

Installation Manual Mortgage Widget

You can find all the information you need to develop the simulation flow.

```
<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



Intro

We present this Mortgage Widget installation manual, which is oriented to developers. You can find all the technical information (HTML and CSS) to implement the simulation flow. Review it with your team so they can learn the integration process.



WE CAN HELP YOU!

If you have some doubts or need some help, contact us: apiprogram@bci.cl and we will give you an answer as soon as we can.



Index

Installation Manual

| | | |
|----|--|----|
| 1. | Sign Up and Create an Application ----- | 5 |
| | Sing Up in API Market | |
| | Create your own Application (project) | |
| 2. | Test in Sandbox Environment ----- | 10 |
| | What is the Sandbox Environment | |
| | Import JavaScript | |
| | Add Widget to the Body | |
| | Parameters | |
| 3. | Widget data Exploration ----- | 15 |
| 1. | Go to Production Environment ----- | 22 |
| 1. | Security Requirements / Support channel ----- | 25 |

GENERAL STEPS FOR MORTGAGE WIDGET INSTALLATION



Sign Up and Create your Project in the API Market

Complete the form on the main Page. Then you can access to your account and manage your projects.



Test in Sandbox Environment

The environment with the same characteristics as a productive environment. APIs can be used with test data.



Get access to Productive Environment

Get access to Mortgage Widget in Productive environment.

SIGN UP AND CREATE YOUR APPLICATION



1





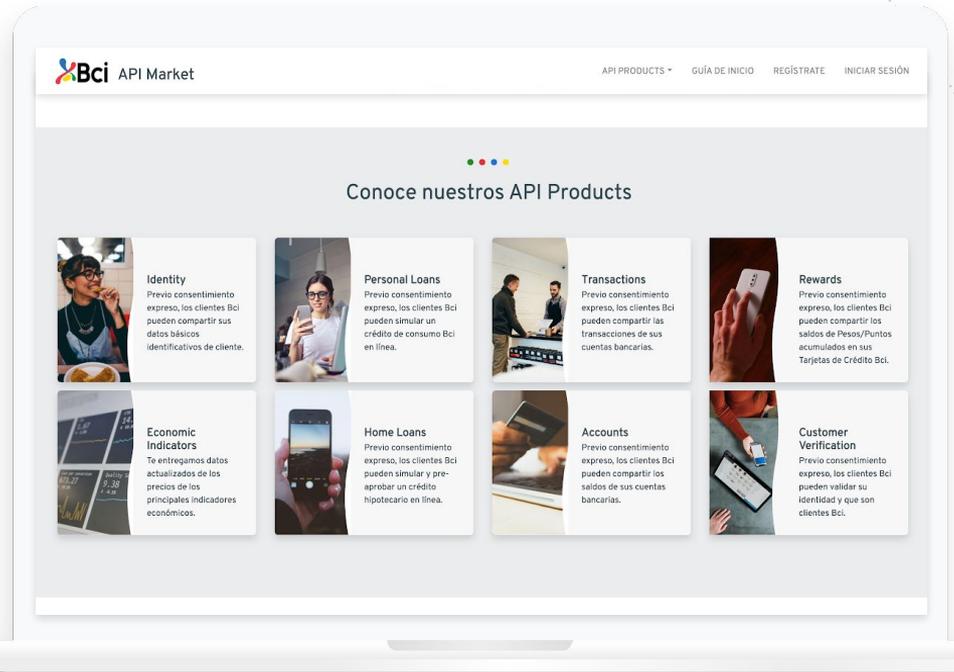
1. Sign up and register your application

1.1 Sign up in API Market

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

In our catalog you will find the available API Products, with their description and an explanation about its functionality and the technical documentation for each API.

To see the technical documentation of the Mortgage Widget you must be [logged in](#).



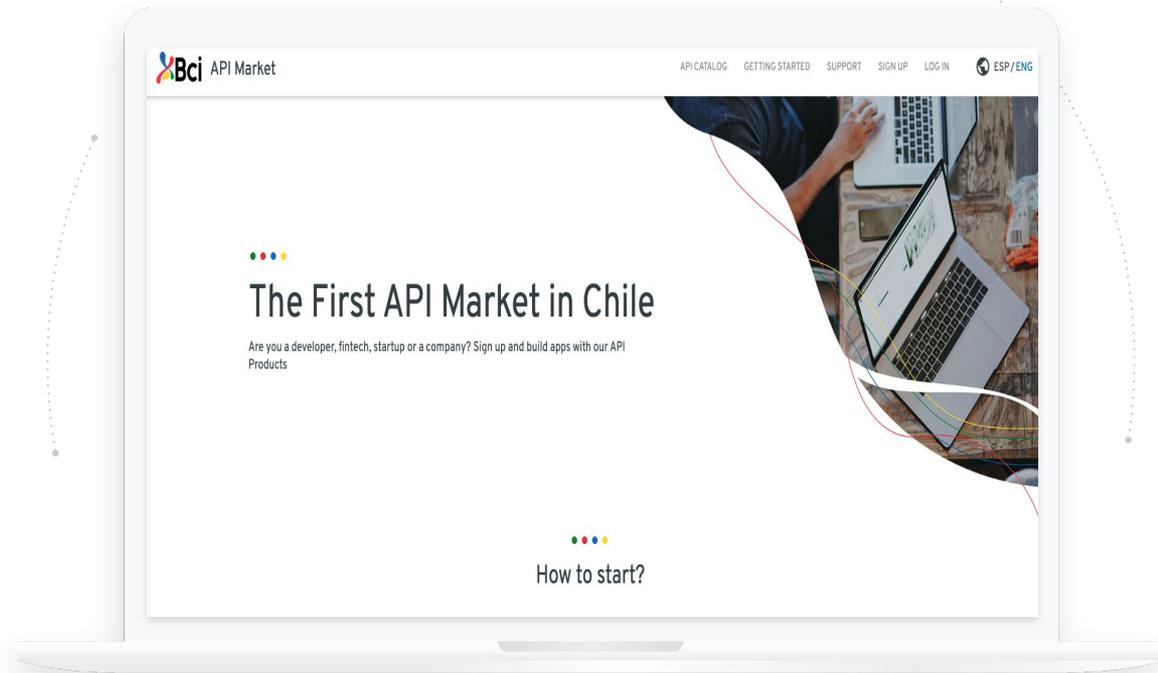


1. Sign up and register your application

1.2 Sign up in API Market

The Mortgage Widget is available for you to test in a Sandbox Environment.

To do this, you have to register. Please, fill in the form from the "Sign up" button on the home page, then you can access your account and manage your apps.(or Register)





1. Sign up and register your application

1.3 Register your application (project)

To use our API Products in Sandbox Environment you need to follow these steps:

1. [Log In](#) into API Market
2. Create an Application (project)

- **Application Name:** : Indicate the name that your application has or will have that will consume APIs. This name will be shown to the BCI client when giving consent.
- **Callback URL:** this way, you must inform the URL on your site where the Oauth (BCI Access) must return after finish
- **Description:** What does your application consist of?
- **PublicKey:** Generates a pair of public and private keys, if you don't know how, you can find a tutorial at the end of the guide. Paste the public key in this form and save the private one since you will use it to consume APIs from your application.
- **TPPID:** Complete with a 5-digit numeric value that will later be used to inform requests where required by the published APIs.
- Associate the API Products that you are interested in consuming.
- Test the integration of your application with the API products in sandbox mode using the technical guides of each API product.

The screenshot shows the 'Create new app' form on the Bci API Market website. The form has the following fields and options:

- App name ***: Text input field.
- Callback URL ***: Text input field.
- Description ***: Text input field.
- TPPID ***: Text input field.
- Public Key ***: Text input field.
- APIs ***: Section with three checkboxes:
 - Business Notifications Simulation
 - Customer Verification Sandbox
 - Widget Hipotecario Sandbox
- ADD APP**: Blue button at the bottom.

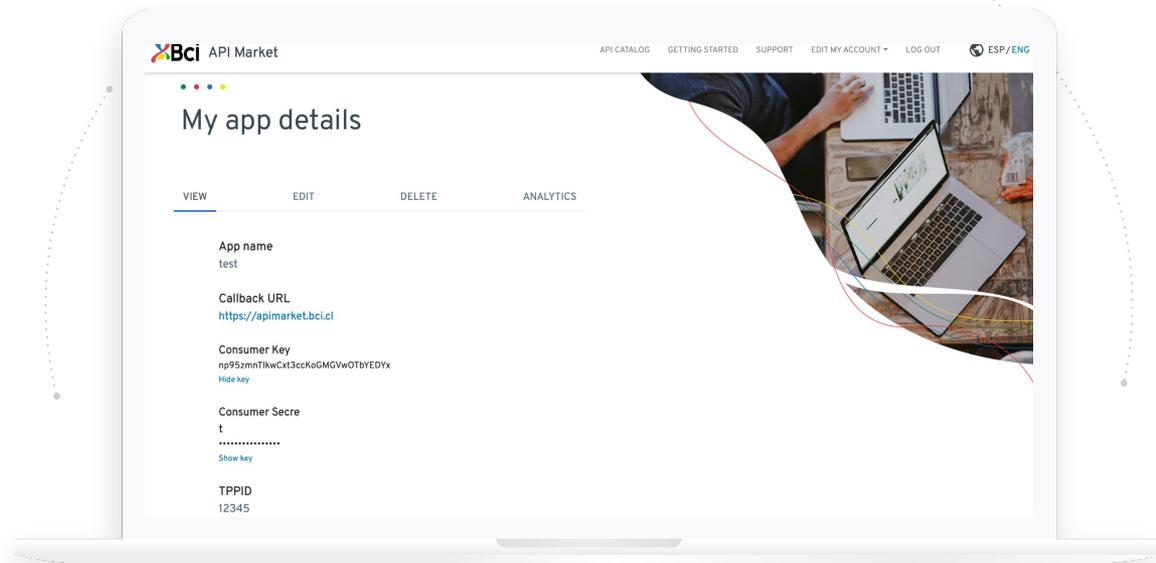


1. Sign up and register your application

1.4 Api Key

To use our API Products it is required the api-key, that is the token of the Access Request.

You will find the api-key going to Edit My Account → My Apps → click in the application name and then in Api Key.



TESTS IN SANDBOX ENVIRONMENT



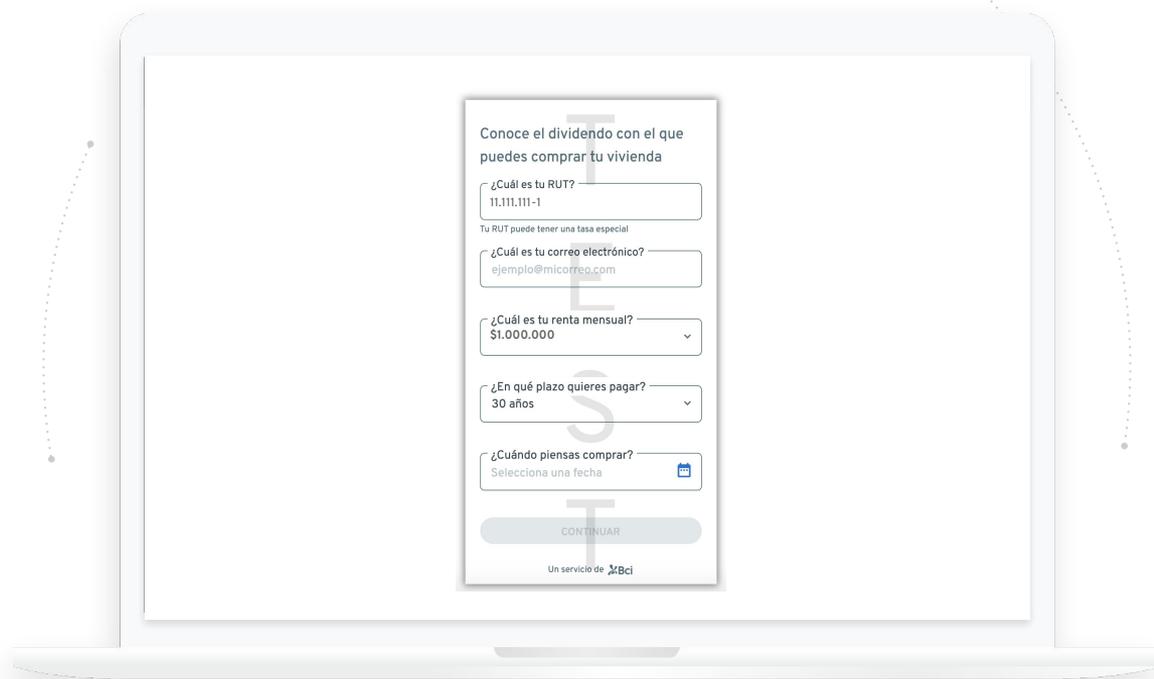


2. Sandbox environment

2.1

Test with the API in Sandbox environment

Also called test environment, it's an environment that shares same features as production you can execute APIs with test data





2. Sandbox environment

2.2 Import JavaScript

This step is essential, it carries out the import of the widget in the portal, which allows its incorporation into the HTML.

You must import the javascript library in the **header**:

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



“Copy and paste this code in your html”



2. Sandbox environment

2.3 Add the Widget to the body

The widget must be added to the site with the api-key assigned (required)

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



In this part of the code you must replace the value with the Api Key that was assigned to you.



2. Sandbox environment

2.4 Parametry

- **api-key (required):** Variable id that identifies the Partner, given by API Market, at the time of signing the collaboration contract (required when invoking the widget).
- **project (optional):** Project on which the user is consulting
- **seller (optional):** Realtor / broker name
- **rut (optional):** User's Rut. It is used to preload the widget data. Must be a valid rut (admits . and -, ex: 11.111.111-1)
- **property-type(optional):** The type of the property that the user is querying. It is used to preload the widget data. Accepted values: "house", "apartment" (case insensitive)
- **antiquity-property (optional):** Age of the property that the user is viewing. It is used to preload the widget data. Accepted values: "new", "used"(case insensitive)
- **price-property (optional):** Property value in UF that the user is querying. It is used to preload the widget data. Must be a valid number
- **user-id (opcional):** In the case of portals that have the user registered on their platform, it is possible to associate this data with the events available through the Widget.
- **email (opcional):** : User's email and is used to be preloaded in the forms and, it must be a valid email or otherwise its value will be omitted.
- **purchase-intention-date (optional):** Estimated date of purchase of the property, it is used to be preloaded. It must be a valid date in the format DD-MM-YYYY, ex: 10-24-2020 and must be greater than the current date, otherwise its value will be ignored.
- **duration-credit (optional):** Duration of the credit that the user wants, must be a value greater than or equal to 5 and less than or equal to 30, otherwise its value will be omitted.
- **phone (optional):** Telephone number of the user, it must have at least 8 numbers, if it has more, the last 8 are taken and the area code is prepended +569 in all cases, ex: +56900000000 or 900000000 or 00000000.



"Copy and paste this code in your html"

```
<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"
  email="ejemplo@micorreo.com"
  fecha-intencion-compra="02-02-2021"
  duracion-credito="20"
  telefono="+56999999999">
</bci-widget>
```

WIDGET DATA EXPLORATION



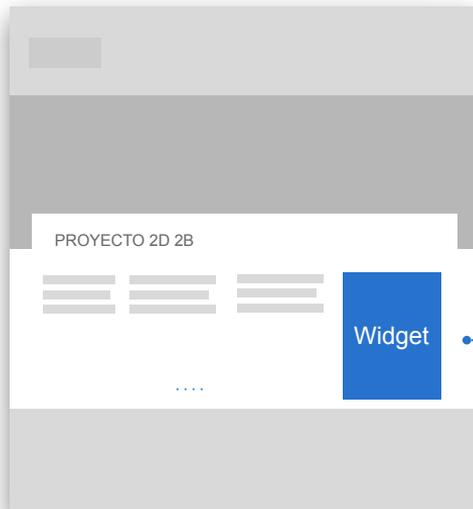
Data capture with listening events

The widget allows to capture the data registered by the client to:

- Simulation
- Pre-approval
- Commercial Pre-approval

¿How?

Through listening events, real estate companies will have the freedom to define where to make the data available to their commercial executives.



Use coding indicated for the event you want to capture (pre-approval simulation)

Depending on where you want to make the data available, the integration must be developed depending on the destination of the data.

Note: Programming knowledge is required

Example of data destinations





3. Widget data exploration

3.1 Events emitted by the Widget:

The Widget emits custom events at three points in its flow: When the user simulates or pre-approves. To connect to these events you only need to add an event listener to the widget (1).

It should be noted that the capture of the events will allow to follow the simulations and pre-approvals carried out on the site. These captures can be stored in the way deem appropriate (text file, database, etc.)

(1) The implementation of these events is optional and does not prevent the correct functioning of the widget

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

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

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



“Copy and paste this code in your html”



3. Widget data exploration

3.2 Events emitted by the Widget:

The event field is of type [CustomEvent](#).
The specific data of each event is available in event.detail

Events:

Simulation:

Name(*simulation*)

Issued when the user performs a simulation. In the detail field of the event the following data is sent:

```
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  
}
```



“Copy and paste this code in your html”



3. Widget data exploration

3.3 Events emitted by the Widget:

Pre-approval:

Name (pre-approval)

Issued when the user performs a pre-approval. The following data is sent in the detail field of the event:

It should be noted that the client's email and telephone number are included in the pre-approval letter that is received in the mail or box defined by the real estate agency.

Possible status values:

Pre-approved, Rejected, Counteroffer

```
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'
}
```



“Copy and paste this code in your html”



3. Widget data exploration

3.4 Events emitted by the Widget:

Approval:

Name (approval)

Issued when the user makes an approval. The following data is sent in the detail field of the event:

Possible status values:

*Approved, Rejected,
Counteroffer, Possible-Codeudor*



“Copy and paste this code in your html”

```
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. Widget data exploration

3.5 Events emitted by the Widget:

Ending:

Name (Finalization)

It is emitted when the user interacts with the end buttons of the flows or aborts at some point in them. The following data is sent in the detail field of the event:

Possible flow values:

Pre-approval, approval o Simulation.

Possible type values:

Return ASimulate, Terminate or Abort.

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



“Copy and paste this code in your html”

GO TO PRODUCTIVE ENVIRONMENT



4





4. Step to productive environment

4.1 We'll send you the new URL and grant you access to the production environment within 48 hours.

Change order to grant production access to the Widget takes up to 48 hrs

teamecochip@bci.cl will get in contact to deliver the new URL and inform instructions.

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

SECURITY REQUIREMENTS AND SUPPORT CHANNEL



5





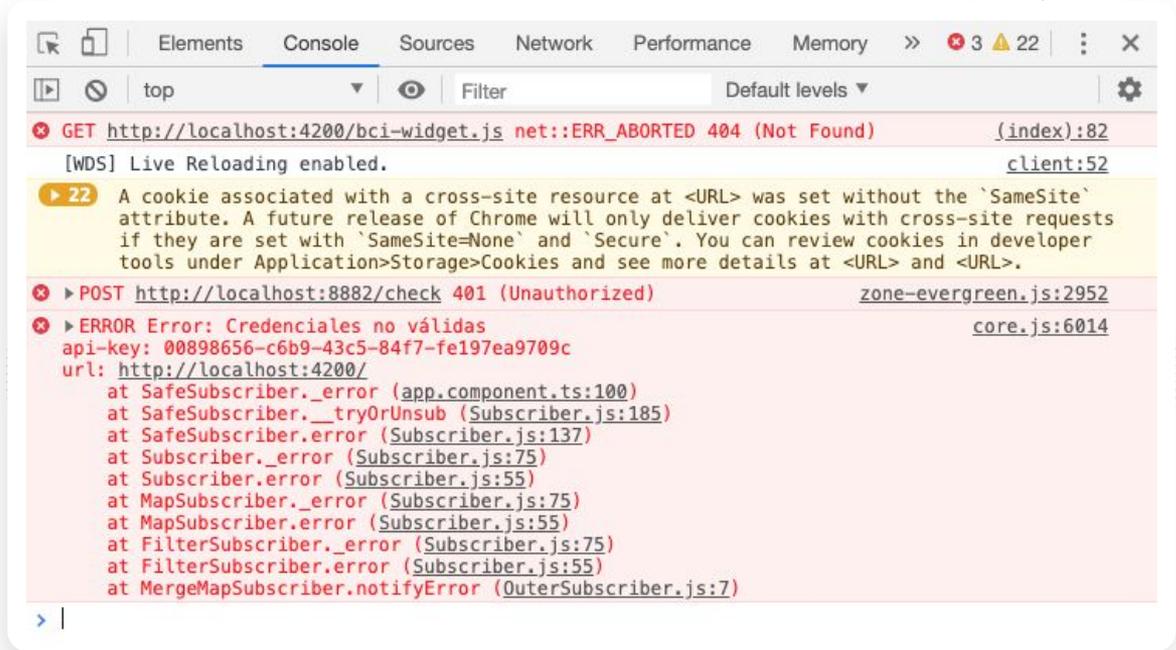
5. Security / Support

Add “X-Frame-Options” headers to the HTTP response to the Web Server configuration, or through the web application, with the value “SAMEORIGIN”

Bug report

If there is an error when installing the widget or during its operation, the following information should be sent to facilitate the solution process.

Example: Screenshot of the error on the console





5. Security / Support

Screenshot of the response of the http call that failed

The screenshot shows the Network tab of a browser's developer tools. The 'XHR' filter is selected. A list of requests is shown, with 'check' highlighted. A red arrow points to this entry. The 'Preview' tab is selected, showing the response body:

```
{autenticado: true, referenceKey: "Bearer eyJhbGciOiJIUzI1NiIsInR5cGU6IjY9.ejZdWIM5MT
autenticado: true
referenceKey: "Bearer eyJhbGciOiJIUzI1NiIsInR5cGU6IjY9.ejZdWIM5MDIyX8hFdHiAzhs-yl6t...
```



5. Security / Support

Example: Screenshot of the headers of the http call that failed.

The screenshot shows the Network tab in a browser's developer tools. The 'XHR' filter is selected, and a list of requests is shown. The 'simulaciones' request is highlighted in red. The 'Headers' sub-tab is selected, showing the following details:

- 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`



5. Security / Support

Screenshot of the data sent in the http call that failed.

The screenshot shows the Network tab in a browser's developer tools. The 'XHR' filter is active, and a list of requests is shown. The 'simulaciones' request is highlighted in red. A red arrow points to this entry. Below the list, the 'Headers' tab is selected, showing the response headers for the failed request. A red arrow points to the 'Headers' tab. The 'Request Payload' section is expanded, showing the JSON data sent in the request.

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM5MTIxOGEwMjU0IiwiaWF0IjoiYXNjaWVudC51b3R9
```

```
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
```

```
{
```

```
  "fechaIntencionCompra": "01-01-2020",
```

```
  "pie": 1000,
```

```
  "plazo": 30,
```

```
  "propiedadNueva": true,
```

```
  "tipoBienRaiz": "DPTO",
```

```
  "valorPropiedad": 3000,
```

```
  "solicitante": {
```

```
    "email": "Ejemplo@micorreo.com",
```

```
    "renta": 1000000,
```

```
    "rut": 111111111,
```

```
    "dv": "1",
```

```
    "vendedor": "Inmobiliaria",
```

```
    "proyecto": "Iniciativa Dharma",
```

```
    "creditoTasaFija": true,
```

```
    "refinanciamiento": false
```

```
  }
```

```
}
```




5. Security / Support

How to report a bug?

If you need to report a bug
write to our email:

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



Make sure to attach a screenshot next to the email detail, so we can solve your problem as soon as possible.

Installation Manual
Mortgage Widget

