



KIBE
Widget customizado
Guía de desarrollo

Contenido

Objetivo:	3
Conceptos:.....	3
Que es un widget?.....	3
Obtención del contenido para construir el Widget.	4
Búsqueda de vuelos:	9
Anexo 1:.....	10
Validaciones recomendadas:Validations recommended:	10

Objetivo:

A lo largo de este instructivo, los usuarios podrán crear sus propios widgets para embeber en las páginas web y de este acceder vía parámetros a las funcionalidades de la KIBE (KIUI INTERNET BOOKING ENGINE).

Este documento es una guía para el desarrollador web que explica cómo obtener y conectar el Widget desarrollado con el motor de reservas.

Esta guía no se adentra en cómo crear el widget, sino, los campos y contenido que debe contar para poder realizar las búsquedas a desplegarse luego en el motor web de KIUI. (KIBE)

Conceptos:

Que es un widget?

Es un formulario en HTML que debe contener estilos .CSS y que permite embeberse dentro de un *iframe* a la página web del cliente y otros sitios.

Ofrece KIUI un widget?

Si. KIUI provee de un widget estándar cuando el producto KIBE es creado, pero, posiblemente este no contempla todos los aspectos que requiere el cliente por lo tanto, este último puede desarrollar su propio widget y conectarse con KIBE.

El widget desarrollado debe contemplar la configuración general de KIBE?

El Widget debe contemplar y obtener el contenido en base a la configuración de la KIBE con el objetivo de realizar las búsquedas de acuerdo a los requerimientos comerciales del cliente. Esta información reside en KIBE Backoffice y debe estar completamente configurado al momento de desarrollar e implementar el widget. En esta guía se explica cómo conectarse y como obtener el contenido a desplegar de acuerdo a la configuración mencionada.

Obtención del contenido para construir el Widget.

Cuando el desarrollador crea el formulario en HTML para generar el widget, hay ciertos campos mandatorios que deben existir para poder enviar la información mínima necesaria a la KIBE y de esta manera obtener resultados.

Con el objetivo de obtener este contenido el desarrollador debe agregar el siguiente Script dentro de la etiqueta `<head>` en el formulario.

```
<script>
    $(document).ready(function() {
        $.ajax({
            url: "[KIBE URL domain]/api/get_complete_config_widget/",
            success: function(response) { formatResponse(response); },
            dataType: "jsonp"
        });
        //Example of how to consume the data
        function formatResponse(response) {
            $("#csrfmiddlewaretoken").val(response.csrf_token);
            $.each(response.airports.keys.origin, function(i,e) {
                $("#id_origin").append("<option value='"+
e.data +'>"+ e.value + "</option>")
            })
            $.each(response.airports.keys.destination, function(i,e) {
                $("#id_destination").append("<option
value='"+ e.data +'>"+ e.value + "</option>")
            })
        }
    });
</script>
```

URL: The Se debe declarar el dominio de la KIBE (Este dato es proveído por KIU al momento de crear la instancia del producto para el cliente) agregando los parámetros `/api/get_complete_config_widget/` (Ej: https://kibe-dev-xx.kiusys.net/api/get_complete_config_widget/).

Esta URL va a devolver la siguiente respuesta en formato *JSON*:

```
{
  "passengers": {
    "max_pax_allowed": 9,
    "pax_config": [
      {
        "max_age": null,
        "enable": true,
        "name": "ADULT",
        "min_age": 12
      },
      {
        "max_age": 12,
        "enable": true,
        "name": "CHILD",
        "min_age": 2
      },
      {
        "max_age": 2,
        "enable": true,
        "name": "INFANT",
        "min_age": 0
      }
    ]
  },
  "csrf_token":
  "vgTgLRd6L1XKeeJdWdMSPDOhyUURzF02XJU10CarwNc3ABHAQeWdMkmgI93QMVOI",
  "airports": {
    "keys": {
      "origin": [
        {
          "code": "AEP",
          "data": "QUVQ",
          "value": "BUENOS AIRES (BUE), JORGE NEWBERY AIRPORT (AEP),
Argentina"
        }
      ],
      "destination": [
        {
          "code": "MDQ",
          "data": "TURR",
          "value": "MAR DEL PLATA (MDQ), ASTOR PIAZZOLA AIRPORT - MERY (MDQ),
Argentina"
        },
        {
          "code": "LAS",
          "data": "TEFT",
          "value": "LAS VEGAS (LAS), MCCARRAN INTERNATIONAL AIRPORT (LAS),
United States"
        }
      ]
    }
  },
  "origin_destination": {
    "keys": [
```

```
{
  "origin": "AEP",
  "destination": "MDQ",
  "not_flying_days_return": [],
  "not_flying_days_going": [],
  "frequency_return": [
    0,
    1,
    2,
    3,
    4,
    5,
    6
  ],
  "frequency_going": [
    0,
    1,
    2,
    3,
    4,
    5,
    6
  ],
  "direct_fly": false
}
]
```

Diccionario de la respuesta:

Nombre	Descripción	Valores / Formato
Passengers	Contiene un objeto con las configuraciones para los distintos tipos de pasajeros y la cantidad máxima permitidos	Objeto
max_pax_allowed	Viene dentro del objeto "Passengers" y es la máxima cantidad de pasajeros permitidas.	Número
pax_config	Es un objeto con las configuraciones necesarias por CATEGORIAS permitidas en la KIBE. Estan son ADULT, CHILD y INFANT.	Objeto
max_age	Es la edad máxima permitida de una categoría. El campo viene dentro cada objeto dentro de pax_config	Número / null
enable	Habilita o deshabilita la categoría. El campo viene dentro cada objeto dentro de pax_config.	true / false
name	Es el nombre de la categoría. El campo viene dentro cada objeto dentro de pax_config	ADULT / CHILD / INFANT
min_age	Es la edad mínima por categoria. El campo viene dentro cada objeto dentro de pax_config	Número
csrf_token	Es el HASH de validación de DJANGO para la validación del SUBMIT del formulario	HASH
airports	Objeto que contiene la información de los aeropuertos necesarias para la validación del formulario	Objeto
keys	Objeto con los aeropuertos de origen y destinos seteados desde el backoffice de la KIBE. El campo viene dentro cada objeto dentro de airports	Objeto
origin	Array de Objetos que contienen la información de los aeropuertos. El campo viene dentro cada objeto dentro de keys	Array
code	Código de IATA del aeropuerto. El campo viene dentro cada objeto dentro de origin	3 letras
data	Es el código que utiliza KIBE para la cotización.	4 letras
value	Texto como se muestra el aeropuerto en el formulario de la KIBE	Alfabético
destination	Idem origin. Mismo formato	Objeto
origin_destination	Objeto de los destinos habilitados en la KIBE con sus respectivas configuraciones	Objeto

Nombre	Descripción	Valores / Formato
keys	Array de objetos, que contiene los destinos habilitados. El campo viene dentro cada objeto dentro de origin_destination	Array
origin	Código IATA del aeropuerto de origen. El campo viene dentro cada objeto dentro de keys	3 Letras
destination	Código IATA del aeropuerto de destino. El campo viene dentro cada objeto dentro de keys	3 Letras
not_flying_days_return	Array con los días de la semana que no tiene permitido volar de vuelta. El campo viene dentro cada objeto dentro de keys	0 a 6 (Siendo 0 Lunes)
not_flying_days_going	Array con los días de la semana que no tiene permitido volar de ida. El campo viene dentro cada objeto dentro de keys	0 a 6 (Siendo 0 Lunes)
frequency_return	Array con los días de la semana que tiene permitido volar de vuelta. El campo viene dentro cada objeto dentro de keys	0 a 6 (Siendo 0 Lunes)
frequency_going	Array con los días de la semana que tiene permitido volar de ida. El campo viene dentro cada objeto dentro de keys	0 a 6 (Siendo 0 Lunes)
direct_fly	Si es vuelo directo o permite conexiones	true / false

Basado en esta respuesta, el desarrollador cuenta con la suficiente información para desplegar dentro de los distintos campos del widget:

- Aeropuertos de salida y arribo
- Rutas habilitadas
- Tipos de pasajeros soportados
- Cantidad máxima permitida de pasajeros para búsquedas
- Tipo de viajes soportados (Ida o Ida y vuelta)

También en base a esta información puede generar filtros y validaciones del lado del cliente.

Búsqueda de vuelos:

Una vez que el formulario ya se encuentra desarrollado y cuenta con el contenido anteriormente descrito, el desarrollador debe enviar estos parámetros a la KIBE, para lo cual, debe agregar dentro de la etiqueta <form> :

- **action:** La URL de destino debe ser el dominio del a KIBE.
- **method:** El método debe ser POST.

Ejemplo:

```
<form class="searcher" id="search-form" name="searchWidgetForm"
action="https://kibe-dev-xx.kiusys.net/" method="POST">
```

Incluidos estos atributos cuando el usuario envía el formulario, KIBE va a tomar los parámetros:

- Origin
- Destination
- Fecha de Salida
- Fecha de regreso (Si existe).
- Cantidad de pasajeros adultos.
- Cantidad de pasajeros Menores e infantes (Si existen).

Realiza una serie de validaciones de datos y procesa el pedido. En este momento, los resultados obtenidos se van a desplegar dentro de la página de KIBE en el ambiente de KIBE.

Anexo 1:

Validaciones recomendadas:

Más allá que la aplicación KIBE tiene sus propias validaciones en el backend, debemos realizar ciertas recomendaciones a los desarrolladores sobre algunas validaciones a realizar del lado del cliente (Front-end) con el objetivo de evitar errores de búsqueda inconsistentes y generar un proceso más eficiente.

1. **Origen - Destino:**

La respuesta de JSON API devuelve la lista de rutas habilitadas para búsquedas configuradas en el backoffice de KIBE. Basado en esta información, el desarrollador puede filtrar y validar las opciones que estén relacionadas y así optimizar los resultados de las búsquedas.

2. **Fechas:**

El formato soportado de fechas es DD-MM-YYYY. El desarrollador puede desplegar en el formulario en diferentes formatos de acuerdo a la zona en la que se encuentre, pero debe considerar hacer la transformación para respetar el formato soportado y evitar de esta manera que la KIBE devuelva error de formato.

3. **Fecha de salida:**

La fecha de salida debe considerarse desde la fecha actual a futuro y no más allá de 330 días en futuro. La aplicación por estándar de la industria soporta búsquedas de disponibilidad hasta 330 días a futuro de la fecha actual.

4. **Fecha de regreso:**

La fecha de regreso debe validarse contra la fecha de salida. Esta no puede ser anterior a la fecha fijada como salida y no debe ser superior a 330 días a futuro.

5. **Pasajero adulto:**

Recomendamos que el valor default sea 1. No se acepte 0, para evitar la solicitud vía canal de ventas web de menores no acompañados que lleva consigo un proceso interno de la línea aérea para su autorización.

6. **Pasajeros infantes:**

Por una lógica propia de la industria, debe existir previamente un adulto antes de agregar un infante. Si existe más de un pasajero Infante, debe existir la misma cantidad de adultos puesto que solamente se puede asociar un pasajero infante y solo uno a un pasajero adulto.