	Crea tu aplicación (proyecto)	
2.	Realiza pruebas en entorno Sandbox -----	10
	Qué es el entorno Sandbox	
	Import JavaScript	
	Agrega el Widget al Body	
	Parametría	
3.	Exploración datos del Widget -----	15
4.	Pasa al entorno productivo -----	22
5.	Requerimientos de seguridad / Canal de Soporte -----	25

PASOS GENERALES INSTALACIÓN API WIDGET HIPOTECARIO



Regístrate y agrega tu proyecto en API Market

En la página principal, completa el formulario. A partir de allí, podrás acceder a tu cuenta y gestionar tus proyectos.



Realiza pruebas en entorno Sandbox

Ambiente con las mismas características que el productivo donde se pueden ejecutar las API con datos de prueba ficticios.



Pasa a entorno productivo

Accede al Widget Hipotecario en ambiente productivo.

REGÍSTRATE Y CREA TU APLICACIÓN



1



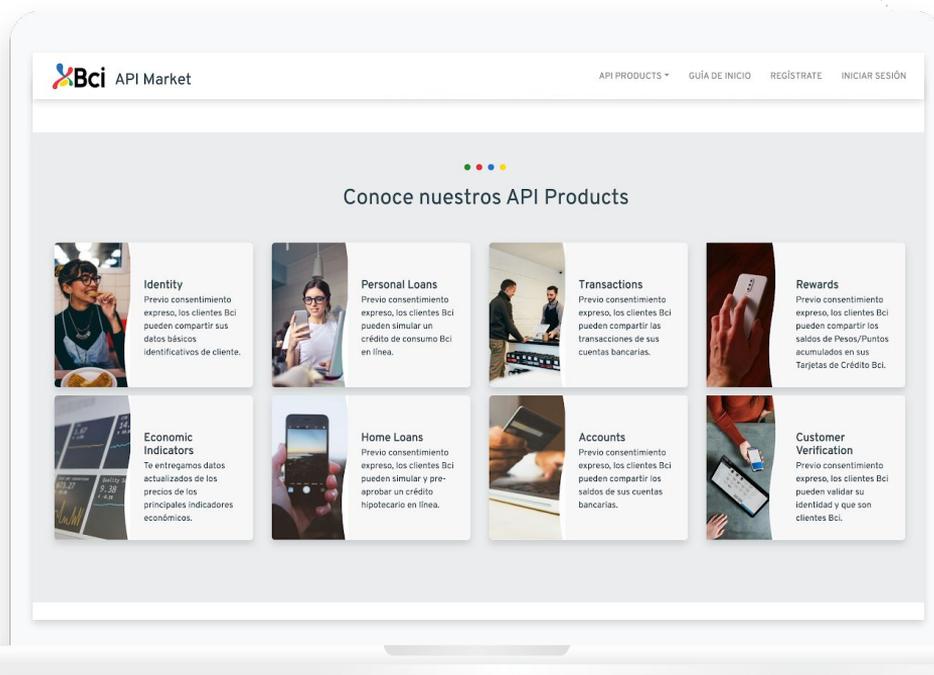
1. Regístrate y crea tu aplicación

1.1 Regístrate en Api Market

<https://apimarket.bci.cl/>

En nuestro catálogo podrás encontrar los API Products disponibles, con su descripción y una explicación de su funcionalidad, así como también, documentación técnica de cada API.

Para ver la documentación técnica del Widget Hipotecario, tienes que tener tu sesión iniciada.





1. Regístrate y crea tu aplicación

1.2 Regístrate en Api Market

El Widget Hipotecario que encontrarás en el catálogo de API Market está disponible para que lo pruebes en entorno Sandbox.

Para ello, solo debes registrarte. Desde el botón “Regístrate” de nuestra página principal, completa el formulario. A partir de allí, podrás acceder a tu cuenta y gestionar tus aplicaciones.





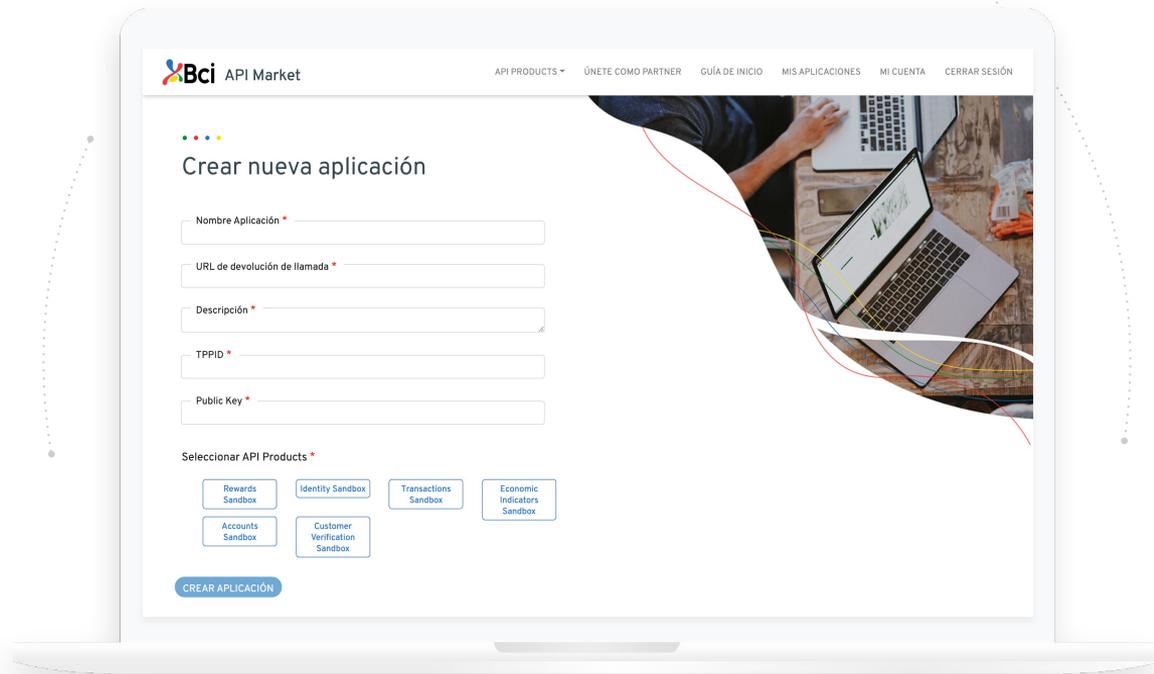
1. Regístrate y crea tu aplicación

1.3 Crea tu aplicación (proyecto)

Para consumir nuestros API Products en ambiente Sandbox, debes seguir los siguientes pasos:

1. Inicia sesión
2. Crea una aplicación (proyecto)

- **Nombre Aplicación:** Debes indicar el nombre que tiene o tendrá tu aplicación que consumirá APIs. Este nombre será mostrado al cliente BCI cuando dé el consentimiento.
- **URL de devolución de llamada:** Aquí debes informar la URL de tu sitio a la cual debe volver el proceso de Oauth (BCI Access) al finalizar
- **Descripción:** ¿En qué consiste tu aplicación?
- **PublicKey:** Genera un par de llaves pública y privada, si no sabes cómo al final de esta guía hay un tutorial de como hacerlo. Debes pegar la llave pública en este formulario. La privada guardala pues la usarás cuando consumas las APIs desde tu aplicación.
- **Tppid:** Completar con un valor numérico de 5 dígitos que luego se utilizará para informar en las peticiones donde así se lo requieran las APIs publicadas.
- Asocia los API Products que te interesa consumir.
- Prueba la integración de tu aplicación con los API Products en modo sandbox utilizando las guías técnicas de cada API Product.



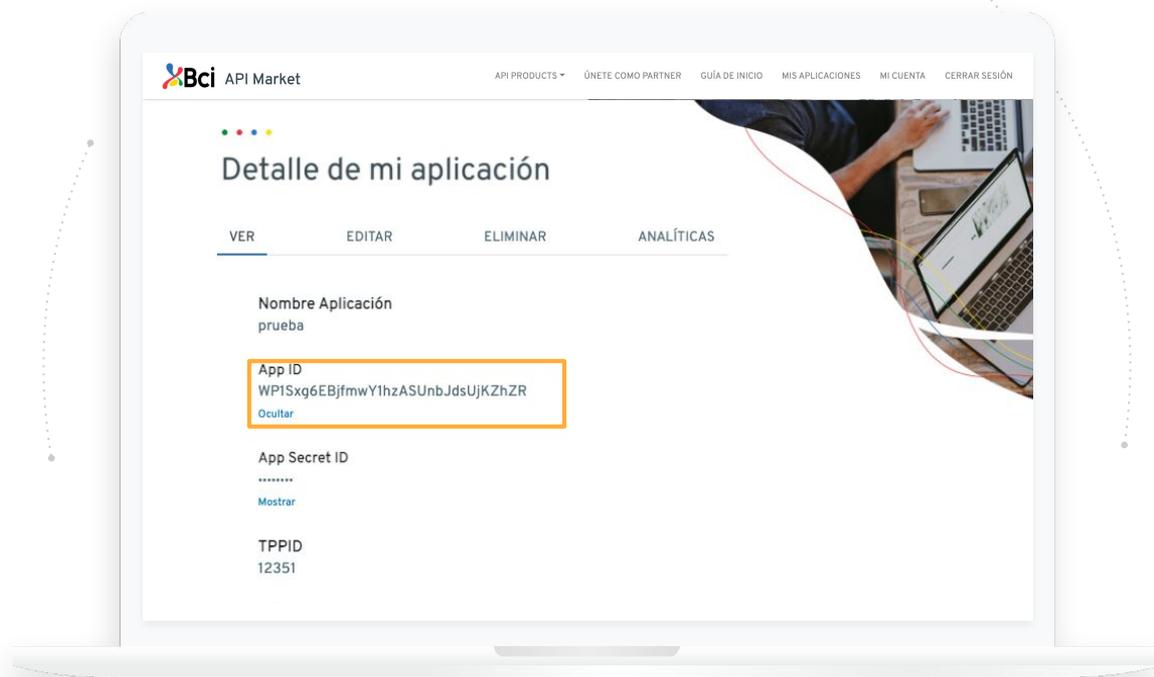


1. Regístrate y crea tu aplicación

1.4 Api Key

Para consumir nuestros API Products se requiere que utilices el api-key, que es el token de solicitud para el Access Request.

Podrás encontrar el api-key ingresando a Mi Cuenta → Mis aplicaciones → click en el nombre de la aplicación y luego en APP ID.



PRUEBAS EN ENTORNO SANDBOX



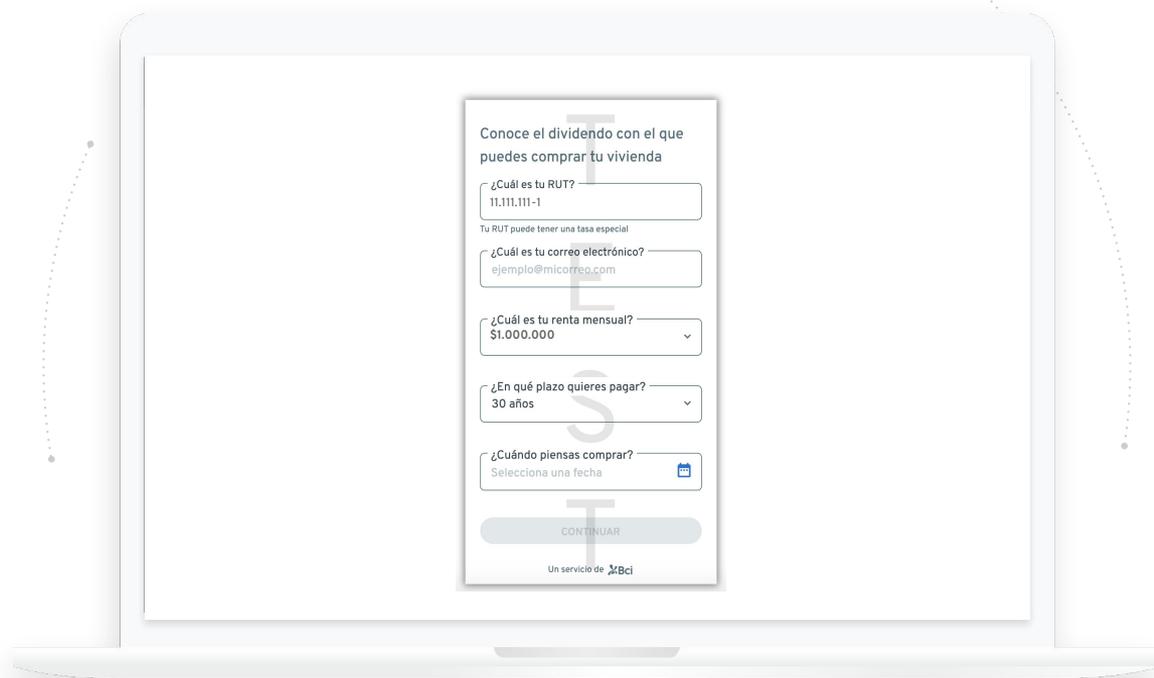


2. Entorno Sandbox

2.1

Realiza pruebas con la API en ambiente Sandbox

También llamado entorno de pruebas, es un ambiente con las mismas características que el productivo donde se pueden ejecutar las API con datos de prueba ficticios.





2. Entorno Sandbox

2.2 Import JavaScript

Este paso es fundamental, realiza la importación del widget en el portal, lo que permite su incorporación en el HTML.

Se debe hacer import de la librería javascript en el **header**:

```
<html>
  <head>
    <script
      src="https://personas.bci.cl/nuevaWeb/fe-ecochip-widget-sandbox/bci-w
      idget.js" defer>
    </script>
  </head>
  <body>
    .
    .
    .
    .
  </body>
</html>
```



“Código editable copiar y pegar”



2. Entorno Sandbox

2.3 Agrega el Widget al body

Se debe agregar el widget al site con la api-key asignada (**obligatorio**)

```
<bci-widget  
  api-key="00898656-c6b9-43c5-84f7-fe197ea9709c">  
</bci-widget>
```



Esta parte del código debes reemplazarla por la Api Key que tienes asignada.



2. Entorno Sandbox

2.4 Parametría

- **api-key (obligatorio):** Variable id que identifica al Partner, dado por API Market, al momento de firmar el contrato de colaboración (requerido al invocar el widget).
- **proyecto (opcional):** Proyecto sobre el cual el usuario está consultando
- **vendedor (opcional):** Nombre de inmobiliaria/corredora
- **rut (opcional):** Rut del usuario. Se utiliza para precargar los datos del widget. Debe ser un rut válido (admite . y -, ex: 11.111.111-1)
- **tipo-propiedad (opcional):** Tipo de la propiedad que el usuario está consultando. Se utiliza para precargar los datos del widget. Valores aceptados: "casa", "dpto" (case insensitive)
- **antigüedad-propiedad (opcional):** Antigüedad de la propiedad que el usuario está consultando. Se utiliza para precargar los datos del widget. Valores aceptados: "nueva", "usada" (case insensitive)
- **precio-propiedad (opcional):** Valor de la propiedad en UF que el usuario está consultando. Se utiliza para precargar los datos del widget. Debe ser un número válido
- **user-id (opcional):** Para el caso de portales que tienen registrado al usuario en su plataforma, es posible asociar este dato a los eventos disponibles por el Widget.
- **email (opcional):** Email del usuario y se utiliza para ser precargado en los formularios y, debe ser un email válido o de lo contrario se omitirá su valor.
- **fecha-intencion-compra (opcional):** Fecha estimada de compra de la propiedad, se utiliza para ser precargada. Debe ser una fecha válida en formato DD-MM-YYYY, ex: 24-10-2020 y debe ser mayor a la fecha actual, de lo contrario se omitirá su valor.
- **duracion-credito (opcional):** Duración del crédito que el usuario desea, debe ser un valor mayor o igual a 5 y menor o igual a 30 de lo contrario se omitirá su valor.
- **telefono (opcional):** Número telefónico del usuario, debe tener al menos 8 números, si tiene más se toman los últimos 8 y se antepone el código de área +569 en todos los casos, ex: +56900000000 ó 900000000 ó 000000000.

```
<bci-widget
  api-key="00898656-c6b9-43c5-84f7-fe197ea9709c"
  proyecto="Iniciativa Dharma"
  vendedor="Inmobiliaria"
  rut="111111111"
  tipo-propiedad="casa"
  antigüedad-propiedad="nueva"
  precio-propiedad="3000"
  user-id="portal-id"
  email="ejemplo@micorreo.com"
  fecha-intencion-compra="02-02-2021"
  duracion-credito="20"
  telefono="+56999999999">
</bci-widget>
```



“Código editable copiar y pegar”

EXPLORACIÓN DATOS DEL WIDGET



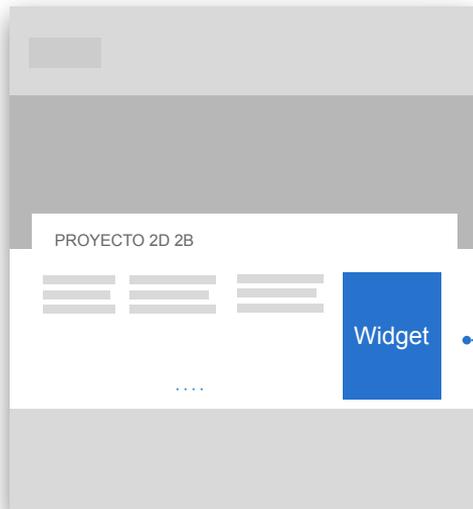
Captura de datos con eventos de escucha

El widget permite la captura de datos registrados por el cliente para:

- Simulación
- Pre-Aprobación
- Aprobación Comercial

¿Cómo?

Por medio de los eventos de escucha, las inmobiliarias tendrán la libertad de definir en donde disponibilizar los datos a sus ejecutivos comerciales.



Utilizar codificación indicada para el evento que se desea capturar (simulación pre-aprobación o aprobación)

Dependiendo donde se quiera disponibilizar los datos, se debe desarrollar la integración dependiendo el destino de los datos.

Nota: se requiere conocimientos de programación

Ejemplo de destinos de datos





3. Exploración de datos del widget

3.1 Eventos emitidos por el Widget:

El Widget emite eventos personalizados en tres momentos de su flujo: cuando el usuario simula, se pre-aprueba o se aprueba. Para conectarse a estos eventos solo se necesita añadir un **event listener** al widget (1).

Cabe destacar que la captura de los eventos les permitirá hacer seguimiento de las simulaciones, pre-aprobaciones y aprobaciones realizadas en el sitio. Estas capturas pueden ser almacenadas de la forma que estimen conveniente (archivo de texto, base de datos, etc.)

(1) La implementación de estos eventos es opcional y no impide el correcto funcionamiento del widget



“Código editable copiar y pegar”

```
document.querySelector('bci-widget').addEventListener(
  'simulacion',
  event => { // tratamiento del evento simulación
    console.log(event.detail.rut) // Ejemplo de imprimir en
  }
)

document.querySelector('bci-widget').addEventListener(
  'preaprobacion',
  event => { // tratamiento del evento preaprobación
    console.log(event.detail.rut) // Ejemplo de imprimir en
  }
)

document.querySelector('bci-widget').addEventListener(
  'finalizacion',
  event => { // tratamiento del evento preaprobación
    console.log(event.detail.flujo) // Ejemplo de imprimir en
  }
)
```



3. Exploración de datos del widget

3.2 Eventos emitidos por el Widget:

El campo event es del tipo [CustomEvent](#).
Los datos específicos de cada evento están disponibles en `event.detail`

Eventos:

Simulación:

Nombre(*simulacion*)

Se emite cuando el usuario realiza una simulación. En el campo `detail` del evento se envían los siguientes datos:

```
interface SimulatedEvent {  
  rut: string  
  plazo: number  
  tipoPropiedad: string  
  propiedadNueva: boolean  
  valorPropiedad: number  
  pie: number  
  dividendo: number  
  tasa: number  
  cae: number  
  userId: string | null  
  email: string  
  tramoRenta: string  
  proyecto: string  
  vendedor: string  
}
```



“Código editable copiar y pegar”



3. Exploración de datos del widget

3.3 Eventos emitidos por el Widget:

Pre-aprobación:

Nombre(*preaprobacion*)

Se emite cuando el usuario realiza una pre-aprobación. En el campo *detail* del evento se envían los siguientes datos:

Cabe destacar que el email y teléfono del cliente se incluyen en la carta de pre-aprobación que se recibe en el correo o casilla definida por la inmobiliaria.

Posibles valores de estado:

Pre-aprobado, Rechazado, Contraoferta

```
interface PreApprovedEvent {  
  rut: string  
  plazo: number  
  tipoPropiedad: string  
  propiedadNueva: boolean  
  valorPropiedad: number  
  telefono: string  
  email: string  
  tipoContrato: string  
  antiguedadLaboral: number  
  tipoRenta: string  
  renta: number  
  userId: string  
  proyecto: string  
  vendedor: string  
  montoSolicitado: number  
  montoAprobado: number  
  dividendo: number  
  tasa: number  
  cae: number  
  estado: 'Pre-aprobado' | 'Rechazado' | 'Contraoferta'  
}
```



“Código editable copiar y pegar”



3. Exploración de datos del widget

3.4 Eventos emitidos por el Widget:

Aprobación:

Nombre(*aprobación*)

Se emite cuando el usuario realiza una aprobación. En el campo *detail* del evento se envían los siguientes datos:

Posibles valores de estado:

*Aprobado, Rechazado,
Contraoferta, Posible-Codeudor*



“Código editable copiar y pegar”

```
interface ApprovedEvent {
  rut: string
  nombre: string
  nivelEstudios: string
  estadoCivil: string
  extranjero: boolean
  residenciaDefinitiva?: boolean
  tipoPropiedad: string
  propiedadNueva: boolean
  valorPropiedad: number
  telefono: string
  email: string
  tipoContrato: string
  antiguedadLaboral: number
  tipoRenta: string
  renta: number
  userId: string
  proyecto: string
  vendedor: string
  montoSolicitado: number
  montoAprobado: number
  plazoSolicitado: number
  plazoAprobado: number
  dividendo: number
  tasa: number
  cae: number
  estado: 'Aprobado' | 'Rechazado' | 'Contraoferta' | 'Posible-Codeudor'
}
```



3. Exploración de datos del widget

3.5 Eventos emitidos por el Widget:

Finalización:

Nombre(*finalización*)

Se emite cuando el usuario interactúa con los botones de final de los flujos o aborta en algún punto de ellos. En el campo *detail* del evento se envían los siguientes datos:

Posibles valores de *flujo*:

Preaprobacion, Aprobacion o Simulacion.

Posibles valores de *tipo*:

VolverASimular, Finalizar o Abortar.



“Código editable copiar y pegar”

```
interface FinalizationEvent {  
    flujo: string;  
    tipo: string;  
}
```

PASO A ENTORNO PRODUCTIVO





4. Paso a entorno productivo

4.1 Te enviaremos la nueva URL y te daremos acceso al entorno productivo en 48 hrs.

El orden de cambio para dar acceso productivo al Widget toma 48 hrs.

teamecochip@bci.cl se contactará para entregar nueva URL e informar instrucciones.

```
<html>
  <head>
    <script
      src="https://personas.bci.cl/nuevaWeb/fe-ecohip-widget-sandbox/b
      ci-widget.js" defer>
    </script>
  </head>
  <body>
    .
    .
    .
    .
  </body>
</html>
```

REQUERIMIENTOS DE SEGURIDAD Y CANAL DE SOPORTE





5. Seguridad / Soporte

Añadir a la configuración del Servidor Web, o por medio de la aplicación web, cabeceras “X-Frame-Options” en la respuesta HTTP, con el valor “SAMEORIGIN”.

Reporte de errores

Si se presenta algún error a la hora de instalar el widget o durante su funcionamiento, se debe enviar la siguiente información para facilitar el proceso de solución.

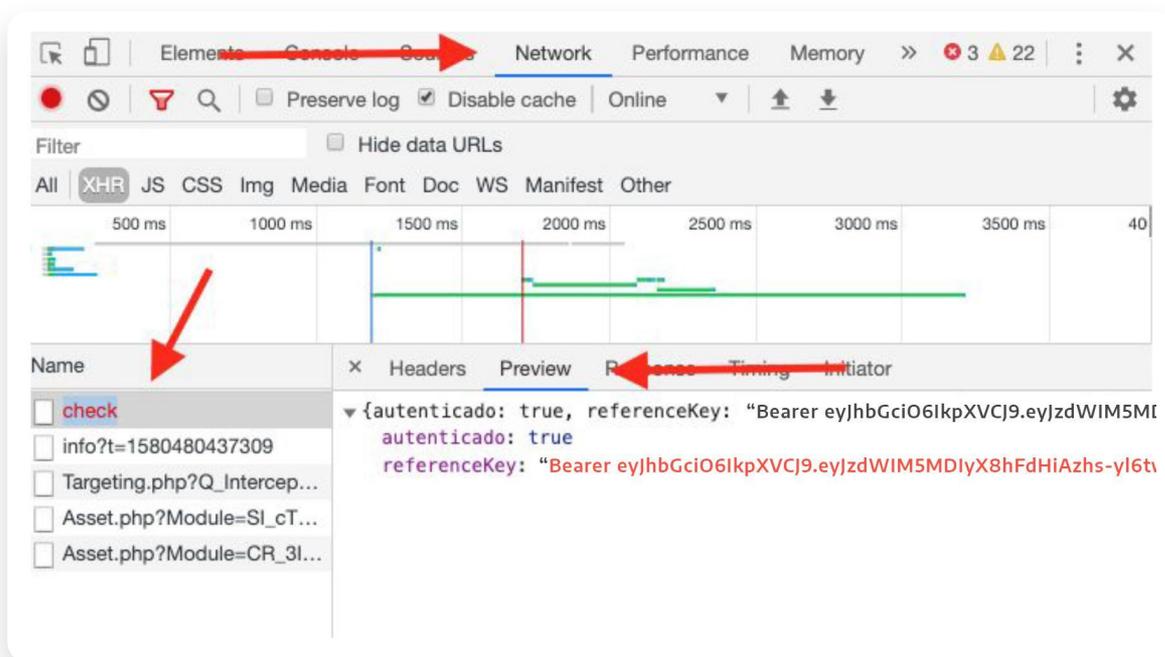
Ejemplo: Captura de pantalla del error en la consola

```
GET http://localhost:4200/bci-widget.js net::ERR_ABORTED 404 (Not Found) (index):82
[WDS] Live Reloading enabled. client:52
▶ 22 A cookie associated with a cross-site resource at <URL> was set without the `SameSite` attribute. A future release of Chrome will only deliver cookies with cross-site requests if they are set with `SameSite=None` and `Secure`. You can review cookies in developer tools under Application>Storage>Cookies and see more details at <URL> and <URL>.
▶ POST http://localhost:8882/check 401 (Unauthorized) zone-evergreen.js:2952
▶ ERROR Error: Credenciales no válidas core.js:6014
  api-key: 00898656-c6b9-43c5-84f7-fe197ea9709c
  url: http://localhost:4200/
    at SafeSubscriber._error (app.component.ts:100)
    at SafeSubscriber.__tryOrUnsub (Subscriber.js:185)
    at SafeSubscriber.error (Subscriber.js:137)
    at Subscriber._error (Subscriber.js:75)
    at Subscriber.error (Subscriber.js:55)
    at MapSubscriber._error (Subscriber.js:75)
    at MapSubscriber.error (Subscriber.js:55)
    at FilterSubscriber._error (Subscriber.js:75)
    at FilterSubscriber.error (Subscriber.js:55)
    at MergeMapSubscriber.notifyError (OuterSubscriber.js:7)
```



5. Seguridad / Soporte

Captura de pantalla de la respuesta de la llamada http que falló.





5. Seguridad / Soporte

Ejemplo: Captura de pantalla de los headers de la llamada http que falló.

The screenshot shows the Network tab of a browser's developer tools. The 'Headers' sub-tab is active, displaying the following information:

- General**
 - Request URL: `http://localhost:8882/simulaciones`
 - Request Method: `POST`
 - Status Code: `500 Internal Server Error`
 - Remote Address: `127.0.0.1:8882`
 - Referrer Policy: `no-referrer-when-downgrade`
- Response Headers (7)**
- Request Headers** (view source)
 - Accept: `application/json, text/plain, */*`
 - Accept-Encoding: `gzip, deflate, br`
 - Accept-Language: `en-US,en;q=0.9,es;q=0.8`

The list of requests on the left includes:

- check
- info?t=1580481368449
- Targeting.php?Q_Intercep...
- Asset.php?Module=SI_cT...
- Asset.php?Module=CR_3I...
- indicadores-financieros
- token
- version
- validate
- simulaciones**
- register



5. Seguridad / Soporte

Captura de pantalla de los datos enviados en la llamada http que falló.

The screenshot shows the Chrome DevTools Network tab with the 'XHR' filter selected. A failed request is visible in the timeline. The 'Headers' tab is open, showing the following response headers:

- Authorization: Bearer eyJhbGciOiJIcXVCJ9.eyJzdWIM5MDlyX8hFdHiAZhs-y
- Cache-Control: no-cache
- client-id: 00898656-c6b9-43c5-84f7-fe197ea9709c
- Connection: keep-alive
- Content-Length: 321
- Content-Type: application/json
- Host: localhost:8882
- Origin: http://localhost:4200
- Pragma: no-cache
- Referer: http://localhost:4200/
- Sec-Fetch-Mode: cors
- Sec-Fetch-Site: same-site
- User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/79.0.3945.130 Safari/537.36
- x-ibm-client-id: 00898656-c6b9-43c5-84f7-fe197ea9709c

The 'Request Payload' section shows the following JSON data:

```
{
  "fechaIntencionCompra": "01-01-2020",
  "pie": 1000,
  "plazo": 30,
  "propiedadNueva": true,
  "tipoBienRaiz": "DPTO",
  "valorPropiedad": 3000,
  "solicitante": {
    "email": "Ejemplo@micorreo.com",
    "renta": 1000000,
    "rut": 11111111,
    "dv": "1",
    "vendedor": "Inmobiliaria",
    "proyecto": "Iniciativa Dharma",
    "creditoTasaFija": true,
    "refinanciamiento": false
  }
}
```



5. Seguridad / Soporte

*Datos de la llamada
http que falló.*

The screenshot shows the Network tab in a browser's developer tools. The 'simulaciones' request is selected, and its details are visible. The status code is 500 Internal Server Error. A context menu is open over the request, and the 'Copy as fetch' option is highlighted. A red arrow points to this option.

Click derecho

Copy as fetch

Request Headers (7)

```
Accept: */*
Accept-Language: es-ES,en;q=0.9,es;q=0.8
Cache-Control: no-cache
Content-Type: application/json
Origin: http://localhost:8882
Referer: http://localhost:8882/simulaciones
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36
x-ibm-client-id: 00898656-c6b9-43c5-84f7-fe197ea9709c
```

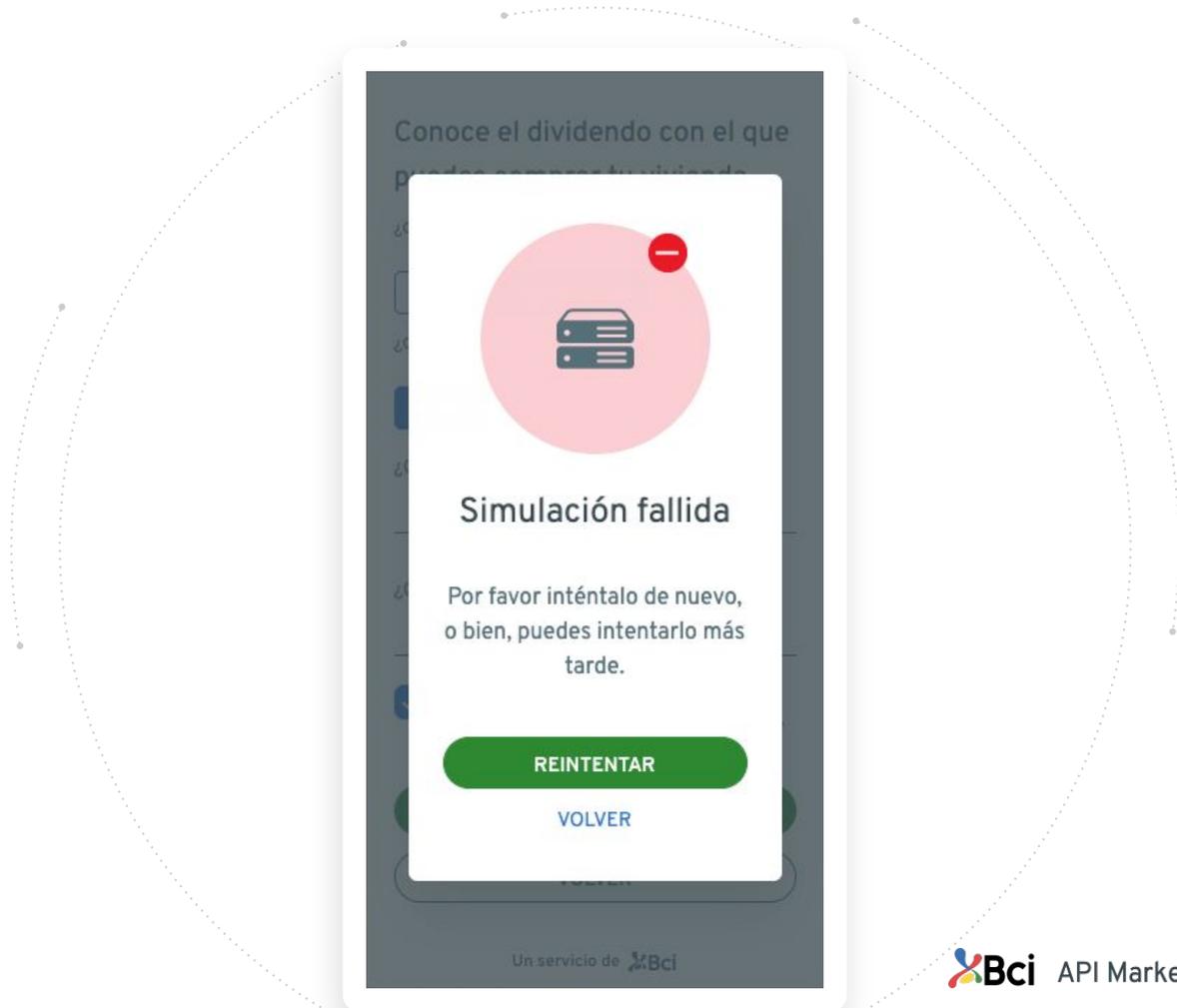
Request Payload

```
{ "fechaIntencionCompra": "01-01-2020", "nie": "1000", "plazo": 30, "promotor": "..." }
```



5. Seguridad / Soporte

Captura de pantalla de error en el widget.





5. Seguridad / Soporte

¿Cómo reportar un error?

Si necesitas reportar un error
escríbenos a nuestro correo:

teamwidgetchip@bci.cl

Para reportar un error y facilitar la respuesta de nuestro equipo de soporte, te recomendamos que en el asunto del correo indique lo siguiente:

[Nombre Empresa] [RUT Cliente] [Tipo de Error]

Ejemplo:

[Nombre empresa] Instalación API

[Nombre empresa] Registro de la API

[Nombre empresa] Eventos emitidos por el widget



Asegúrate de **adjuntar una captura de pantalla junto al detalle del correo**, así podremos resolver tu problema lo antes posible.

Manual de instalación
Widget Hipotecario

